



INSTITUTO
TECNOLÓGICO
DE AERONÁUTICA

Curso de Especialização em Data Science

Elton Felipe Sbruzzi
Filipe Alves Neto Verri
Vitor Venceslau Curtis
(Orgs.)

Projetos de Pesquisa em Data Science

Estudos de Caso do Curso CEDS IBAPE-PR



IBAPE-PR
INSTITUTO BRASILEIRO
DE AVALIAÇÕES E PERÍCIAS DE ENGENHARIA

científica digital



EDITORA CIENTÍFICA DIGITAL LTDA

Guarujá - São Paulo - Brasil

www.editoracientifica.com.br - contato@editoracientifica.com.br

Diagramação e Arte	Edição © 2024 Editora Científica Digital
Equipe Editorial	Texto © 2024 Os Autores
Imagens da Capa	1ª Edição - 2024
Adobe Stock - 2024	Acesso Livre - Open Access
Revisão	
Os Autores	

© COPYRIGHT DIREITOS RESERVADOS. A editora detém os direitos autorais pela edição e projeto gráfico. Os autores detêm os direitos autorais dos seus respectivos textos. Esta obra foi licenciada com uma Licença de Atribuição Creative Commons – Atribuição 4.0 Internacional, permitindo o download e compartilhamento integral ou em partes, desde que seja citada a fonte, com os créditos atribuídos aos autores e obrigatoriamente no formato Acesso Livre (Open Access) e sem a possibilidade de alteração de nenhuma forma. É proibida a catalogação em plataformas com acesso restrito e/ou com fins comerciais.



Dados Internacionais de Catalogação na Publicação (CIP)

P964

Projetos de pesquisa em Data Science - Estudos de caso do curso CEDS IBAPE-PR / Organização de Elton Felipe Sbruzzi, Filipe Alves Neto Verri, Vitor Venceslau Curtis. – Guarujá-SP: Científica Digital, 2024.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui Bibliografia

ISBN 978-65-5360-680-7

DOI 10.37885/978-65-5360-680-7

1. Pesquisa acadêmica no Brasil. I. Sbruzzi, Elton Felipe (Organizador). II. Verri, Filipe Alves Neto (Organizador). III. Curtis, Vitor Venceslau (Organizador). IV. Título.

CDD 001.42

Elaborado por Janaína Ramos – CRB-8/9166

Índice para catálogo sistemático:

I. Pesquisa acadêmica no Brasil

E-BOOK

ACESSO LIVRE ON LINE - IMPRESSÃO PROIBIDA

2024

Elton Felipe Sbruzzi
Filipe Alves Neto Verri
Vitor Venceslau Curtis
(Orgs.)

Projetos de Pesquisa em Data Science : estudos de caso do curso CEDS IBAPE-PR

1ª EDIÇÃO



científica digital

2024 - GUARUJÁ - SP

CONSELHO EDITORIAL

Prof. Dr. André Cutrim Carvalho
Prof. Dr. Antônio Marcos Mota Miranda
Profª. Ma. Auristela Correa Castro
Prof. Dr. Carlos Alberto Martins Cordeiro
Prof. Dr. Carlos Alexandre Oelke
Profª. Dra. Caroline Nóbrega de Almeida
Profª. Dra. Clara Mockdece Neves
Profª. Dra. Claudia Maria Rinhel-Silva
Profª. Dra. Clecia Simone Gonçalves Rosa Pacheco
Prof. Dr. Cristiano Marins
Profª. Dra. Cristina Berger Fadel
Prof. Dr. Daniel Luciano Gevehr
Prof. Dr. Diogo da Silva Cardoso
Prof. Dr. Ernane Rosa Martins
Prof. Dr. Everaldo dos Santos Mendes
Prof. Dr. Fabricio Gomes Gonçalves
Profª. Dra. Fernanda Rezende
Prof. Dr. Flávio Aparecido de Almeida
Profª. Dra. Francine Náthalie Ferraresi Queluz
Profª. Dra. Geuciane Felipe Guerim Fernandes

Prof. Dr. Humberto Costa
Prof. Dr. Joachin Melo Azevedo Neto
Prof. Dr. Jónata Ferreira de Moura
Prof. Dr. José Aderval Aragão
Prof. Me. Julianno Pizzano Ayoub
Prof. Dr. Leonardo Augusto Couto Finelli
Prof. Dr. Luiz Gonzaga Lapa Junior
Prof. Me. Marcelo da Fonseca Ferreira da Silva
Profª. Dra. Maria Cristina Zago
Profª. Dra. Maria Otília Zangão
Prof. Dr. Mário Henrique Gomes
Prof. Dr. Nelson J. Almeida
Prof. Dr. Octávio Barbosa Neto
Prof. Dr. Pedro Afonso Cortez
Prof. Dr. Reinaldo Pacheco dos Santos
Prof. Dr. Rogério de Melo Grillo
Profª. Dra. Rosenery Pimentel Nascimento
Prof. Dr. Rossano Sartori Dal Molin
Prof. Me. Silvio Almeida Junior
Profª. Dra. Thays Zigante Furlan Ribeiro
Prof. Dr. Wescley Viana Evangelista
Prof. Dr. Willian Carboni Viana
Prof. Dr. Willian Douglas Guilherme

Acesse a lista completa dos Membros do Conselho Editorial em www.editoracientifica.com.br/conselho

Parecer

Esta obra é uma produção colaborativa, tornando-se uma coletânea com reservas de direitos autorais para os autores. Alguns capítulos podem ser derivados de outros trabalhos já apresentados em eventos acadêmicos ou derivados de teses e dissertações, todavia, os autores foram instruídos ao cuidado com o autoplágio. A responsabilidade pelo conteúdo de cada capítulo, bem como por sua originalidade e pela integridade das informações publicadas, recai exclusivamente sobre os respectivos autores e autoras, não representando, necessariamente, a opinião da editora, dos organizadores ou dos membros do conselho editorial.

APRESENTAÇÃO

Temos o prazer de apresentar a Coletânea Projetos de Pesquisa em Data Science - Estudos de caso do curso CEDS IBAPE-PR. Este volume reúne os projetos finais desenvolvidos pelos alunos da turma de 2024, composta por profissionais e líderes em diversas áreas de atuação, incluindo profissionais do IBAPE-PR.

O CEDS ITA

O Curso de Especialização em Data Science (CEDS) do ITA é uma pós-graduação lato sensu que combina o rigor acadêmico e a excelência do ITA com uma forte ênfase na aplicação prática. O programa visa capacitar profissionais na área de Ciência de Dados, preparando-os para enfrentar desafios complexos e gerar insights estratégicos através do uso de técnicas e ferramentas avançadas. O curso destaca-se por sua abordagem prática, focada em solucionar problemas reais e gerar valor para diversas áreas de negócio.

O ITA

O **Instituto Tecnológico de Aeronáutica (ITA)**, fundado em 1950, é uma instituição pública de ensino superior e pesquisa, reconhecida pela sua excelência acadêmica e sua relevância internacional. Situado em São José dos Campos, SP, o ITA oferece programas de graduação em Engenharia e diversas opções de pós-graduação, incluindo o CEDS em Data Science. A instituição é renomada por sua formação de alto nível em áreas estratégicas, contribuindo significativamente para o desenvolvimento tecnológico e científico do Brasil.

A IBAPE-PR

O Instituto Brasileiro de Avaliações e Perícias de Engenharia do Paraná (IBAPE-PR), fundado em 1978, é uma entidade de referência que reúne engenheiros, arquitetos e agrônomos especializados em avaliações e perícias de engenharia. Com sede em Curitiba, o instituto tem como missão a busca pela inovação, a divulgação de informações e avanços técnicos, além de promover o desenvolvimento contínuo e a excelência profissional de seus associados.

Um dos marcos mais significativos dessa trajetória foi a criação do Curso de Especialização em Data Science (CEDS), uma pós-graduação lato sensu oferecida em parceria com o Instituto Tecnológico de Aeronáutica (ITA). A primeira turma do curso foi concluída em 2024, e a segunda turma teve início no mesmo ano, mantendo o formato, a qualidade e o rigor técnico que caracterizam o curso, atendendo à crescente demanda por especialização nas áreas emergentes da engenharia.

O sucesso desse projeto inovador é fruto da liderança visionária do Presidente do IBAPE-PR durante os biênios 2020/2021 e 2022/2023, Eng. Civil João Augusto Barão Michelotto, e do Eng. Civil Luciano Ventura, atual Presidente do IBAPE Nacional. Ambos desempenharam papéis essenciais no desenvolvimento e implementação deste curso de pós-graduação em parceria com o ITA, consolidando um avanço significativo na formação em Data Science para profissionais de engenharia e áreas afins.

Sob a liderança do atual presidente do IBAPE-PR, Eng. Edson Haluch, o curso segue sua trajetória de sucesso com a segunda turma, iniciada em agosto de 2024. Sob sua gestão, o instituto continua a fortalecer seu compromisso com a inovação e a qualificação de seus associados, garantindo que o curso se mantenha como um marco de vanguarda na formação técnica de profissionais da engenharia, avaliações e perícias.

O IBAPE-PR expressa profunda gratidão a Thaís Ramim, Carmen Lúcia de Almeida Belem e Maria do Carmo Falavinha Marcon, cujos esforços administrativos, dedicados e competentes foram essenciais para o êxito deste projeto.

Os Trabalhos de Conclusão de Curso

Os Trabalhos de Conclusão de Curso (TCCs) desta coletânea apresentam soluções inovadoras para uma variedade de desafios enfrentados atualmente. Utilizando técnicas de Data Science, os alunos aplicaram seus conhecimentos para resolver problemas reais, demonstrando a capacidade transformadora da Ciência de Dados em diferentes contextos.

Conclusão

Esta Coletânea de Trabalhos de Conclusão de Curso do CEDS ITA é uma prova do impacto positivo que a aplicação da Ciência de Dados pode ter em diversas áreas. Os projetos aqui apresentados refletem a qualidade do ensino oferecido pelo CEDS ITA e o imenso potencial da Data Science para gerar insights valiosos e informar decisões estratégicas.

Convidamos você a explorar os trabalhos desta coletânea e a descobrir como a Data Science pode ser um poderoso aliado na solução de problemas reais.

SUMÁRIO

Capítulo 01

TÉCNICAS DE MACHINE LEARNING PARA IDENTIFICAÇÃO DE DESVIOS DE PERFORMANCE EM PARQUES EÓLICOS

Paulo Régis Cavalcanti Oliveira; Filipe Alves Neto Verri

doi 10.37885/241218352 12

Capítulo 02

COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR FINANCIAL TREND PREDICTION USING OHLCV DATA AND TECHNICAL INDICATORS

Diogo Xavier Gouveia Junior; Elton Felipe Sbruzzi

doi 10.37885/241218354 21

Capítulo 03

ANALISE PREDITIVA DAS QUANTIDADES DE SERVIÇOS DE INTERNAÇÃO HOSPITALAR UTILIZADOS ATRAVÉS DOS EDITAIS DE CREDENCIAMENTO NO HOSPITAL DE FORÇA AÉREA DE BRASÍLIA

Ricardo Varela de Araújo Filho; Emília Colonese Carrard

doi 10.37885/241218356 48

Capítulo 04

DETECTION OF ANOMALOUS VEHICLE INTERACTIONS ON HIGHWAYS: TOWARDS AN AUTOMATED METHOD FOR APPLICATION ON COMPUTER VISION DATA OBTAINED FROM TRAFFIC CAMERAS

Guilherme Horbatink Villanova; Elcio Hideiti Shiguemori

doi 10.37885/241218357 68

Capítulo 05

MODELO DE INTELIGÊNCIA ARTIFICIAL PARA PREDIÇÃO DO PREÇO DO BITCOIN UTILIZANDO ANÁLISE DE SENTIMENTO DE DADOS DO TWITTER

Douglas Lessmann; Emilia Colonese Carrard

doi 10.37885/241218358 102

Capítulo 06**SISTEMA DE NEGOCIAÇÃO DE DÓLAR FUTURO UTILIZANDO REGRESSÃO LOGÍSTICA**

Ramon Vitor Malaquias Serra; Elton Felipe Sbruzzi

doi 10.37885/241218359 119**Capítulo 07****INTEGRAÇÃO DE MACHINE LEARNING E SISTEMAS AUTOMATIZADOS NA AVALIAÇÃO DE IMÓVEIS**

Luciano Ventura; Filipe Alves Neto Verri

doi 10.37885/241218360 141**Capítulo 08****MODELO DE PREVISÃO PARA O MERCADO DE COMMODITIES AGRÍCOLAS CONSIDERANDO FATORES MACROECONÔMICOS E INDICADORES FINANCEIROS**

Eduardo Ferreira Vieira D'Almeida; William Ordiales Oliveira; Elton Felipe Sbruzzi

doi 10.37885/241218361 167**Capítulo 09****APLICAÇÃO DE APRENDIZAGEM DE MÁQUINA PARA PREVER A CRONICIDADE EM PACIENTES COM CHIKUNGUNYA**

Tiago Machado de Alencar; André Machado de Siqueira

doi 10.37885/241218362 180**Capítulo 10****UM PROTÓTIPO DE SISTEMA DE APOIO NO CUIDADO DE IDOSOS**

Marcus Vinício de Oliveira Braga; Marcos Ricardo Omena de Albuquerque Maximo; Johnny Cardoso Marques

doi 10.37885/241218363 197**Capítulo 11****UTILIZAÇÃO DE LLMS E GRAPHRAG PARA CONSULTAS NO SETOR FINANCEIRO**

Valmir Rodrigues Bueno Júnior; Lilian Berton

doi 10.37885/241218364 204

Capítulo 12

PREDICTIVE ANALYSIS OF LOAN DEFAULT USING MACHINE LEARNING MODELS

Alejandro Aviles; Fernando Luiz Scherer; João Víctor Neves Loose; Johnny Cardoso Marques

doi 10.37885/241218365 240

Capítulo 13

AN ATTENTION-BASED REGRESSION METHOD FOR REAL ESTATE VALUATION

Renato Quinderé Carneiro; Filipe Alves Neto Verri

doi 10.37885/241218366 274

Capítulo 14

ARTIFICIAL INTELLIGENCE MODEL FOR DEMAND PREDICTION THROUGH REMOTE SENSING

Wellington Brosko da Rosa; Elcio Hideiti Shiguemori

doi 10.37885/241218367 291

Capítulo 15

MÉTODOS PARA TRATAR MISSING DATA NA GERAÇÃO DE ENERGIA SOLAR: ESTIMANDO DESEMPENHO EM CENÁRIOS DE PERDAS DE DADOS

Glauco Lauria Marques Filho; Paulo André Lima de Castro

doi 10.37885/241218368 300

Capítulo 16

É POSSÍVEL ESTIMAR O RISCO DE ATRASOS OU PARALISAÇÕES DE OBRAS PÚBLICAS NO BRASIL COM O USO DE INTELIGÊNCIA ARTIFICIAL?

Diego de Lira Andrade; Filipe Alves Neto Verri

doi 10.37885/241218369 324

Capítulo 17

ESTUDO DE DESEMPENHO DE MODELOS DE LINGUAGEM DE GRANDE ESCALA EM MARATONAS DE PROGRAMAÇÃO

Vitor Laguerra Netto Gioia; Cesar Augusto Cavalheiro Marcondes

doi 10.37885/241218370 354

Capítulo 18**INSTANCE SEGMENTATION ALGORITHM FOR DETECTING AND CLASSIFYING DEFECTS IN AIRCRAFT FUSELAGE SKINS WITH CONVOLUTIONAL NEURAL NETWORKS**

Juan Pedro Baena Cassal; Elcio Hideiti Shiguemori

doi 10.37885/241218371 373**Capítulo 19****EXPLORING INTERPRETABILITY IN MACHINE LEARNING MODELS FOR HEALTHCARE DECISION SUPPORT SYSTEMS**

Leandro Zanon Siqueira; Lilian Berton

doi 10.37885/241218372 412**Capítulo 20****PROCESSING OF METEOROLOGICAL IMAGES TO PREDICT FLIGHT HOLDING MANEUVERS**

Stéfano Campos de Oliveira; Filipe Alves Neto Verri

doi 10.37885/241218373 428**Capítulo 21****ENHANCING REAL-TIME OPTIMIZATION WITH MACHINE LEARNING PREDICTIONS OF LOWER CALORIFIC VALUES**

Filipe Andersonn Teixeira da Silveira; Elcio Hideiti Shiguemori

doi 10.37885/241218374 445**Capítulo 22****MODELO PREDITIVO PARA IDENTIFICAÇÃO DE OPORTUNIDADES DE FINANCIAMENTO IMOBILIÁRIO EM INSTITUIÇÕES FINANCEIRAS**

Marcelo Kawate; Elton Felipe Sbruzzi

doi 10.37885/241218375 467**SOBRE OS ORGANIZADORES** 491**ÍNDICE REMISSIVO** 492

TÉCNICAS DE MACHINE LEARNING PARA IDENTIFICAÇÃO DE DESVIOS DE PERFORMANCE EM PARQUES EÓLICOS

Paulo Régis Cavalcanti Oliveira

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)
pauloregis@gmail.com

Filipe Alves Neto Verri

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

RESUMO

A potência gerada nas turbinas eólicas depende fortemente das condições ambientais, operacionais e da interação com turbinas e parques eólicos vizinhos. Uma das variáveis mais importantes para a análise da potência gerada é a velocidade do vento, esta explica a maior parte da variância da potência gerada, por outro lado, falhas de medição de vento, falhas nos sensores, etc. são bem comuns na operação de parques eólicos, portanto estimar a potência gerada sem considerar essa variável é de interesse do setor. Neste trabalho, utilizamos 2 anos de dados, com frequência de 10 minutos, provenientes do sistema SCADA de um parque eólico do Reino Unido. Esse parque possui seis turbinas de potência nominal de 2.050 KW. Foi feito um tratamento inicial dos dados, com remoção de alguns outliers, e posteriormente foi utilizado a técnica de regularização Lasso para selecionar as variáveis mais importantes para o modelo. Após seleção das variáveis, foi utilizado um modelo polinomial de grau 3 com regularização Ridge devido a combinação da flexibilidade da regressão polinomial com a estabilidade e a robustez da regularização Ridge, esse método é útil para se evitar overfitting, que pode ocorrer facilmente em modelos polinomiais. Foi obtido excelentes coeficientes de determinação - R^2 e baixos valores da Raiz do Erro Quadrático Médio - RMSE nos modelos de cada turbina.

INTRODUÇÃO

Uma das etapas iniciais do desenvolvimento de um projeto de parque eólico é a realização da certificação de vento medida na região do parque eólico e a respectiva certificação da estimativa de produção de energia. Esta estimativa de produção de energia é, muitas vezes, utilizada pelos empreendedores como meta de geração do parque eólico durante a operação dos mesmos.

Alguns fatores podem levar o não atingimento destas metas durante a operação dos parques, entre eles, velocidade do vento menores que as médias históricas, perdas elétricas reais maiores que as estimadas nos estudos, tempo de manutenções maiores que as utilizadas nos estudos, baixa performance do aerogerador no atingimento a curva de potência teórica, etc.

A curva de potência, foco desse estudo, mostram a relação entre a velocidade do vento e potência gerada, essa relação é geralmente definida nos anexos dos contratos de fornecimento das turbinas eólicas, onde existe uma tabela que mostra a relação entre os valores de velocidade de vento versus potência para vários cenários de densidade. A correta medição e acompanhamento da curva é fundamental para otimização da performance da geração em parques eólicos. A norma IEC 61400-12-1 traz os critérios para a correta medição, porém é extremamente custoso a realização da medição da curva de potência conforme os critérios definidos na norma. Apesar dos custos e dificuldades envolvidas, a utilização dessa técnica é prática comum no setor, pois são exigidas na grande maioria dos contratos de fornecimento de turbinas vigentes no Brasil, que preveem penalidades no não atingimento da performance contratual da curva. Por outro lado, apesar de útil para a prática do setor, não é suficiente para o monitoramento preciso do desempenho das turbinas eólicas. Por essa razão, existe uma vasta literatura sobre análise da curva de potência de turbinas eólicas para fins de monitoramento de desempenho, utilizando técnicas de aprendizado de máquina neste acompanhamento, pois além da não necessidade das formalidades da norma, possuem uma aplicação mais simples e usam basicamente dados do sistema de supervisão SCADA dos parques eólicos. A ideia por trás das técnicas de aprendizado de máquina é que os dados podem ser empregados para construir modelos de comportamento normal, e as anomalias são identificadas como discrepâncias entre as medições reais e as estimativas do modelo. Detectado um

desvio, ações corretivas planejadas podem ser executadas para se encontrar o motivo dos desvios.

Conforme comentado acima, as técnicas de aprendizado de máquina usam dados SCADA com frequência de 10 minutos, aqui chamaremos de variáveis. As variáveis disponíveis no sistema SCADA podem ser classificadas em: i) ambientais, ii) operacionais e iii) temperaturas internas [Cascianelli *et al.* 2022].

As variáveis ambientais são principalmente: velocidade do vento, intensidade da turbulência, direção do vento, temperatura ambiente, já as variáveis operacionais são principalmente: posição da nacele, velocidade do rotor, velocidade do gerador, ângulo das pás; posição do yaw e por fim, as variáveis de temperaturas internas são principalmente medidas: no hub, na nacele, inversores, conversores, rolamentos, geradores e transformador.

Uma das variáveis mais importantes para a análise da potência gerada é a velocidade do vento, esta explica a maior parte da variância da potência gerada, por outro lado, falhas de medição de vento, falhas nos sensores e viés [Rabanal *et al.* 2019], são bem comuns na operação de parques eólicos. Além disso utilizar esses dados falhos podem levar a uma performance não real, gerando grande impacto financeiro nos empreendimentos, portanto estimar a potência gerada sem considerar essa variável é de interesse do setor, e será o foco principal deste estudo.

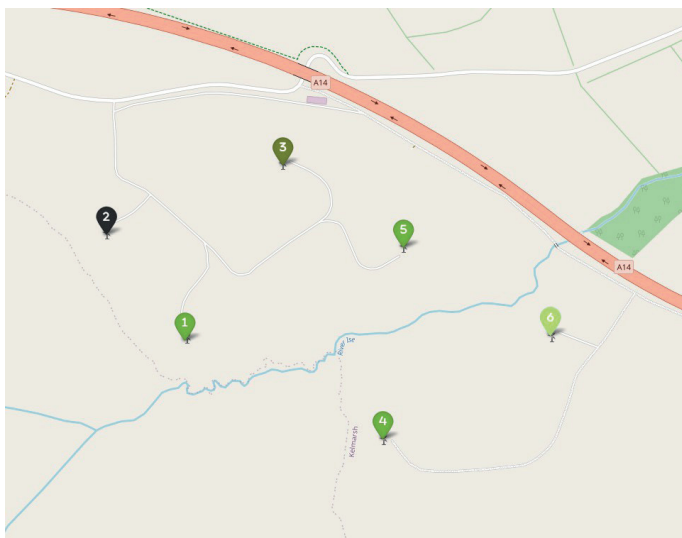
METODOLOGIA

A abordagem proposta utiliza técnicas de aprendizado de máquina para modelar o comportamento da potência ativa, output, em função de diversas variáveis, inputs, sendo que a velocidade do vento foi excluída das variáveis de input.

A base de dados selecionada, foi extraída de dados SCADA de um parque eólico no Reino Unido, composto por 6 turbinas eólicas, de 2.050 KW de potência nominal fornecido por [Plumley 2022].

Os dados foram selecionados para um período de 2 anos, composta inicialmente por 47 variáveis distintas e 604.015 linhas com intervalos de 10 minutos. O layout do parque eólico pode ser visto na figura 1.

Figura 1 - Layout do parque eólico.

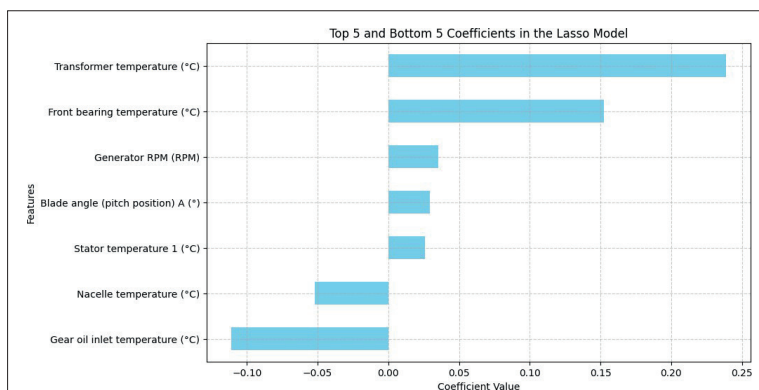


Todos os dados de 10 minutos afetados por parada de turbina foram removidos da análise, além disso foi realizada uma análise de outliers. Algumas colunas também foram excluídas em função de serem colunas auxiliares para o filtro de parada. Após esses tratamentos a base de dados ficou com 34 variáveis e 477.490 linhas.

Foi feito uma análise exploratória inicial para observarmos a relação entre as variáveis numéricas com a potência gerada.

Após conhecido o comportamento de cada variável, aplicamos uma técnica de regularização Lasso para selecionar as variáveis mais importantes para o modelo, resultando em 7 variáveis, além da potência gerada, nosso output. As variáveis selecionadas estão na figura 2, com os respectivos coeficientes do modelo de Lasso.

Figura 2 - Variáveis selecionadas após método de regularização Lasso.



Após seleção das variáveis, para estimarmos a potência gerada em função das variáveis de input selecionadas, utilizamos modelo polinomial de grau 3 com regularização Ridge, com 90% dos dados para treino e 10% para teste. Esse método foi escolhido devido a combinação da flexibilidade da regressão polinomial com a estabilidade e a robustez da regularização Ridge, esse método foi útil para evitar overfitting, que pode ocorrer facilmente em modelos polinomiais.

RESULTADOS

A performance dos modelos foram quantificadas através das métricas de RMSE, MAE e coeficiente de determinação R^2 , resumidas na tabela 1.

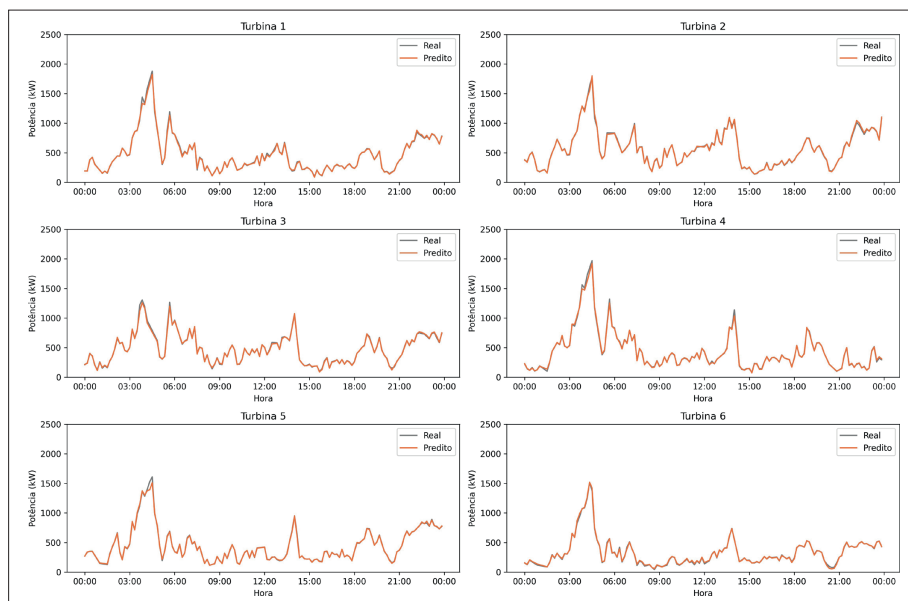
É possível observar que os modelos performaram muito bem, quando comparados com a tabela II de [Cascianelli *et al.* 2022]. A média do MAE de treino e teste foram 18,55 e 18,68 respectivamente, representando aproximadamente 0,9% da potência nominal.

Tabela 1 - Resultados de Métricas por Turbina.

WTG	Treinamento			Teste		
	RMSE	MAE	Score	RMSE	MAE	Score
1	30.63	18.25	0.998	31.52	18.48	0.998
2	32.04	19.25	0.998	31.82	19.20	0.998
3	31.79	19.80	0.998	31.16	19.51	0.998
4	31.15	18.47	0.998	31.34	18.67	0.998
5	29.72	17.51	0.998	31.09	18.12	0.998
6	29.31	18.04	0.998	29.37	18.13	0.998

De maneira visual, selecionado um dia aleatório qualquer, é possível observar na figura 3, que os valores de potência reais estão se comportando de maneira similar aos da potência predita, mostrando mais uma vez o desempenho satisfatório do modelo.

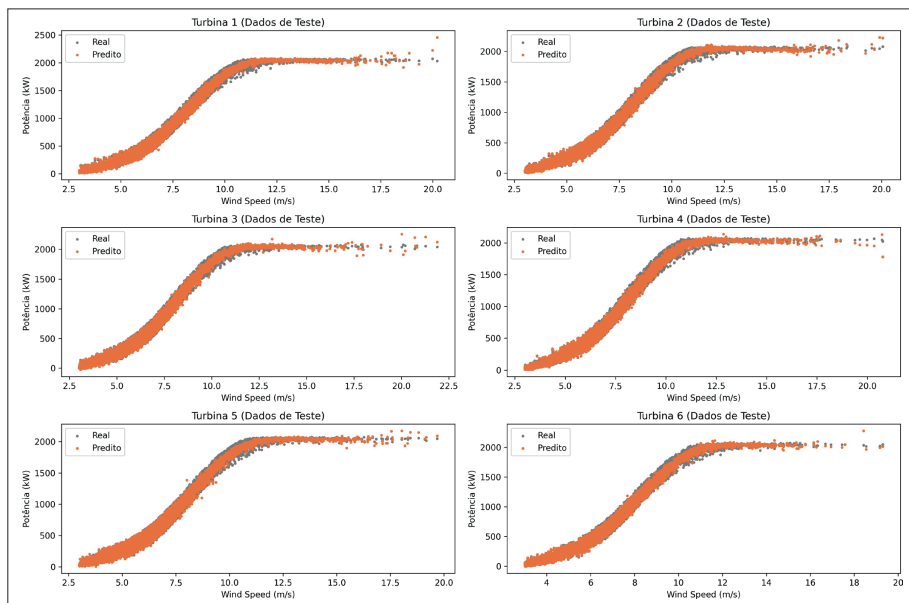
Figura 3 - Potência real versus potência predita para um dia.



Para que seja possível avaliar o modelo, conforme o setor normalmente avalia, ou seja, via análise da curva de potência, foi elaborado curvas de potência para as 6 turbinas, durante o período de teste, conforme pode ser visto na

figura 4 Mais uma vez é possível observar que a predição está bem alinhada com os valores reais.

Figura 4 - Curvas de potência real versus predita para o período de teste.



CONCLUSÃO

Neste trabalho de conclusão do curso, foi proposto prever a potência gerada por 6 turbinas eólicas sem considerar variáveis de velocidade do vento como input dos modelos. Através de técnicas de regularização Lasso para seleção de variáveis mais importantes para o modelo e regressão polinomial de grau 3 com regularização Ridge para modelar o comportamento, foi possível encontrar modelos com excelentes coeficientes de determinação - R^2 e baixos valores de RMSE e MAE, quando comparados com outros estudos similares.

Para complementar as visualizações deste estudo, foi elaborado um BI que pode ser acesso neste link.

REFERÊNCIAS

[Cascianelli *et al.* 2022] Cascianelli, S., Astolfi, D., Castellani, F., Cucchiara, R., and Fravolini, M. L. (2022). Wind turbine power curve monitoring based on environmental and operational data. *IEEE Transactions on Industrial Informatics*, 18(8).

[Plumley 2022] Plumley, C. (2022). Kelmash wind farm data. Conjunto de dados. [Rabanal *et al.* 2019] Rabanal, A., Ulazia, A., Ibarra-Berastegi, G., Sáenz, J., and Elosegui, U. (2019). Midas: A benchmarking multi-criteria method for the identification of defective anemometers in wind farms. *Energies*, 12(1).

COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR FINANCIAL TREND PREDICTION USING OHLCV DATA AND TECHNICAL INDICATORS

Diogo Xavier Gouveia Junior

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

Elton Felipe Sbruzzi

Division of Computer Science – Aeronautics Institute of Technology (ITA)

ABSTRACT

This study explores the application of machine learning models to predict financial market trends using OHLCV data and technical indicators. The models evaluated include Logistic Regression, Random Forest, MLP, and LSTM, with a focus on classifying "buy" and "sell" signals. Logistic Regression and LSTM models emerged as the top performers among the machine learning models in several tests, but they still fell short of the effectiveness of the Buy and Hold strategy.

INTRODUCTION

Contextualization

The use of Machine Learning (ML) models in finance is gaining momentum in both academia and industry. Deep Learning (DL) models, in particular, have garnered significant attention recently due to their superior performance over classical models such as Autoregressive Integrated Moving Average ARIMA or other type of regressors [Ozbayoglu *et al.* 2020].

However, despite these advancements, deploying DL models in practice remains challenging due to the high costs of training and the substantial data requirements [Mashrur *et al.* 2020]. These challenges often render the implementation of DL models, from the simplest to the most complex, across various datasets impractical. Moreover, as [Lohpetch and Corne 2009] highlights, many models tend to underperform when applied in a financial backtesting environment, often failing to surpass the performance of straightforward strategies like Buy and Hold, which involves holding an asset over an extended period without frequent trading.

This study is motivated by the need to leverage publicly accessible datasets, such as OHLCV (Open, High, Low, Close, Volume) prices, along with technical indicators derived from them, to predict daily market trends using a diverse spectrum of ML models, ranging from classical statistical algorithms like Logistic Regression to more advanced neural networks, such as Long Short-Term Memory (LSTM), which, while less dense than deep learning models, share a similar structure. Additionally, this research aims to compare the performance of these ML models against a simple Buy and Hold strategy. This approach seeks to provide valuable insights for individual investors while assessing whether advanced computational techniques offer a tangible advantage over traditional investment strategies.

Research Question

The central question guiding this study is: **“How do various machine learning models applied to an OHLCV dataset perform in terms of predictive**

accuracy and financial return in a backtesting environment, compared to a traditional Buy and Hold strategy".

Primary Objective

The primary objective of this study is to compare and evaluate the performance of various ML algorithms on an OHLCV dataset with and without technical indicators, focusing on their ability to predict market trends when implemented in a backtesting environment. This study aims to determine which model offers the best outcomes in classification and financial metrics, particularly when compared to traditional investment strategies like Buy and Hold.

Specific Objectives

The specific objectives present the actions intended to be achieved in the primary objective already presented. In this way, the following were identified:

- Deploy various ML algorithms (Multilayer Perceptron (MLP), Logistic Regression, Random Forest, LSTM) using an OHLCV dataset;
- Evaluate and compare the predictive accuracy of these algorithms, including an analysis of their performance when technical indicators—such as RSI, Williams
- %R, EMA, WMA, RoC, CCI, MACD, and Chaikin — are incorporated into the dataset, alongside the application of two feature normalization methods: Logarithmic and Min-Max normalization;
- Conduct a backtest to measure the financial performance of the predictions made by these algorithms, comparing them to a Buy and Hold strategy;
- Identify the model that offers the best financial return, risk-adjusted return, and predictive accuracy, providing insights into the practical applicability of these techniques in trading strategies.

RELATED WORK

What drives asset returns is a staple in modern finance [Bender *et al.* 2013]. One of the pioneering studies in creating a robust approach to the stock pricing prediction problem was conducted by [Sharpe 1964]. The development of the Capital Asset Pricing Model (CAPM) addressed the stock pricing prediction problem by using a single variable—the market beta—to explain the expected return. The use of technical indicators to recognize patterns for prediction of price trends is widely used. Studies like [Brock *et al.* 1992] have shown that simple technical indicator such as moving averages can yield significant returns.

The use of machine learning (ML) and classical models for predicting price trends is an area of ongoing research, with numerous approaches being explored but no exhaustive solution yet. [Tucker 1996] suggests that Logistic Regression is particularly suitable for financial applications, as it is well-adapted to modeling dichotomous outcomes, such as "buy" or "sell" signals. Unlike traditional multivariate regression methods, Logistic

Regression does not require assumptions of normally distributed errors or equal variance-covariance matrices, making it flexible and applicable in a wider range of scenarios.

Random Forest has also been recognized as a powerful tool for financial forecasting, especially in complex and volatile environments like stock markets. [Yin *et al.* 2023] emphasizes its advantages over traditional statistical models, highlighting its ability to manage diverse data inputs and capture non-linear patterns. By combining multiple decision trees, Random Forest enhances predictive accuracy and reduces overfitting. Optimizing the model through feature importance ranking and parameter tuning can further boost performance, making it an excellent choice for stock trend prediction.

MLP is another model well-suited for stock price prediction due to its ability to capture complex, nonlinear relationships within financial data. [Devadoss and Ligorì 2013] notes that MLPs are particularly adept at time-series forecasting tasks, as they can approximate any continuous function without assuming a specific data distribution. This adaptability makes MLPs highly effective for stock market predictions, which often involve dynamic and nonlinear price movements. In [Lazcano *et al.* 2024] the simplicity of MLPs makes them one of the most studied and

efficient models for prediction tasks, as they can effectively establish relationships between inputs and outputs. Despite lacking time dependence and being simpler than more complex models like LSTM, the author prefers MLPs due to their robustness and effectiveness in a wide range of prediction tasks, maintaining strong predictive power with lower complexity.

Additionally, LSTM networks, with their unique architecture of input, output, and forget gates, are particularly effective in financial time series forecasting. According to [Ozbayoglu *et al.* 2020], LSTMs excel in capturing both short-term and long-term dependencies, which is crucial for understanding temporal dynamics in stock prices. This makes them an ideal model for predicting future market movements by learning from historical trends. By using LSTMs, this study seeks to improve prediction accuracy by leveraging a model that can effectively address the sequential nature of stock price data, offering more robust performance compared to traditional approaches.

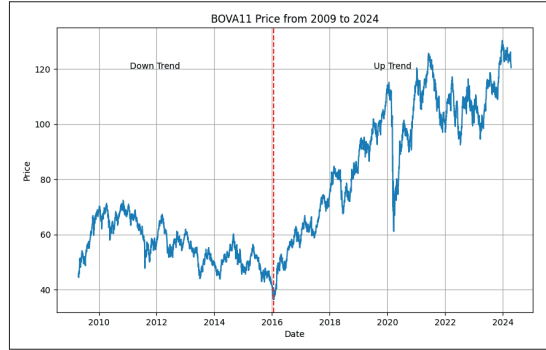
METHOD

This section outlines the approach and techniques employed in this study, detailing the dataset and technical indicators extraction, the data normalization, model deployment, and the selection of evaluation metrics for assessing model performance.

Data

Our sample period spans from April 2009 to April 2024 and it is defined by the availability of the BOVA11 (Exchange Traded Fund (ETF) that follows Brazilian market index iBovespa trend). Given the cyclic characteristics of financial time series, we decided to split the time frame between downward trend (from April 17, 2009 to January 26, 2016) and an upward trend (from January 27, 2016 to April 17, 2024) for testing the models in both circumstances. Figure 1 shows the plot of the whole series and the split moment that characterise the up and downward trend. The data was sourced from Yahoo Finance using the Python *yfinance* library [Aroussi 2021].

Figure 1 - Dataset BOVA11 up and downward trend split.



The OHLCV dataset was expanded by including several technical index. These indicators selectetion were based on the studies by [Sezer and Ozbayoglu 2018] and [Gudelek *et al.* 2017]. Table 1 provides the data dicionary of the dataset used. The formulas for calculating these indicators are detailed in the appendix in the technical index section. The technical indicators were extracted using the Python TA library [Pérez 2018].

Data Transformation and Training Setup

During the pre-processing stage of the pipeline, normalization was the primary transformation applied, as the dataset did not contain any missing values. This study explored two normalization techniques: the Min-Max normalization and the logarithmic transformation. In Min-Max normalization, defined by Equation 1, the observations of the features will range-typically between 0 and 1 [Bhanja and Das 2018].

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

Where:

- x_{norm} is the normalized value.
- x is the original value.
- $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the feature,

respectively.

In the training pipeline, beyond the train-test split, we employed a cross-validation approach known as stratified k-folds. This method involves dividing the dataset into k subsets, or "folds," and ensuring each class is uniformly distributed among the folds. In contrast, regular cross-validation randomly partitions the dataset into n folds without considering class distributions [Zeng and Martinez 2000]. We had used from the Python library Scikit-Learn the method *StratifiedKFold*¹ to implement this technique.

Additionally, we performed hyperparameter tuning using a algorithm called *RandomizedSearchCV*² also a method from the Scikit Learn library [Pedregosa *et al.* 2011]. Unlike traditional grid search methods that exhaustively test all possible combinations of hyperparameters, RandomizedSearchCV randomly selects a predefined number of hyper-parameter combinations. This approach is more efficient, especially when dealing with high-dimensional hyperparameter spaces, and allows for finding the best combination of hyperparameters that optimize model performance in a shorter amount of time. The training process was focused on achieving the highest accuracy for each model.

Table 1 - Dataset Dictionary - Variables Used in the Trained Models.

Feature Name	Description
Open	First traded price
High	Highest price of the day
Low	Lowest price of the day
Close	Closing price
Adjusted Close	Closing price adjusted for stocks splits and dividends payments
Volume	Traded volume in BRL
Close Shift	Diference between Adj Close _t – Adj Close _{t-1}
Return	Percent change of the close shift. Deleted after target conception

¹ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

² https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

Feature Name	Description
Relative Strength Index (RSI)	It provides a measure of the strength or weakness of a stock market over a recent period (was used 14 days window).
Williams %R	It measures overbought and oversold conditions in the market.
Exponential Moving Average (EMA)	It gives more weight to recent prices in calculating a moving average, making it more responsive to new information (was used 14 days window).
Weighted Moving Average (WMA)	It assigns more weight to recent data points in the moving average, placing greater emphasis on the latest information (was used 14 days window).
Rate of Change (RoC)	It measures the percentage change in price over a specified period, indicating the speed at which the price is changing (was used 14 days window).
Commodity Channel Index (CCI)	It identifies cyclical changes in stock market trends, helping to detect overbought and oversold conditions.
Moving Average Convergence and Divergence (MACD)	It indicates the relationship between two moving averages of a stock's price, used to identify changes in the strength, direction, momentum, and duration of a trend (was used 26/12 days window).
Chaikin Money Flow (CMF)	It measures the amount of Money Flow Volume over a specific period, indicating the buying and selling pressure on a stock.

The target feature was generated through a two-step process. First, using the "Close" feature of the dataset, we calculated the absolute difference between consecutive closing prices (close shift) and then computed the percent change of this difference (return). Based on this percent change, a binary target label was created: 0 for negative percent changes (indicating a sell signal) and 1 for positive percent changes (indicating a buy signal). After generating the target column, the Return feature were dropped to prevent model overfitting. This process is summarized in Equation 2.

$$\text{Target} = \begin{cases} \text{"Buy"} & \text{if } \frac{\text{Adj Close}_t - \text{Adj Close}_{t-1}}{\text{Adj Close}_{t-1}} > 0 \\ \text{"Sell"} & \text{if } \frac{\text{Adj Close}_t - \text{Adj Close}_{t-1}}{\text{Adj Close}_{t-1}} \leq 0 \end{cases} \quad (2)$$

Among the specific objectives of this study was the evaluation of the performance of various ML models within the OHLCV dataset and their effectiveness in a backtesting environment. The models applied in this analysis included:

Logistic Regression

Logistic regression is a fundamental statistical method widely used for binary classification problems. The model estimates the probability that a binary dependent variable Y equals 1 given a set of independent variables X . This is achieved by applying the sigmoid function which maps any real-valued number into the range between 0 and 1 [Hosmer Jr *et al.* 2013]. Specifically, the logistic regression model is defined as:

$$\hat{P}(Y = 1|X) = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \dots + \hat{\beta}_n X_n \quad (3)$$

Where:

- $\hat{P}(Y = 1|X)$ is the probability of the event (e.g., "buy" or "sell")
- $\sigma(z)$ is the sigmoid function, which maps any real-valued number to the range (0, 1)
- $\hat{\beta}_0$ is the intercept
- $\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_n$ are the coefficients

To enhance the performance of the Logistic Regression model, we utilized the `scikit-learn` library to implement and tune the model's hyperparameters. Table 2 summarizes the hyperparameters evaluated during this process.

Table 2 - Hyperparameters Tested for the Logistic Regression Model.

Parameter	Values Tested
Regularization Strength (C)	<code>uniform(0.01, 1)</code>
Regularization Penalty	None, L1 (Lasso), L2 (Ridge), Elastic Net (combination of L1 and L2)
Solver	<code>lbfgs</code> , <code>liblinear</code> , <code>saga</code>

Random Forest

Introduced by [Breiman 2001], random forests aggregate the predictions from multiple decision trees, making decisions based on the majority vote in classification tasks. They automatically analyze various factors to provide investment

advice, making decisions based on the majority vote from multiple decision trees. [Wang *et al.* 2024] points the advantages of random forests include their applicability to both regression and classification tasks, the ability to highlight the importance of input features, and their robustness against overfitting. However, the main drawback is that a large number of trees can slow down prediction times, making the algorithm less suitable for real-time applications. Equations 4 and 5 define the random forest model.

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_k(x)) \quad (4)$$

Where:

- $\hat{y}$ is the predicted final class.
- $T_1(x), T_2(x), \dots, T_k(x)$ are the individual predictions of the k trees.

Each tree $T_i(x)$ is defined as:

$$T_i(x) = \arg \max_c \sum_{j=1}^N I(h_{ij}(x) = c) \quad (5)$$

- c represents the possible classes.
- N is the number of nodes or splits considered within the i -th tree.
- $h_{ij}(x)$ is the decision at the j -th node of the i -th tree.
- $I(\cdot)$ is the indicator function that returns 1 if the condition is true (i.e., if the j -th node of the i -th tree predicts class c) and 0 otherwise.
- $\arg \max_c$ selects the class c with the highest sum of indications.

For this study, the model parameters for the Random Forest were selected to balance complexity and efficiency. The hyperparameters tested are summarized in Table 3.

Table 3 - Hyperparameters Tested for the Random Forest Model.

Parameter	Values Tested
Number of Estimators	range(50, 100)
Maximum Depth of Trees	range(3, 7)
Minimum Samples Split	range(5, 20)
Minimum Samples per Leaf	range(1, 10)
Bootstrap	True, False

Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) is a fundamental type of feedforward artificial neural network that consists of an input layer, one or more hidden layers, and an output layer. Introduced and popularized by [Rumelhart *et al.* 1986], MLP is a type of feedforward neural network composed of an input layer, one or more hidden layers, and an output layer. The network is trained using backpropagation, where errors in the output layer are calculated and weights in previous layers are adjusted iteratively to minimize this error. Each neuron in the network processes inputs from the previous layer, multiplies them by a set of weights, adds a bias, and then applies an activation function, typically a sigmoid or hyperbolic tangent in the hidden layers, to generate its output. The forward pass of an MLP can be mathematically represented as:

$$x_j^l = f \left(\sum_i w_{ji}^l x_i^{l-1} + b_j \right) \quad (6)$$

where:

- x_j^l is the output of the j -th neuron in the l -th layer,
- w_{ji}^l is the weight connecting the i -th neuron in the $(l - 1)$ -th layer to the j -th neuron in the l -th layer,
- x_i^{l-1} is the output of the i -th neuron in the $(l - 1)$ -th layer,
- b_j is the bias term for the j -th neuron in the l -th layer,
- $f(\cdot)$ is the activation function, such as the sigmoid or hyperbolic tangent function.

For a network with one hidden layer and one output neuron, the output of the MLP can be represented as described in equations 6 and 7:

$$z = f \left(\sum_{j=1}^N w_j y_j + b_z \right) \quad (7)$$

where:

- z is the output of the MLP,
- y_j is the output of the j -th neuron in the hidden layer,
- w_j is the weight connecting the j -th neuron in the hidden layer to the output neuron,
- b_z is the bias term for the output neuron.

The activation function $f(\cdot)$ in the hidden layers is typically a sigmoid or hyper-bolic tangent function, and in the output layer, it is often linear for regression tasks or a softmax function for classification tasks.

For this study, the MLP model was implemented using the `scikit-learn` library, and hyperparameter tuning was performed to optimize the model's performance. The hyperparameters tested are summarized in Table 4.

Table 4 - Hyperparameters Tested for the Multilayer Perceptron Model.

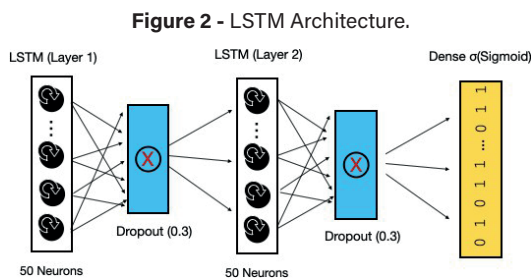
Parameter	Values Tested
Hidden Layer Sizes	• Single hidden layer: (32) • Two hidden layers: (64, 32) • Three hidden layers: (100, 100, 50)
Activation Functions	tanh, relu
Solvers	adam, sgd
Regularization Parameter	uniform(0.0001, 0.05)
Learning Rate	constant, adaptive

Long Short Term Memory (LSTM)

For [Ozbayoglu *et al.*, 2020] LSTM has become the dominant model in financial time series forecasting, where the temporal nature of the data is crucial. The LSTM (Long Short-Term Memory) [Hochreiter and Schmidhuber 1997]

network is a specialized type of deep learning model designed for sequential data analysis. Its key advantage is the ability to retain both short-term and long-term dependencies in the data, making it particularly well-suited for tasks like time-series forecasting, speech recognition, and language translation. LSTM networks consist of units that include input, output, and forget gates, which regulate the flow of information and allow the network to remember important values over varying time intervals. Due to its robust structure and adaptability.

Figure 2 defines the architecture starts with an LSTM layer containing 50 units, which processes the input data and captures temporal dependencies. The output is fed into the next LSTM layer, allowing for the extraction of more complex patterns across time steps. The use of Dropout layers with a rate of 0.3 after each LSTM layer helps prevent overfitting by randomly dropping units during training, which forces the network to generalize better. The network's final layer is a Dense layer with a sigmoid activation function, which outputs a probability score suitable for binary classification tasks, such as predicting "buy" or "sell" signals. Regarding the parameters to be tuned, the model includes the batch size of 16, 32, and 64 were tested to find the optimal balance between training speed and model convergence and epochs that were tested at 20 and 50 to determine the ideal amount of training without overfitting or underfitting.



Evaluation metrics

Machine Learning Classification Metrics

In evaluating the performance of machine learning models in financial time series, metrics such as accuracy, precision, recall, and F1-score are vital for assessing the predictive capabilities of the models. According to [Olorunnimbe

and Viktor 2023], accuracy provides a general measure of how often the model's predictions are correct, while precision and recall offer more detailed insights into the model's ability to correctly identify specific classes, such as "buy" and "sell" signals. Precision reflects the proportion of true positive predictions among all positive predictions, whereas recall indicates the proportion of actual positives that the model successfully identifies. The F1-score, a harmonic mean of precision and recall, is particularly useful in scenarios with imbalanced datasets, as it balances the trade-off between these two metrics. The formulas used for these metrics are summarized in Table 5.

Table 5 - Evaluation Metrics and Their Formulas.

Metric	Formula
Accuracy	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\text{Precision} = \frac{TP}{TP + FP}$
Recall (True Positive Rate)	$\text{Recall} = \frac{TP}{TP + FN}$
F1 Score	$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Financial Metrics

In addition to evaluating machine learning models through traditional classification metrics, it is crucial to assess their financial performance, particularly within the context of trading strategies. This study employs a backtesting approach using the Backtrader library [Rodriguez 2023], which simulates the application of trading strategies on historical data to evaluate their real-world applicability. Orders are executed at the next day's opening price, and positions are held until a new order signal is received, without any stop gain/loss mechanisms or order management for open positions. To manage risk, the strategy limits the order size to 1% of the current cash balance. Transaction costs are not considered in this analysis.

When assessing the performance of trading strategies, three critical financial metrics are often employed: return, volatility, and the Sharpe ratio. Return

quantifies the percentage change in the portfolio's value, indicating the overall profitability of the strategy. It is calculated using the initial price and the final price of the portfolio. Equation 8 describe this:

$$Return = \frac{P_f - P_i}{P_i} * 100 \quad (8)$$

Where:

- P_f is the final price
- P_i is the initial price

Volatility, on the other hand, measures the extent of variation in the portfolio's value, providing insight into the level of risk and uncertainty associated with the strategy [Hull *et al.*, 2013]. Higher volatility typically signals greater risk and potential for significant price fluctuations. Equation 9 detail this:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (R_i - \mu)^2} \quad (9)$$

Where:

- R_i is the period return
- N is the period
- μ average return of period
- σ Volatility

The Sharpe ratio (10) [Sharpe, 1966] , a key measure of risk-adjusted return, evaluates how much excess return a portfolio generates relative to its risk, where risk is defined as the standard deviation of portfolio returns. In this study, the excess return is calculated over a 10% annual risk-free rate. A higher Sharpe ratio reflects a more attractive risk-return profile, making it a crucial metric for comparing the effectiveness of different trading strategies.

$$SharpeRatio = \frac{R_p - R_f}{\sigma_p} \quad (10)$$

Where:

- R_p is the mean return
- R_f Risk-free rate
- σ_p volatility

To provide a benchmark, a Buy and Hold strategy will be implemented. The Buy and Hold strategy, which involves purchasing the asset at the beginning of the backtesting period and holding it until the end, is a widely recognized passive investment approach [Malkiel and Mei 1999]. It serves as a baseline for comparing the performance of machine learning-based trading strategies. By comparing the returns, volatility, and Sharpe ratio of the Buy and Hold strategy against those generated by the machine learning models, we can determine whether the models offer any advantage over a simple, passive investment strategy.

RESULTS AND DISCUSSION

In our experiments, the raw OHLCV data served as the baseline for evaluation. We conducted two experiments to assess the impact of adding TA indicators and normalizing the data during upward and downward trend timelines. The first experiment focused on the raw data during an uptrend, and the results are presented in Table 6.

From this baseline evaluation, it is observed that Logistic Regression notably out-performs the other models in terms of machine learning classification metrics in all metrics. In contrast, the LSTM and MLP models exhibit accuracies close to random guessing, indicating limited effectiveness when only raw data is used. The Random Forest model ranks second overall, demonstrating good general accuracy and a high recall for the Buy class. However, its precision for the Buy class is lower. This means that while the Random Forest model is effective at identifying most actual Buy signals (high recall), the lower precision indicates that a significant number of its Buy predictions are incorrect that is, it generates a considerable number of false positives.

From the financial metrics perspective, the Buy and Hold strategy outperforms the other strategies with a higher average monthly return and Sharpe ratio. Despite being the worst models in machine learning classification metrics, the LSTM and MLP strategies were the best among the ML models in terms of financial performance, achieving an average monthly return of 1.22% and a Sharpe Ratio of 0.06. The strong classification performance of Logistic Regression did not translate into favorable financial metrics, which highlights the necessity of backtesting strategies to assess their real-world applicability.

Table 6 - Performance Metrics for OHLCV Raw Data - Upward Trend.

Model	Accuracy		Precision		Recall		F1 Score		Return (%)	Volatility (%)	Sharpe
	Buy	Sell	Buy	Sell	Buy	Sell	Buy	Sell			
Logistic Regression	0.96	0.96	0.98	0.93	0.93	0.98	0.96	0.96	-0.077	0.6516	-1.3970
Random Forest	0.70	0.70	0.66	0.77	0.84	0.56	0.74	0.65	1.0076	5.0851	0.0342
MLP	0.51	0.51	0.51	0.00	1.00	0.00	0.68	0.00	1.2241	5.7915	0.0674
LSTM	0.51	0.51	0.51	0.00	1.00	0.00	0.68	0.00	1.2241	5.7915	0.0674
Buy and Hold	-	-	-	-	-	-	-	-	2.1851	8.5607	0.1579

The addition of technical analysis (TA) indicators and the Min-Max normalization transformation, as shown in Table 7, resulted in mixed outcomes in the classification metrics. Logistic Regression maintained the first place regarding overall accuracy, but experienced a decrease from 0.96 to 0.81. This decline was also observed in the Buy class performance, with reductions in recall and F1-score, indicating that the model became less effective at identifying Buy opportunities. The Random Forest model showed negligible change compared to the baseline. However, the LSTM and, notably, the MLP models demonstrated significant improvements. The addition of TA indicators and normalization enhanced these models' ability to predict Sell signals, which improved their overall metrics.

From a financial perspective, the models again did not outperform the Buy and Hold strategy in terms of average monthly return and Sharpe ratio. Among the machine learning models, the MLP was the best performer, achieving the highest return (0.87%) and a slightly positive Sharpe ratio (0.008). On the other hand, the Logistic Regression and LSTM models incurred negative returns and higher volatility, highlighting that the enhanced dataset did not lead to better financial results for

these models. This outcome underscores that while the addition of TA indicators and normalization can improve certain models' predictive capabilities, it does not necessarily translate into superior financial performance across all models. It also emphasizes the importance of evaluating both classification metrics and financial outcomes when assessing the effectiveness of trading models.

Table 7 - Performance Metrics for Min-Max + TA Indicators Data - Upward Trend.

Model	Accuracy		Precision		Recall		F1 Score		Return (%)	Volatility (%)	Sharpe
	Buy	Sell	Buy	Sell	Buy	Sell	Buy	Sell			
Logistic Regression	0.81	0.81	1.00	0.73	0.64	1.00	0.78	0.84	-1.6519	109.3140	-0.0227
Random Forest	0.69	0.69	0.70	0.68	0.70	0.69	0.70	0.69	0.0800	0.7501	-1.0043
MLP	0.79	0.79	0.72	0.92	0.95	0.62	0.82	0.74	0.8734	4.7849	0.0083
LSTM	0.64	0.64	0.70	0.60	0.51	0.77	0.59	0.68	-200.5061	947.0843	-0.2125
Buy and Hold	-	-	-	-	-	-	-	-	2.1851	8.5607	0.1579

In the baseline scenario for the downward trend shown in Table 8, the Logistic Regression model experienced a slight decrease in performance but remained the best among the machine learning models. The LSTM and MLP models again had the worst performance, now performing worse than random guessing. In contrast, the Random Forest model performed better than in the uptrend scenario, achieving higher accuracy and more balanced precision, recall, and F1-score metrics.

From the financial metrics perspective, the downward trend scenario benefited some of the machine learning models, especially since the Buy and Hold strategy did not contribute significantly to the monthly average return. Logistic Regression emerged as the top strategy, achieving monthly average returns exceeding 1% and recording the highest, although modest, Sharpe ratio of 0.02. This was the first instance where the model with the highest accuracy also yielded the highest return. The dichotomy between financial and classification performance observed previously with the LSTM and MLP models did not occur again, as these two models were the worst among the strategies in both metrics. The improvement in classification metrics did not translate into better financial results for the Random Forest model compared to Table 6, but among

the models tested in this scenario, it ranked second in return and first in terms of lower volatility.

Table 8 - Performance Metrics for OHLCV Raw Data - Downward Trend.

Model	Accuracy		Precision		Recall		F1 Score		Return (%)	Volatility (%)	Sharpe
	Buy	Sell	Buy	Sell	Buy	Sell	Buy	Sell			
Logistic Regression	0.87	0.87	0.97	0.82	0.73	0.98	0.84	0.89	1.004	6.8321	0.0249
Random Forest	0.79	0.79	0.77	0.80	0.75	0.82	0.76	0.81	0.1841	1.7399	-0.3731
MLP	0.45	0.45	0.45	0.00	1.00	0.00	0.62	0.00	-0.0981	2.8519	-0.3265
LSTM	0.45	0.45	0.45	0.00	1.00	0.00	0.62	0.00	-0.0981	2.8519	-0.3265
Buy and Hold	-	-	-	-	-	-	-	-	-0.0084	5.9712	-0.1409

In the scenario depicted by Table 9, which analyzes models enhanced with technical indicators and Min-Max normalization on downtrend data, we observe significant shifts in performance. The MLP model emerged as the top performer among machine learning models, achieving the highest accuracy of 0.87. It demonstrated balanced and high precision, recall, and F1-scores for both Buy and Sell classes, indicating effective

prediction of both signals. The Logistic Regression model experienced a decrease in accuracy to 0.74, with high precision for the Buy class (0.90) but lower recall (0.48), suggesting it was precise but missed over half of the actual Buy opportunities. The LSTM model showed moderate performance with an accuracy of 0.67, while the Random Forest model had the lowest accuracy at 0.64, highlighting varying effectiveness among the models in this scenario.

Table 9 - Performance Metrics for Min-Max + TA Indicators Data - Downward Trend.

Model	Accuracy		Precision		Recall		F1 Score		Return (%)	Volatility (%)	Sharpe
	Buy	Sell	Buy	Sell	Buy	Sell	Buy	Sell			
Logistic Regression	0.74	0.74	0.90	0.69	0.48	0.96	0.63	0.80	2.2659	13.5146	0.1060
Random Forest	0.64	0.64	0.57	0.74	0.77	0.53	0.66	0.62	-0.2044	2.0663	-0.5022
MLP	0.87	0.87	0.88	0.87	0.83	0.91	0.85	0.89	0.3125	2.4254	-0.2147
LSTM	0.67	0.67	0.64	0.69	0.59	0.73	0.61	0.71	0.002969	0.022812	-0.2351
Buy and Hold	-	-	-	-	-	-	-	-	-0.0084	5.9712	-0.1409

From a financial metrics perspective, the Logistic Regression model again achieved the highest monthly average return of 2.2659%, accompanied by the highest volatility (13.5146%) and a Sharpe ratio of 0.106002, indicating better returns but increased risk. Despite strong classification performance, the MLP model attained only a modest monthly return of 0.3125%, with low volatility and a negative Sharpe ratio, suggesting high predictive accuracy did not translate into significant financial gains. The LSTM model barely broke even, and the Random Forest model incurred a slight negative return. Notably, the Buy and Hold strategy resulted in a slight negative return (-0.0084%), underscoring challenges for passive investment in downtrend markets. Overall, the divergence between classification metrics and financial outcomes emphasizes the complexity of converting predictive accuracy into profitable trading strategies.

CONCLUSION

This study investigated the application of various machine learning (ML) models, including Logistic Regression, Random Forest, MLP, and LSTM, to predict daily market trends using financial time series data. The analysis explored the effects of Min-Max normalization and the addition of technical indicators on model performance during both upward and downward market trends.

The findings reviewed that in a downward trend scenario MLP, Random Forest and Logistic Regression can beat the Buy and Hold strategy in terms of return and Sharpe ratio, but in uptrend scenarios, the models did not outperformed the Buy and Hold. The study showed that models that achieved the highest classification metrics, including accuracy and F1 scores, did not necessarily translate into superior financial performance.

Contrary to expectations, the inclusion of technical indicators and data normalization techniques yielded mixed results. While these enhancements improved the predictive capabilities of certain models, they did not consistently lead to better financial performance. In some cases, they even resulted in increased volatility and negative returns. For example, the LSTM model, despite being adept at capturing temporal dependencies, underperformed financially and sometimes resulted in significant losses with high volatility. The Random Forest model demonstrated

balanced classification metrics but did not translate these into profitable trading strategies, often yielding minimal or negative returns.

These findings emphasize the need for further research into optimizing machine learning models for financial applications, particularly in the areas of feature engineering, data preprocessing, and model selection. While machine learning offers promising tools for analyzing financial data, this study illustrates the complexities and limitations of deploying these models in real-world trading scenarios. Ultimately, traditional strategies like Buy and Hold may still offer a more stable and reliable approach for individual investors, unless machine learning models can be better adapted to handle the inherent volatility and unpredictability of financial markets.

APPENDIX

Technical Indexes

Relative Strength Index (RSI) The RSI measures the magnitude of recent price changes to evaluate overbought or oversold conditions.

$$RSI = 100 - \frac{100}{1 + \frac{\text{Average Gain}}{\text{Average Loss}}}$$

Williams %R Williams %R is a momentum indicator that measures overbought and oversold levels.

$$\text{Williams \%R} = \frac{\text{Highest High} - \text{Close}}{\text{Highest High} - \text{Lowest Low}} \times -100$$

Exponential Moving Average (EMA) EMA gives more weight to recent prices in the calculation of a moving average.

$$EMA = \text{Price}_{\text{today}} \times \frac{2}{N+1} + EMA_{\text{yesterday}} \times \left(1 - \frac{2}{N+1}\right)$$

Weighted Moving Average (WMA) WMA assigns more weight to recent data points in the moving average calculation.

$$WMA = \frac{n \times \text{Price}_n + (n-1) \times \text{Price}_{n-1} + \dots + 1 \times \text{Price}_1}{n + (n-1) + \dots + 1}$$

Rate of Change (RoC) RoC measures the percentage change in price over a given period.

$$RoC = \frac{\text{Price}_{\text{today}} - \text{Price}_{n\text{-days ago}}}{\text{Price}_{n\text{-days ago}}} \times 100$$

Commodity Channel Index (CCI) CCI identifies cyclical trends in the market.

$$CCI = \frac{\text{Typical Price} - \text{SMA of Typical Price}}{0.015 \times \text{Mean Deviation}}$$

Moving Average Convergence and Divergence (MACD) MACD indicates the relationship between two moving averages of a stock's price.

$$MACD = EMA_{12\text{-day}} - EMA_{26\text{-day}}$$

Chaikin Money Flow (CMF) CMF measures the amount of Money Flow Volume over a specific period.

$$CMF = \frac{\sum_{i=1}^n \left(\frac{(\text{Close}_i - \text{Low}_i) - (\text{High}_i - \text{Close}_i)}{\text{High}_i - \text{Low}_i} \right) \times \text{Volume}_i}{\sum_{i=1}^n \text{Volume}_i}$$

LSTM

LSTM networks consist of memory cells that include input, output, and forget gates, which regulate the flow of information through the network [Goodfellow *et al.*, 2016]. The input gate controls the extent to which new information flows into the cell, the forget gate determines what information is discarded from the cell state, and the output gate decides how much of the cell state is used to compute the output. This gating mechanism allows LSTM networks to remember important values over varying time intervals, effectively capturing complex temporal patterns in data [Graves and Graves 2012].

The Long Short-Term Memory (LSTM) network maintains three gates: the input gate (i_t), forget gate (f_t), and output gate (o_t). These gates control the flow of information through the LSTM cells.

1. Gates in LSTM

1.1 Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

1.2 Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

1.3 Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

2. Cell State Update

2.1 Cell State

$$\begin{aligned}\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t\end{aligned}$$

3. Hidden State

$$h_t = o_t \cdot \tanh(C_t)$$

4. Activation Functions

4.1 Sigmoid Function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

4.2 Hyperbolic Tangent Function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

REFERENCES

- Aroussi, R. (2021). yfinance: Download market data from yahoo! finance's api. <https://github.com/ranaroussi/yfinance>. Accessed: August 2024.
- Bender, J., Briand, R., Melas, D., and Subramanian, R. A. (2013). Foundations of Factor Investing. *SSRN Electronic Journal*.
- Bhanja, S. and Das, A. (2018). Impact of Data Normalization on Deep Neural Network for Time Series Forecasting.
- Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.
- Brock, W., Lakonishok, J., and LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, 47(5):1731–1764.
- Devadoss, A. V. and Ligor, T. A. A. (2013). Forecasting of stock prices using multi layer perceptron. *International journal of computing algorithm*, 2(1):440–449.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. Google-Books-ID: Np9SDQAAQBAJ.
- Graves, A. and Graves, A. (2012). *Supervised sequence labelling*. Springer.
- Gudelek, U., Boluk, A., and Ozbayoglu, M. (2017). A deep learning based stock trading model with 2-D CNN trend detection. Pages: 8.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780. Conference Name: Neural Computation.
- Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.
- Hull, J., Treepongkaruna, S., Colwell, D., Heaney, R., and Pitt, D. (2013). *Fundamentals of futures and options markets*. Pearson Higher Education AU.
- Lazcano, A., Jaramillo-Morán, M. A., and Sandubete, J. E. (2024). Back to basics: The power of the multilayer perceptron in financial time series forecasting. *Mathematics*, 12(12):1920.
- Lohpetch, D. and Corne, D. (2009). Discovering effective technical trading rules with genetic programming: towards robustly outperforming buy-and-hold. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 439–444.
- Malkiel, B. G. and Mei, J. (1999). *Glofbal Bargain Hunting: The Investor's Guide to Profits in Emerging Markets*. Simon and Schuster.
- Mashrur, A., Luo, W., Zaidi, N. A., and Robles-Kelly, A. (2020). Machine Learning for Financial Risk Management: A Survey. *IEEE Access*, 8:203203–203223. Conference Name: IEEE Access.
- Olorunnimbe, K. and Viktor, H. (2023). Deep learning in the stock market—a systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, 56(3):2057–2109.
- Ozbayoglu, A. M., Gudelek, M. U., and Sezer, O. B. (2020). Deep Learning for Financial Applications : A Survey. arXiv:2002.05786 [cs, q-fin, stat].

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Pérez, R. (2018). Technical analysis library in python. <https://github.com/bukosabino/ta>.
- Rodriguez, D. (2023). *Backtrader Documentation*. Available at <https://www.backtrader.com/docu/>.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.
- Sezer, O. and Ozbayoglu, M. (2018). Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach. *Applied Soft Computing*, 70.
- Sharpe, W. F. (1964). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk*. *The Journal of Finance*, 19(3):425–442. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1964.tb02865.x>.
- Sharpe, W. F. (1966). Mutual fund performance. *The Journal of business*, 39(1):119–138.
- Tucker, D. J. (1996). NEURAL NETWORKS VERSUS LOGISTIC REGRESSION IN FINANCIAL MODELLING: A METHODOLOGICAL COMPARISON.
- Wang, J., Xu, J., Cheng, Q., and Kumar, R. (2024). Research on finance credit risk quantification model based on machine learning algorithm. *Academic Journal of Science and Technology*, 10(1):290–298.
- Yin, L., Li, B., Li, P., and Zhang, R. (2023). Research on stock trend prediction method based on optimized random forest. *CAAI Transactions on Intelligence Technology*, 8(1):274–284.
- Zeng, X. and Martinez, T. R. (2000). Distribution-balanced stratified cross-validation for accuracy estimation. *Journal of Experimental & Theoretical Artificial Intelligence*, 12(1):1–12.

ANALISE PREDITIVA DAS QUANTIDADES DE SERVIÇOS DE INTERNAÇÃO HOSPITALAR UTILIZADOS ATRAVÉS DOS EDITAIS DE CREDENCIAMENTO NO HOSPITAL DE FORÇA AÉREA DE BRASÍLIA

Ricardo Varela de Araújo Filho

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Emília Colonese Carrard

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

The Hospital de Força Aérea Brasileira (HFAB) is a healthcare organization that is part of the Brazilian public administration. For the HFAB to provide healthcare services to the beneficiaries of the Sistema de Saúde da Aeronáutica (SISAU), it often needs to use contracts with civilian healthcare organizations through Credentialing Notices. One of the challenges in drafting the Notices is quantifying the services and care that will be used through Credentialing. For this reason, this work aimed to find methodologies based on Artificial Intelligence algorithms to provide better quantification of these quantities. To achieve this, statistical models, machine learning and deep learning models were trained with the goal of predicting values in the time series of the quantities of hospital inpatient services used by HFAB through Credentialing.

RESUMO

O Hospital de Força Aérea (HFAB) é uma organização de saúde que faz parte da Administração pública brasileira. Para o HFAB prestar o atendimento de saúde aos beneficiários do Sistema de Saúde da Aeronáutica (SISAU), muitas vezes, necessita utilizar de contratos com organizações civis de saúde através dos Editais de Credenciamento. Uma das dificuldades na elaboração dos Editais é a quantificação dos serviços e atendimentos que serão utilizados através do Credenciamento. Por esse motivo, este trabalho procurou encontrar metodologias baseadas nos algoritmos da Inteligência Artificial a fim de proporcionar uma melhor quantificação das quantidades. Para tanto, foram treinados modelos estatísticos, de aprendizado de máquina e de aprendizado profundo com o objetivo de prever valores nas séries temporais das quantidades de serviços de internação hospitalar, utilizado pelo HFAB através do *Credenciamento*.

INTRODUÇÃO

O Hospital de Força Aérea Brasileira é uma organização de saúde da Força Aérea Brasileira que tem como missão prestar assistência médico-hospitalar e ambulatorial aos beneficiários do Sistema de Saúde da Aeronáutica (SISAU). Ao todos, dentre militares, seus dependentes, civis e pensionistas, o HFAB atende ao número de aproximadamente 26.000 (vinte e seis mil) beneficiários que demandam por serviços de saúde que vão de consultas de rotina até cirurgias complexas.

Para atender às demandas dos usuários, o efetivo do HFAB é composto por um corpo clínico multidisciplinar e equipado com aparelhos que executam diversos serviços médico-hospitalares e ambulatoriais. No entanto, apesar dos esforços contínuos para internalizar os atendimentos aos beneficiários, quando um determinado serviço de saúde não pode ser realizado pelo HFAB, os usuários do SISAU são encaminhados para pessoas físicas ou jurídicas contratadas por meio de credenciamento para a execução dos serviços.

O nome dado a essa forma de contratação é o credenciamento. Sendo assim, pode-se chamar, também, essas pessoas físicas e jurídicas contratadas de empresas credenciadas ou rede credenciada. O nome do contrato utilizado para firmar esse acordo é o “Termo de Adesão ao Credenciamento”, instrumento firmado através da publicação de um Edital de Credenciamento.

Por sua vez, o Edital de Credenciamento é um documento, publicado pela administração pública, que possui todas as informações referentes aos serviços que o HFAB necessita que sejam complementados pelas organizações de saúde, os valores base que serão pagos por tipo de serviço, a quantidade estimada que o Hospital pretende utilizar de um determinado serviço e os procedimento para habilitação e formalização do credenciamento.

O objetivo do Edital é dar ampla divulgação, ao setor privado, sobre a intenção que o HFAB tem em firmar credenciamento com as organizações de saúde e obter o maior número possível de interessados que se enquadrem nos critérios de habilitação previamente definidos. Para atingir esses objetivos, o Edital cita os principais serviços de saúde que os beneficiários do SISAU podem utilizar através da rede credenciada, o valor base que o HFAB irá pagar pelos

procedimentos realizados e a quantidade de atendimentos com prestações de serviços que o hospital pretende utilizar no período de um ano.

É com base nas informações do Edital que as organizações de saúde demonstram, ou não, o interesse em firmar o Termo de Adesão ao Credenciamento com o HFAB. Por essa razão, uma estimativa inadequada das quantidades dos serviços e dos valores utilizado pelo HFAB apresentada no Edital pode afastar o interesse das organizações de saúde em se credenciar com o HFAB, ou até mesmo gerar uma falsa expectativa de retorno econômico para outras empresas que decidirem firmar o Credenciamento.

Atualmente, os valores base informados no Edital para os procedimentos que podem ser realizados por meio do credenciamento são obtidos através das orientações da Instrução Normativa SEGES/ME nº 65 de 7 de julho de 2021. A instrução orienta que os preços sejam obtidos por meio de bancos de preços públicos, mídia especializada, outras contratações similares ou por meio de pesquisa direta com os fornecedores. Por conta da necessidade de seguir as orientações da IN nº 65/2017, este trabalho não irá abordar a previsão dos valores dos procedimentos médicos informados no Edital. O foco será a quantidade de serviços ou atendimentos estimadas no Edital.

No último Edital de Credenciamento, as quantidades dos serviços delimitados foram determinadas através da aplicação de média simples dos atendimentos realizados, agrupando os serviços de características semelhantes em grupos definidos pela área técnica. Esses dados foram obtidos pelo Sistema de Saúde Complementar (SISAUC), que armazena todas as informações dos serviços prestados pelas Credenciadas.

Por utilizar-se de média simples, o objetivo deste trabalho é aplicar técnicas de análise preditiva em séries temporais para verificar se é possível determinar, com uma melhor taxa de acurácia, as quantidades de serviços/atendimentos a serem divulgadas nos Editais de Credenciamento, a fim de alcançar uma maior eficiência administrativa.

Para tanto, o capítulo 2 irá apresentar as séries temporais e introduzir às principais técnicas utilizadas para realizar a análise preditiva de séries temporais. O capítulo 3 irá explicar qual foi a metodologia de avaliação adotada por este trabalho para alcançar o objetivo e irá demonstrar como o experimento de treinamento dos modelos foi realizado. Já o capítulo 4 irá demonstrar os resultados

e as análises obtidas através da aplicação do experimento. Por fim, as conclusões e considerações finais serão definidas no capítulo 5.

BACKGROUND E TRABALHOS RELACIONADOS

Séries Temporais

As séries temporais são dados que podem ser provenientes de vários tipos diferente de fontes e campos de atuação. Eles abrangem desde históricos de compras até medições feitas por equipamentos ou sensores (Nielsen, 2021, p. xi). Pelo fato da análise de séries temporais ser uma tentativa de usar extrair resumos e informações estatísticas de pontos de dados organizados cronologicamente, é comum encontrar uma ampla aplicação dessa análise nas diversas áreas do conhecimento (Nielsen, 2021, p. 1). Dessa maneira, ao longo dos anos, uma grande quantidade de técnicas estatísticas e de aprendizado de máquina foram elaboradas com o objetivo de realizar essas análises (Cao, 2024, p. 1), obtendo resultados como, por exemplo, a predição de comportamentos futuros.

Dentre as principais técnicas desenvolvidas, pode-se destacar as baseadas em métodos estatísticos, as que utilizam de ferramentas clássicas do aprendizado de máquina, as baseadas em redes de aprendizado profundo, as que utilizam metodologias ensemble, combinando o resultado de vários modelos e, mais recentemente, modelos de previsão que se utilizam da tecnologia dos Transformers, modelo esse que foi criado para resolver alguns problemas de limitação das redes neurais e dos grandes volumes de dados (Cristian, 2024, p. 6).

Além disso, é importante ressaltar a crescente necessidade da inovação e da aplicação de métodos de aprendizado de máquina para melhoria das tomadas de decisão e dos processos feitos nas áreas da saúde, como em administração de hospitais, e no setor público. Quanto a este último, o princípio da eficiência é um dos pilares que sustentam a missão das organizações públicas. No entanto, para superar os problemas de morosidade processual, déficit de pessoal e de recursos financeiros, tem-se procurado buscar soluções de tecnologia e de inteligência artificial para contribuir com os objetivos institucionais (Desordi, 2020, p. 2) e (Veras *apud* Gomes, 2022, p. 7).

No Brasil, por exemplo, o Tribunal de Contas da União (TCU), um órgão público brasileiro, tem utilizado de novas tecnologias, como o robô Alice, acrônimo para Analisador de Licitações, Contratos e Editais, para aumentar sua produtividade (Desordi, 2020, p. 13). A Alice é uma ferramenta criada com o objetivo de identificar automaticamente indícios de irregularidades nas licitações, através do uso da IA, que faz uma varredura nos dados das licitações publicadas pelos entes públicos (Panis, 2020, p. 16).

Além disso, nos últimos dias, iniciativas, como a utilização de modelos de análises preditivas, tem desempenhado um papel fundamental na medicina devido ao grande volume de dados da saúde que tem sido produzido por diversas fontes (Badawy, 2023, p. 2). Contudo, para o funcionamento adequado de um hospital, é necessário que toda sua cadeia de suprimento também funcione, haja em vista que sem os medicamentos e os materiais médico-hospitalares não é possível fornecer um serviço de saúde adequado aos pacientes. Para o caso das cadeias de suprimento, já é comum a aplicação de modelos estatísticos, como análises de séries temporais e análises de regressões, para prever a demanda por insumos.

Com tudo isso, deve-se considerar que o Hospital de Força Aérea de Brasília (HFAB), além de ser um órgão público, é também uma organização do sistema de saúde. Por esse motivo, as dificuldades e os problemas desses dois setores também podem ser enfrentados pela administração do HFAB.

Modelos para Séries Temporais

A fim de realizar as análises das séries temporais, diversos modelos foram criados e aprimorados ao longo do tempo. Este subtópico irá apresentar alguns dos modelos que foram aplicados neste trabalho.

Modelos baseados em métodos estatísticos

Os modelos baseados em métodos estatísticos utilizam das correlações existentes entre os pontos de dados da própria série temporal para prever futuros pontos. Por sua simplicidade, continuam sendo amplamente utilizados para tarefas de predição em séries temporais, tanto em pesquisas acadêmicas como em variados campos de atuação (Nielsen, 2021, p. 159). Os modelos estatísticos

mais comuns são os modelos autorregressivos (AR), os modelos de média móvel (MA) e os modelos autorregressivos integrados de média móvel (ARIMA).

O modelo autorregressivo (AR), toma como premissa de que o passado prediz o futuro. Logo, ele supõe que o valor de um ponto em uma série temporal, pode ser explicado por uma função dos valores da série em pontos anteriores no tempo. (Nielsen, 2021, p.162). Já o modelo de média móvel (MA), ao invés de utilizar valores passados da variável de previsão em uma regressão, um modelo de média móvel, ele utiliza os erros de previsão passados em um modelo semelhante a uma regressão (Hyndman, 2021, Cap. 8.4).

Por fim, o modelo autorregressivo integrado de média móvel (ARIMA) é a combinação dos modelos AR e MA, com a adição de uma etapa de diferenciação (Nielsen, 2021, p. 181). A diferenciação é um método utilizado para garantir a estacionariedade dos dados. Nela subtrai-se o valor anterior ao valor base, evitando, assim, a presença de tendências.

Esses modelos tradicionais, no entanto, apresentam, com frequência, limitações, quando comparados com modelos de aprendizado de máquina e aprendizado profundo para séries temporais, principalmente quando se trata de grandes conjuntos de dados (Cristian, 2024, p. 4).

Modelos baseados em aprendizado de máquina

De acordo com Masini (2021): “Aprendizado de Máquina (ML) é a combinação de algoritmos computacionais automatizados com métodos estatísticos poderosos para aprender (descobrir) padrões ocultos em conjuntos de dados complexos” (p. 2), e seus métodos podem ser divididos em três principais grupos: aprendizado supervisionado, não-supervisionados e por reforço. No caso das séries temporais, comumente é utilizado apenas as metodologias de aprendizado supervisionado, embora haja também aplicações dos outros métodos para determinadas tarefas (Nielsen, 2021, p. 263-264).

Como existem diversos algoritmos de aprendizado de máquina que podem ser aplicados a análise preditiva em séries temporais, foram escolhidos apenas 2 como base de estudo: As Máquinas de Vetores de Suporte (SVM, do inglês Support Vector Machine) e as Árvores Aleatórias (Random Trees).

O SVM é um algoritmo de aprendizado de máquina usado tanto para problemas de classificação como de regressão, através da Regressão por Valores de Suporte (SVR, do inglês Support Vector Regression). Ele encontra a melhor linha ou hiperplano que MELHOR separa as classes de dados. No caso do SVR, ele encontra um hiperplano que prevê valores contínuos com uma margem de tolerância mínima através de uma função de regressão.

Já a Árvore Aleatória é um algoritmo que utiliza a combinação de várias Árvores de Decisão para realizar as tarefas de classificação ou de regressão. (Bruce, 2019, p. 237). A Árvore de Decisão é um conjunto de regras "if-then-else" que, ao serem treinadas, descobrem padrões nos dados e os separam em "ramos" e "folhas" (Bruce, 2019, p. 227). Sendo assim, as Árvores Aleatórias combinam os resultados obtidos pela criação de várias Árvores de Decisão, escolhendo como resultado do seu modelo, aquele valor que mais apareceu como resultado no treinamento das diversas árvores, por exemplo.

Modelos baseados em aprendizado profundo

Devido a algumas limitações existentes nos modelos estatísticos e de aprendizado de máquina, principalmente devido aos pré-requisitos de estacionariedade dos dados, diversos modelos não-lineares, baseados nas técnicas de aprendizado profundo, como a de redes-neurais artificiais (Artificial Neural Networks), foram propostos para realizar as análises preditivas em séries temporais (Kaushik, 2020, p. 2).

Apesar de mais recentes, os modelos de aprendizado profundo têm ganhado destaque também para a tarefa de predição de dados temporais. Devido a flexibilidade de sua técnica, sua aplicação pode ser vantajosa, principalmente quando se há uma complexidade nas relações comportamentais dos dados e quando não se atende aos requisitos de linearidades exigidos pelos modelos estatísticos (Nielsen, 2021, p. 281).

Dentre os modelos de aprendizado profundo, há um destaque para as redes neurais recorrentes (RNN) que são um tipo de rede neural que processa sequências de dados, mantendo uma memória interna para capturar dependências temporais ou sequenciais.

Modelo base (nulo)

A fim de verificar se o desempenho obtido pelos modelos será significativo, os resultados serão comparados com os de um modelo base. Esse modelo consistirá na aplicação da média simples nas quantidades para prever as futuras quantidades, assim como já é feito na elaboração dos Editais de Credenciamento.

EXPERIMENTO

Origem dos dados

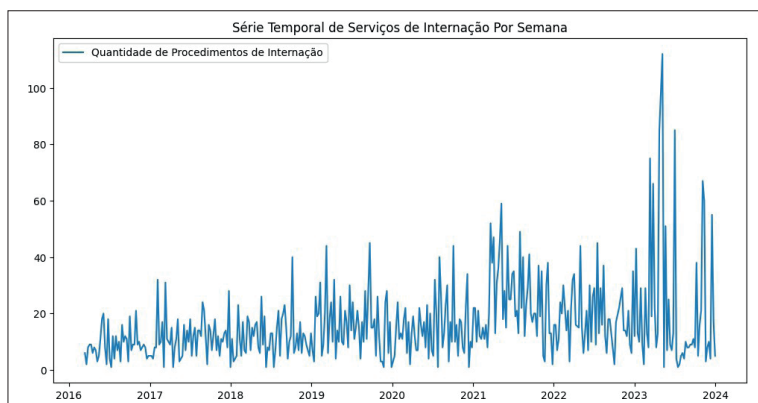
Os dados utilizados nesse estudo são provenientes do Sistema de Atendimento à Saúde Complementar (SISAUC). Esse sistema armazena todas as informações das Guias de Atendimento ao Beneficiário (GAB), documento que contém os dados da prestação dos serviços por meio de credenciamento.

Como existem diversos serviços, este trabalho limitará a avaliação da previsão das quantidades apenas do grupo de serviço de internação, visto que este grupo não sofre muito a influência de fatores organizacionais, como a falta de uma especialidade de um médico, e por representar uma grande parcela dos valores gastos no Hospital.

Os dados do SISAUC são referentes à itens de GAB, ou seja, cada linha (registro) é um procedimento que foi realizado. No registro, é possível identificar a data em que o item de GAB foi gerado. Essa será a data utilizada como base para a série temporal.

Sendo assim, após tratar os dados e selecionar as variáveis importantes para a análise, conforme demonstração presente no notebook do experimento, os dados das quantidades de serviços de internação do HFAB foram agrupados por semana conforme demonstra na visualização da Figura 1.

Figure 1 - Série Temporal de Serviços de Internação por Semana.



É possível perceber que não há um padrão aparente de variação nas quantidades e que em 2023 houve um pico de utilização dos serviços.

Metodologia de Avaliação de Resultados

Para a avaliação dos resultados apresentados pelos modelos, serão comparados os valores do coeficiente de determinação (R^2) e os valores dos erros quadráticos médios das raízes (RMSE, do inglês Root Mean Squared Error) dos modelos com os índices apresentados pelo modelo base. O R^2 verifica o quanto que o modelo explica os dados e o RMSE demonstra a precisão das previsões obtidas pelos algoritmos.

Modelagem dos Dados

O experimento realizado consistiu na aplicação dos modelos ARIMA, SVM, Random Forest e Redes Neurais Recorrentes (RNN) a fim de realizar as tarefas de predição. Após a aplicação dos modelos, foi feita uma comparação dos valores de R^2 e de RMSE dos modelos a fim de aferir o desempenho.

Aplicação de Modelos Baseados em Métodos Estatísticos

Para o primeiro treinamento, foi aplicado o modelo ARIMA. Para que o ARIMA possa ter um desempenho adequado, é necessário que os dados cumpram

algumas premissas. Uma delas é a de que os dados são estacionários, ou seja, que a média e a variância se mantem constante ao longo da série.

Para verificar a estacionariedade dos dados, foi aplicado o teste de Augmented Dickey-Fuller (ADF test). A hipótese nula deste teste é de que a série temporal não é estacionária, portanto, a hipótese alternativa é de que a série é estacionária. Os resultados obtidos pela aplicação do teste podem ser encontrados na Tabela 1.

Tabela 1 - Teste de Augmented Dickey-Fuller (ADF).

ADF Statistic	p-value
-4.14869983	0.000804

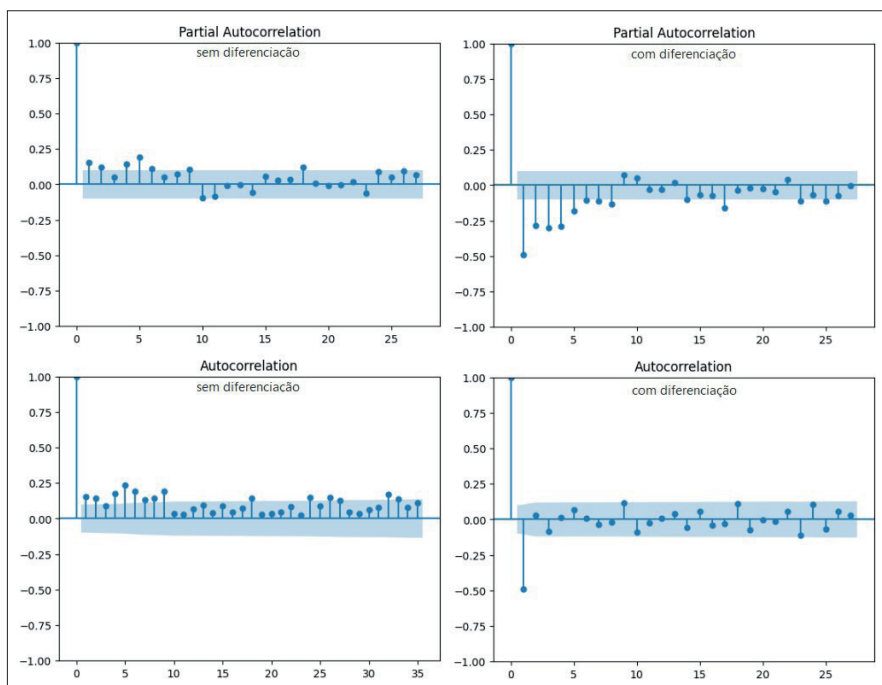
Como o nível de confiança definido foi de 0.05 e o p-valor teve um valor menor que 0.05, isso significa que a hipótese nula pode ser rejeitada e assumimos a nossa hipótese alternativa. Logo podemos afirmar que os dados são estacionários com um nível de 95% de confiança.

Uma outra premissa necessária para o ARIMA é a de que as observações podem ser previstas de acordo com as informações de observações anteriores, autocorrelação. Para isso, o gráfico de Autocorrelação (ACF) e de Autocorrelação Parcial (PACF) auxiliam nessa verificação.

Além disso, o ACF e o PACF podem ser utilizados para escolher os parâmetros do modelo ARIMA. Os gráficos de ACF e PACF podem ser visualizados na Figura 2.

A partir da análise dos gráficos, os parâmetros escolhidos foram $p=5$, $d=0$ e $q=10$ para os dados sem diferenciação. Já para os dados com diferenciação, os parâmetros seriam $p=5$, $d=1$ e $q=1$. No entanto, para definição final dos parâmetros do modelo ARIMA, foi utilizado uma busca pela combinação dos melhores parâmetros. Essa busca utilizou o critério de Akaike como fator decisivo para a escolha do modelo. Dessa forma, os parâmetros encontrados, com o menor critério de Akaike (AIC) foram $p=3$, $d=1$ e $q=1$.

Figure 2 - Gráficos do PACF e ACF do modelo, sem e com diferenciação.



Com base nisso o modelo ARIMA foi treinado e os dados gerados pelas previsões podem ser visualizados na Figura 3.

Aparentemente os dados não tiveram um ajuste muito bom no modelo original. Os valores de R^2 e de RMSE do modelo foram disponibilizados na Tabela 2.

Aplicação de Modelos Baseados em Aprendizado de Máquina

Como informado no capítulo 2, existem diversos tipos de algoritmos de aprendizado de máquina para previsão de séries temporais. Para o experimento, foram escolhidos apenas dois. Para a aplicação dos modelos, foi necessário adicionar novas colunas ao conjunto de dados para que o algoritmo tivesse mais dados para fazer sua predição.

Sendo assim, além da coluna de números de procedimento "Num. Proced." o conjunto de dados já possuía os atributos de "Valor Estimado", "Valor Final" e "Idade Beneficiário". Todas essas três colunas representam o valor médio das observações agrupadas em um período de uma semana. Além disso, foram criadas

mais 8 (oito) colunas de lags que possuíam o valor da variável dos números de procedimentos em pontos anteriores.

Figure 3 - Resultado da previsão com o modelo ARIMA.

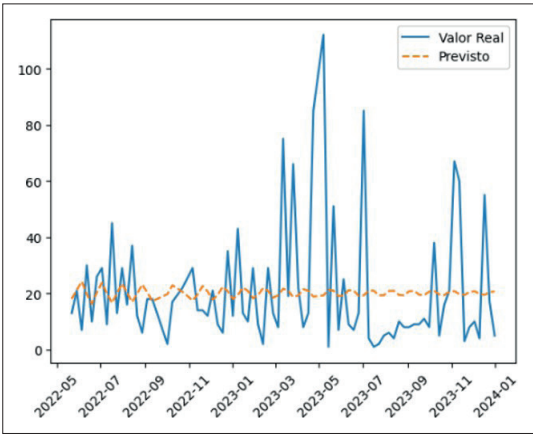


Tabela 2 - Métricas de desempenho do modelo ARIMA.

R ²	RMSE
-0.029	22.040

Aplicando a Random Forest

Dessa forma, o primeiro algoritmo a ser treinado foi a Random Forest. Os parâmetros utilizados para o treinamento do modelo foi n estimator=7, max depth=3 e os demais foram os parâmetros default. A Figura 4 demonstra os resultados da previsão do algoritmo de Random Forest.

Já os valores de R² e de RMSE da Random Foreste, foram dispostos na Tabela 3.

Aplicando o SVM

Para a aplicação do SVM, o conjunto de dados com os lags foram mantidos e o resultado do treinamento do modelo ficou conforme demonstrado na Figura 5.

Os valores do R² e do RMSE para o SVM foram disponibilizados na Tabela 4.

Figure 4 - Resultado da previsão com o modelo de Random Forest.

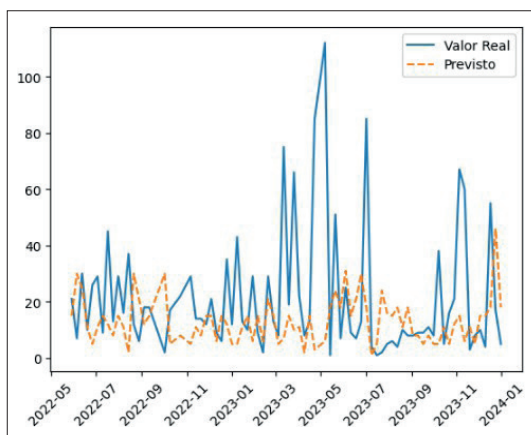


Tabela 3 - Métricas de desempenho do modelo de Random Forest.

R^2	RMSE
-0.375	25.608

Aplicação de Modelos Baseados em Aprendizado Profundo

Por fim, foi realizado o treinamento de uma Rede Neural Recorrente (RNN) a fim de realizar a tarefa de predição dos dados. A Rede foi treinada com apenas uma camada simples com 50 neurônios que utilizam a função de ativação "relu", e uma camada densa de saída com um único neurônio.

Com base nisso o algoritmo criou a seguinte previsão para os dados de teste, demonstrada na Figura 6.

E os valores de R^2 e de RMSE para o RNN seguem conforme a Tabela 5.

Figure 5 - Resultado da previsão com o modelo SVM.

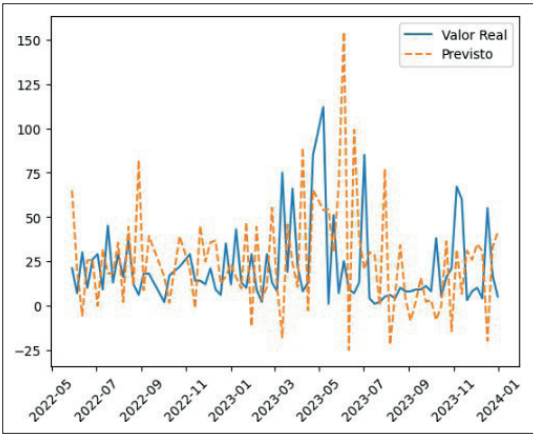


Tabela 4 - Métricas de desempenho do modelo SVM.

R ²	RMSE
-1.899	37.185

Aplicação do Modelo Base (nulo)

A previsão dos quantidades por meio do modelo base (ou nulo) é a média dos valores anteriores, conforme demonstrado na Figura 7. Dessa forma, o desempenho do modelo nulo pode ser verificado pelos valores de R² e RMSE demonstrados na Tabela 7.

RESULTADOS

Análise dos Resultados Apresentados

Com base na aplicação dos modelos e nas métricas de desempenho escolhidas, o resultado final das métricas de desempenho podem ser observados na Tabela 6.

Dentre todos os modelos avaliados, apenas um apresentaram um R² e o RMSE melhores do que o do modelo base. Esse foi o modelo ARIMA que apresentou R² = -

Figure 6 - Resultado da previsão com o modelo de RNN.

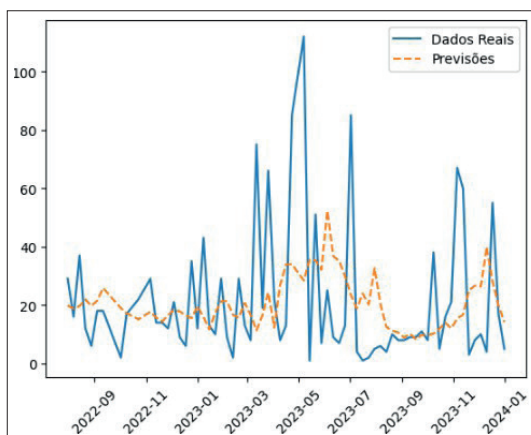


Tabela 5 - Métricas de desempenho do modelo de RNN.

R ²	RMSE
-0.069	23.576

0.029 e RMSE = 22.04, enquanto que o modelo base apresentou um $R^2 = -0.076$ e RMSE= 22.53.

O Modelo de RNN apresentou um R^2 melhor do que o do modelo nulo ($R^2 = -0.069$), porém um pior valor de RMSE (RMSE = 23.58).

Com base nesses resultados, verifica-se que o desempenho dos modelos não demonstrou ser muito superior a aplicação de média simples dos valores de momentos anteriores da série temporal. Modelos mais simples, como o ARIMA, tiveram um desempenho ligeiramente superior a modelos mais complexos como os de aprendizado de máquina e os baseados em aprendizado profundo.

Previsão da Demanda

Os Editais de Credenciamento costuma incluir a estimativa da quantidade de serviços que será utilizada em um período de um ano. Dessa maneira, o melhor modelo para poder informar qual é a quantidade ideal a ser inserida no Edital é o ARIMA. Portanto, a estimativa de quantidade para um ano foi feita com base no ARIMA e comparada com a estimativa de uma ano do modelo base, conforme apresentado na Tabela 8.

Figure 7 - Previsões do modelo nulo.

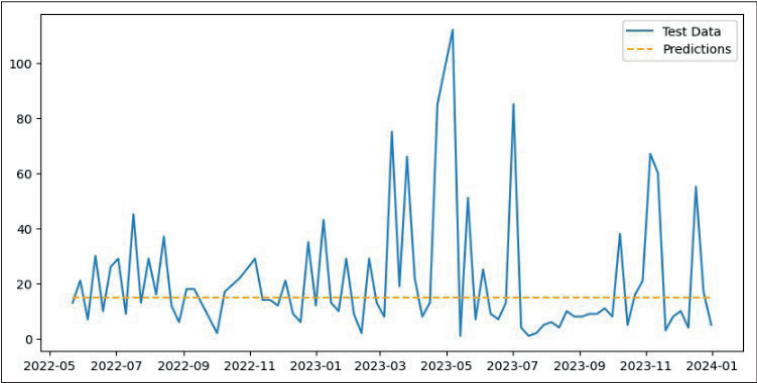


Tabela 6 - Métricas de desempenho do modelo base (nulo).

R^2	RMSE
-0.076	22.531

Pontos de Melhoria

Muitos pontos podem ser ajustados para que o projeto tenha resultados mais exitosos. Primeiramente, pode-se procurar mais atributos que auxiliem os algoritmos a melhorarem o seu desempenho e seu poder de predição. Para isso, a área técnica pode ser consultada a respeito dos dados.

Pode-se também procurar entender melhor as relações já existentes nos dados, em busca de explicações que possam contribuir para um desempenho superior, dentro do mesmo conjunto de dados.

Os parâmetros dos modelos podem ser aprimorados. Apenas para o ARIMA, foi feito uma busca pelos melhores parâmetros. Talvez por isso ele tenha apresentado uma melhor performance. Por essa razão, uma busca por parâmetros que maximizem os resultados do modelo pode alcançar melhores predições para a série temporal.

Outros algoritmos de regressão podem ser testados a fim de encontrar uma função que melhor explique a variação dos dados ao longo da série temporal. Com essa função bem definida, seria possível prever as quantidades em qualquer momento posterior da série. Além disso, a utilização de algoritmos mais simples para realizar essa tarefa de predição poderia facilitar a explicação do funcionamento do modelo e de seus resultados.

Tabela 7 - Comparação das métricas de todos OS modelos.

MODELO	R ²	RMSE
ARIMA	-0.029	22.040
Random Forest	-0.375	25.608
SVM	-1.899	37.185
RNN	-0.069	23.576
Modelo base (nulo)	-0.076	22531

Tabela 8 - Quantidades de Serviços de Internação Previstas para um Ano.

Previsão do modelo nulo	Previsão do ARIMA
843	1048

CONCLUSÃO

Após a avaliação dos resultados apresentados, pode-se concluir que o objetivo do trabalho, que é aplicar técnicas de análise preditiva em séries temporais para verificar se é possível determinar, com uma melhor taxa de acurácia, as quantidades de serviços/atendimentos a serem divulgadas nos Editais de Credenciamento, a fim de alcançar uma maior eficiência administrativa, não foi alcançado com êxito, pois os modelos não apresentam uma acurácia significativamente superior ao modelo nulo (base).

Sendo assim, é possível realizar diversas melhorias que contribuam para o alcance do objetivo proposto, como a engenharia de características (feature engineering), a busca por melhores parâmetros, um entendimento melhor do conjunto de dados e a aplicações de outras metodologias e algoritmos para a realização das predições.

Por esse motivo, a sugestão para trabalhos futuros é realizar as melhorias sugeridas neste trabalho a fim de buscar melhores desempenhos. Pode-se também procurar realizar a predição de outras variáveis como os valores totais gastos por item de serviço no HFAB ou em outros Hospitais.

REFERÊNCIAS

- BADAWY, Mohammed; RAMADAN, Nagy; HEFNY, Hesham Ahmed. **Healthcare pre-dictive analytics using machine learning and deep learning techniques: a survey**. Journal of Electrical Systems and Information Technology. 2023.
- BRUCE, Peter; BRUCE, Andrew. **Estatística Prática para Cientista de Dados: 50 conceitos essenciais**. Alta Books. 2019.
- CAO, Rui; Wang, Qiao. **An Evaluation of Standard Statistical Models and LLMs on Time Series Forecasting**. IEEE. 2024.
- CRISTIAN, Rares *et al.* **Inter-Series Transformer: Attending to Products in Time Series Forecasting**. International Journal of Forecasting. 2024.
- DESORDI, Danubia; BONA, Carla Della. **A Inteligência Artificial e a Eficiência na Administração Pública**. Viçosa: Revista de Direito. 2020.
- HYNDMAN, Rob J.; ATHANASOPOULOS, George. **Forecasting: Principles and Practice**. Australia: Monash University. 3º Ed. 2021.
- KAUSHIK, Shruti *et al.* **AI in Healthcare: Time-Series Forecasting Using Statistical, Neural, and Ensemble Architectures**. Frontiers in Big Data. 2020.
- MASINI, Ricardo P.; MEDEIROS, Marcelo C.; MENDES, Eduardo F. **Machine Learning Advances for Time Series Forecasting**. 2021.
- NIELSEN, Aileen. **Análise Prática de Séries Temporais: Predição com Estatística e Aprendizado de Máquina**. Alta Books. 2021.
- PANIS, Amanda da Cunha. **Inovação em Compras Públicas: Estudo de Caso do Robô Alice da Controladoria-Geral da União (CGU)**. Brasília: Universidade de Brasília. 2020.
- VERAS, Karina de Oliveira; BARRETO, Gabriela. **A Inteligência Artificial no Setor Público: Uma análise do Projeto Victor no Poder Judiciário**. São Paulo: IX Encontro Brasileiro de Administração Pública. 2022.

DETECTION OF ANOMALOUS VEHICLE INTERACTIONS ON HIGHWAYS: TOWARDS AN AUTOMATED METHOD FOR APPLICATION ON COMPUTER VISION DATA OBTAINED FROM TRAFFIC CAMERAS

Guilherme Horbatink Villanova

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

Elcio Hideiti Shiguemori

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

ABSTRACT

This paper presents an automated method for identifying and cataloging interactions between vehicles on two-lane, two-way roadways using computer vision data captured from traffic cameras. Following a brief description and exploratory analysis of the original data, we outline a step-by-step process: from classifying vehicle directions to segregating lanes for same-direction vehicles. We discuss the specifics of each step, including the handling of outliers. Finally, a comprehensive listing of interactions between vehicles in the same lane is provided, along with a use case demonstrating the identification of anomalous vehicle interactions derived from this listing.

INTRODUCTION

Context

In 2023, Brazilian highways registered 5,605 deaths resulting from traffic accidents. There were 67,620 accidents of which 56,666 resulted in casualties (injuries or death) [CNT 2023]. This paints a worrisome picture which calls for new and more elaborate ways of traffic surveillance in order to try to minimize the risks posed by driving a vehicle on Brazilian roads and streets.

One of the tools used in initiatives to combat this issue is remote sensing. In the context of road monitoring the main source of sensing material is footage from road monitoring cameras and drone captured footage. From this footage a myriad of processes can take place: tracking processes can be applied to identify the trajectories of each vehicle shown in the footage, anomaly detection techniques or trajectory-based analysis can follow, etc. [Santosh *et al.* 2020].

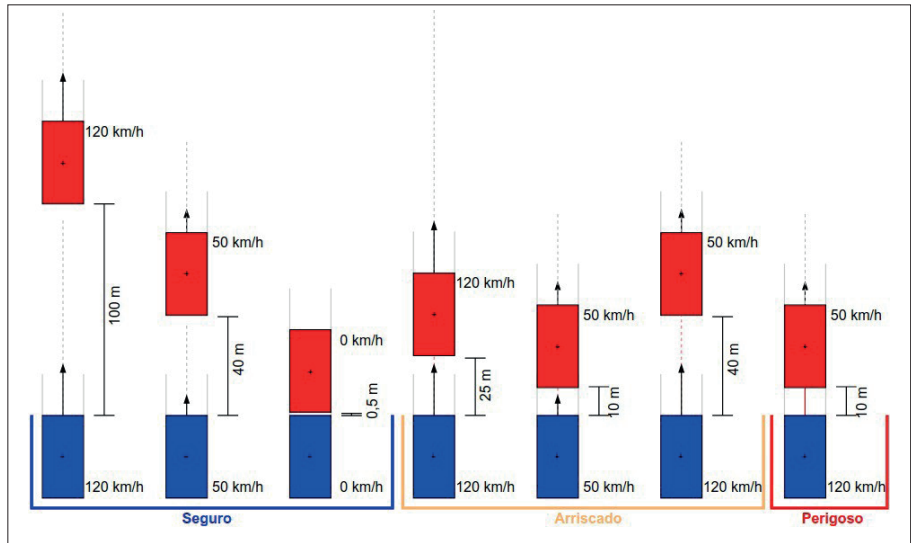
This paper is an expansion upon [Andrade 2022], in which computer vision techniques were applied to more than 30000 video frames captured by drones on the highways of São Paulo in order to produce an application to detect driver behaviors. The vehicle detection on the footage was done with YOLOv4, the tracking with the Deep SORT algorithm and the production of behavioural profiles was a result of the classification of the displacements as normal or abnormal by LSTM (long-short term memories) and MLP (multi layer perceptrons) networks [Andrade, 2022].

On Andrade's research, the behaviours were judged using a Gaussian function, exclusive to each dataset formed by each roadway portion, to classify the vehicles time-series, obtained by the succession of the tracking boxes for an specific identified, ID based on a combination of it's average speed, distance traveled and trajectory smoothness as normal or abnormal. With these classifications, the neural networks were trained to identify anomalous behaviours. Andrade's results were deemed by the author as not robust considering the dynamic scenarios of each roadway portion (p. 113).

In the wake of the non robustness of said line of inquiry, we pick up a thread left by Andrade [2022]: another dimension in which to classify behaviors is the proximity between vehicles. Two vehicles close each other does not

necessarily means danger, but the higher the speed they are going the greater the distance must be to ensure safety, as the drivers are not ready for obstacles on the road such as an abrupt stop by another vehicle. [Čulík *et al.*, 2022] This is illustrated on Figure 1.

Figure 1 - Illustration of the "Proximity Question". From Left to Right: "Safe", "Risky" and "Dangerous".
Reproduction of Andrade [2022].



Our main goal will be to produce a method for automatizing the process of obtaining and analyzing interactions between vehicles on a given portion of a highway. This is a qualitative different approach: we'll need not only to keep track of each individual vehicle on every frame, but also of the interaction between vehicles that are active on the same frame.

Research Problem

The main problem that will be dealt with on this research project relates to the feasibility to automate the process of obtaining the interactions between two vehicles on a given portion of a highway and to classify those interactions: **"How to fully automate the steps between obtaining the output of a vehicle tracking process on a highway and providing classification of interactions between vehicles?"**.

Main Objectives

The main objective of this research is, thereafter, to automate the process of detecting anomalous interactions between two vehicles on a highway on the context of continuing the research done by Andrade [2022].

Specific Objectives

They are as follow:

- Evaluate the data provided by the tracking process present on Andrade [2022] and perform an exploratory analysis of it.
- Setup process and steps of data processing that result on the listing of interactions between vehicles on a given portion of a highway.
- Produce behaviour dimensions on which the interactions between the vehicles could be analyzed.
- Application of the process to the available data.
- Analysis of the results of the previous step.
- Comprehensive overview of the process, final evaluation and road map of further work.

DEVELOPMENT

The Data

The data is the result, as previously discussed, of the efforts conducted by Andrade [2022]. On it, we find the complete work from footage capture, video processing, data preparation for the application of the chosen algorithm of Computer Vision (Yolo v4), the full processing of training, test and application of the Convolutional Neural Networks to tracking, labeling and classification of the vehicles captured on the video footage. After, there's the processing required to form the tabular data previously mentioned.

The full data provided and to be used on this paper's efforts is:

- Six Tabular Data labeled as the highway's specific kilometer of where the original footage was recorded and of which of the tabular data is a result of the tracking process applied to it.
- Folders labeled the same as the six previously discussed tabular data with individual files for each vehicle listed uniquely labeled on the complete data.
- Six AVI video files with the corresponding footage from which the tracking process was conducted.

The video footage analyzed by the algorithm was 10 minutes long compressed to 15 Frames per Second. The original footage was captured in 60 FPS.

Our work, here, will be mainly on the Tabular Data, since it contains the full information of the tracking and will allow us to proceed with the breakdown of interactions between to vehicles.

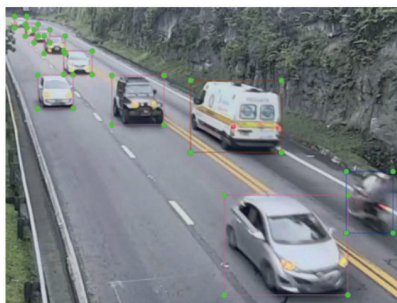
The Main Data Source: A Breakdown

The Columns are as follows:

- **Frame:** Indicates the frame in which the box for each vehicle, represented by the coordinates X (Min and Max) and Y (Min and Max), was identified. The dataset is ordered firstly by this column.
- **ID:** The assigned ID for that specific vehicle. The tracking method is able to keep track of how boxes of different frames are products of the same entity (in this case a vehicle). Without this identification process our analysis wouldn't be possible. The Tabular Data is ordered secondly by this column.
- **Class:** Each specific ID is labeled as an specific class such as Car, Pickup, Motorcycle, etc. It would be checked on the exploratory analysis if each ID is always labeled as the same class.
- **X:** The tracking is made on a 2D plane. This value refers to the horizontal portion of the coordinate that marks the central point of the tracking box of that vehicle on said frame. Variations on this value, as well as on the Y-dependent variables, must correspond 1:1 to pixels on the original footage processed by the computational vision algorithm.

- **Y:** The vertical portion of the central point coordinate on the 2D plane.
- **Xmin:** The tracking box, as the name suggests, is defined by four limits. This value marks the "left" limit.
- **Ymin:** The "bottom" limit.
- **Xmax:** The "right" limit.
- **Ymax:** The "top" limit.

Figure 2 - Example of Tracking Instance, Andrade [2022].



CLASSE	PONTO MÁXIMO	PONTO MÍNIMO
CARRO	(0.763672, 0.752083)	(0.196094, 0.281944)
MOTOCICLETA	(0.896484, 0.631250)	(0.074219, 0.173611)
VAN	(0.683984, 0.388889)	(0.144531, 0.202778)
CARRO	(0.528125, 0.343056)	(0.082812, 0.138889)
CARRO	(0.396875, 0.319444)	(0.056250, 0.094444)
CARRO	(0.432812, 0.237500)	(0.045312, 0.066667)
CARRO	(0.395703, 0.186806)	(0.033594, 0.048611)
CARRO	(0.375781, 0.162500)	(0.025000, 0.036111)
CARRO	(0.359375, 0.145833)	(0.020313, 0.033333)
CARRO	(0.341797, 0.122917)	(0.019531, 0.031944)

Exploratory Data Analysis

The Exploratory Data Analysis here will be based on the process delineated in [Mount and Zume 2019]. The outlined model is characterized as a way to spot problems with the data that can be conducted by either producing summary statistics or graphics and another visualization techniques.

Preliminary Context

Data Integrity Checks

One of the cornerstones of Exploratory Data Analysis is to check for missing values on the dataset. If there are, a method of dealing with them must be chosen and applied [Mount and Zume 2019, p. 55]. All the Six Tabular Data Files we'll be working with are exempt from missing values.

Another Data Integrity check is if each specific ID is labeled always as the same class. If this is not the case, we'll have an internal inconsistency problem

that, just as with the hypothetical NaNs. After checking every ID for each of the six files: class is always consistent within the context of each individual ID.

One last check will be to see if there are any vehicles that appear only on one frame. For our intentions and purposes here, a one-frame-vehicle would be an entity from which we can't derive any analysis from and would not work to form an interaction with another vehicle. There are no such cases where this happens across the 6 datasets.

An Analysis of the road portions

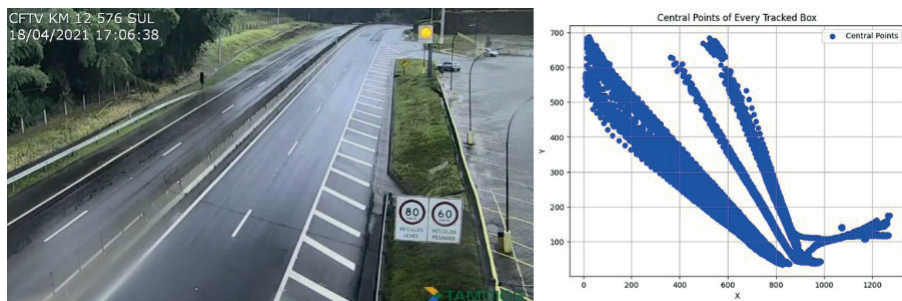
On this section we'll take a look at each full set Tabular Data.

T12: As we can see from the frame on Figure 3, the road has two-way traffic. On each way it has two lanes. This two considerations, as we shall see, will be central points of analysis and involve careful consideration on the implementation front.

When we look to the plot of the coordinates, we can clearly see the outline of the two lanes of the right portion (assuming the orientation we have when are looking at the picture) of the highway. This could be the function of smaller traffic and less to none vehicles trafficking on the lane line. Since the road-monitoring camera is positioned on the "right side" the lanes on this way closer to the camera will occupy a bigger portion of the frame and this could also contribute with a bigger "clarity" on this portion.

The rightmost part of the picture must as well be a object of considerations. When we go back to Andrade [2022], we'll see that on this portion we find a restaurant and a gas station [Andrade 2022]. This can be an important source of outlier vehicle behavior. From this frame we already see two stationed vehicles. When we watch the complete footage, we can see that there's even a car that is going on right lane of the right portion and turns into the gas station and makes its way down the frame.

Figure 3 - A frame of the analyzed footage of KM 12 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.



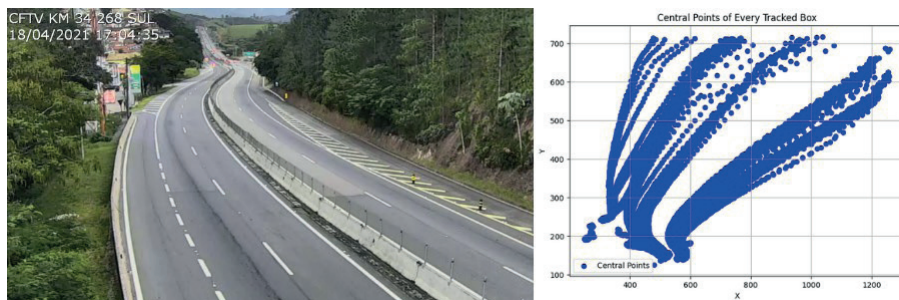
T20: Here (as we can see on Figure 4) we also have two way traffic and four lanes, two on each side. When we look to the coordinate plot we see the 4 lanes somewhat more clearly than on T12. We also see three clear outliers: the two clusters of points on the bottom of the plot and what it looks like a single vehicle making it's way on the highway's shoulder. When we look at the footage it is actually a bicycle going that way.

T34: Here (Figure 5) we come in contact with an previously unseen complexity. Figure 4 shows us, like T12 and T20, two ways and four lanes. When we look to the coordinates plot, though, it is not as quick of a visual examination to understand the lane placement. The big gap on the middle can be used to determine visually which way is which, but on the left portion of both the footage and the plot we have what would "normally look like" both "left side" lanes but also another lines. This will become a point of greater analysis on the following sections of analysis. We also see an exist to the left side of the frame. Accordingly to [Andrade 2022] description, it is a gas station as well as the exit into a residential neighbourhood.

Figure 4 - A frame of the analyzed footage of KM 20 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.

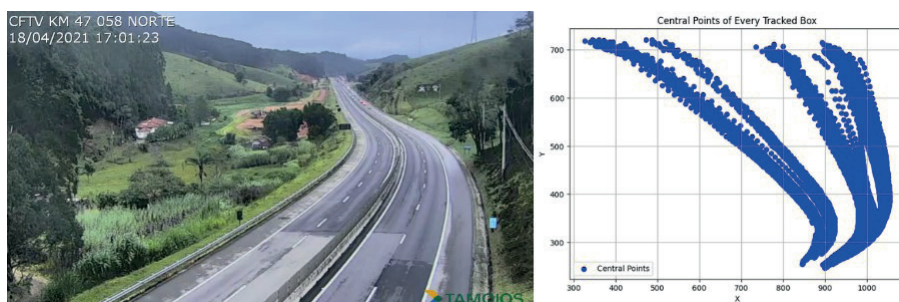


Figure 5 - A frame of the analyzed footage of KM 34 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.



T47: The Figure 6 shows us, when compared to the previous subjects of analysis, a quite elongated roadway portion. We can see the four lanes on the X and Y plot and there not obvious, visually discernible outlying points.

Figure 6 - A frame of the analyzed footage of KM 47 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.



T58: T58 (Figure 7 is has the most exits out of any of the analyzed road portions.

Ont he X and Y plot we see clear outliers. The way the camera is positioned makes the central barrier between the ways of the road more clearly discernible on the plot.

Figure 7 - A frame of the analyzed footage of KM 58 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.



T80: All the other five road portions that will be analyzed have four lanes: two on each way of the highway. On T80 (Figure 8), though, we have a different situation. Although we still have two ways and the "left way" has two lanes, the "right" (frame orientation) one has only one lane. This shows us a glimpse of the diversity of situations that any automation attempt must be ready to encounter.

Figure 8 - A frame of the analyzed footage of KM 80 Rodovia Tamoios and a plot displaying every X and Y coordinates on that portion's Tabular Data.



Distribution of Vehicle Types

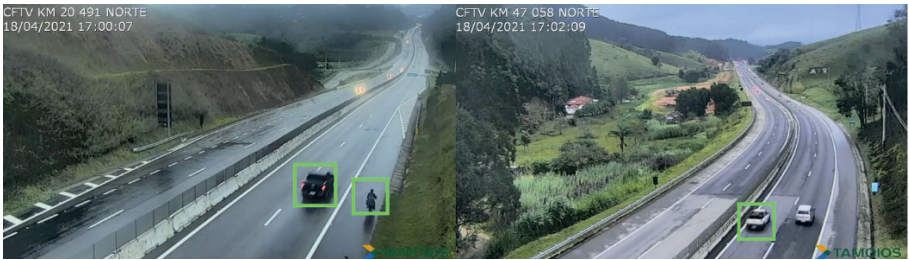
On the 6 datasets we'll encounter seven different types of vehicles on the "Class" column: Car, Motorcycle, Pickup, Bicycle, Van, Truck and Bus. Some examples are illustrated on Figure 9. On Table 1 we have a breakdown of each class prevalence, as per unique IDs, for each of the datasets. Close to 90% of the IDs are classified as cars. The class 'Bicycle' that must go through special considerations. Accordingly to the Article 58 of the Brazilian Traffic Code, the cyclist must always make its way through the shoulder of the highway when possible [CTB 2021].

When there's no highway shoulder, it must traverse on the right board of the lane. This will make it so that when there's a Bicycle on the frame it will be an "outlier": either by going on the shoulders or boards of the lane or by dangerously making it's way through the middle of the road as if it was an automobile.

Table 1 - Vehicle Class Distribution.

Tabular Data	Car	Motorcycle	Pickup	Bicycle	Van	Truck	Bus
T12	386	6	9	4	3	3	2
T20	547	8	40	1	0	11	0
T34	380	9	14	0	2	2	0
T47	446	6	4	0	0	0	0
T58	276	4	3	0	0	3	0
T80	302	17	6	1	4	4	4

Figure 9 - Example of classification, from left to right: Car, Bicylc and Picukp. Boxes with illustration purposes, does not match Tracking Box.



A preliminary analysis of Speed

Speed

Speed, here, will be the function of the euclidean distance between the first and the last detected tracking box of an specific ID and how many frames there were between these two instances. This approach is the one adopted on Andrade [2022].

What we'll be left with is a value that represents the function: "pixels traveled by frames". This produces a number that represents something that is relative to the camera placement on the original footage captured. Assuming a constant definition output (X by Y pixels) for the capture device, the "amount of area" covered on the screen is a function of the angle of the camera in relation to

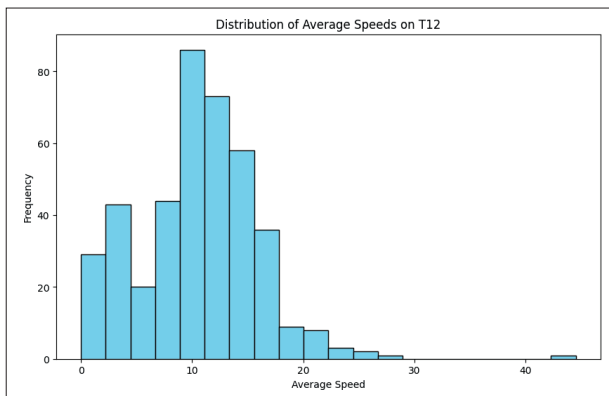
the what's being captured as well as the distance between the objects and the camera lens. As long as the camera stays fixed, we'll have an internally consistent distance-to-pixel ratio since the path, the road, will not move.

As we can see on the plot on Figure 10, the distribution of average speeds on T12 has the outlining of a Gaussian distribution centered around 10 pixels/frame.

The center of the tracking box

For the continuation of this processes, the option will be for utilizing the center of the tracking boxes, defined by the X and Y values and not the full tracking box. This decision is based on the fact, as discussed on Andrade [2022], a vehicle can obscure another when they get close together on the highway. When this happens, we would have a distance of 0 between the boxes. The use of the center of the boxes permits the maintenance of the value of distance between two vehicles and it also allows us to perform the following operations with points and not areas.

Figure 10 - Distribution of Average Speed on T12.



Orientation

We have seen that on each of the six road portions from which our datasets derive it's information from we have instances of double way trafficking. This poses a fundamental problem on our endeavors of identifying anomalous interactions: How do identify on what way each vehicle is going?

This will be our first endeavor of active data transformation and we'll propose a criteria for identifying both the vehicles going on each direction as well as outliers that appear to be stopped or going on anomalous patterns.

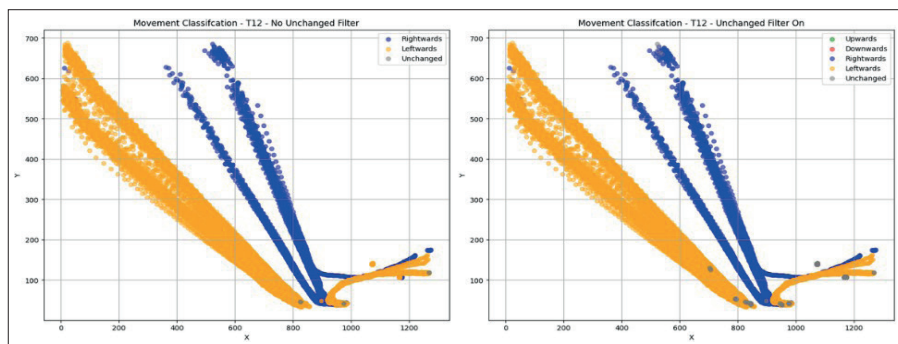
The process will follow this logic:

- Check to see if there's a bigger mean variation in the ID's routes between the horizontal and vertical axis.
- Establish the axis on which the separation will occur. If it is based on the dislocation on the horizontal axis (X), the ways will be labeled as either Left or Right. If it will be based on the Y variation, we'll have either an 'upwards' or 'downwards' labelling. If there's no movement, the label will stay as 'Unchanged' as it is the value when initialized.
- We'll take the First and Last frame of each ID and compare it using the preferred axis established on the previous step. Thus will provide us with a label for that ID.

Figure 11 shows us examples of the process sorting in relation to X or Y axis.

When we ran this on the process on the T12, which had the vehicles stopped on the roadside gas station, we noticed that the ID wasn't being labeled as an "unchanged" case. When we checked we found that the tracking attributes varying 1 pixel change of its X and Y coordinates. So we ran the process again, now adding the condition: If the euclidean distance between coordinates of the IDs first and last frame is less than 10, may it remain classified as unchanged. Now that and other cases of non-movement were caught as outliers vehicles and will be excluded from the analysis process.

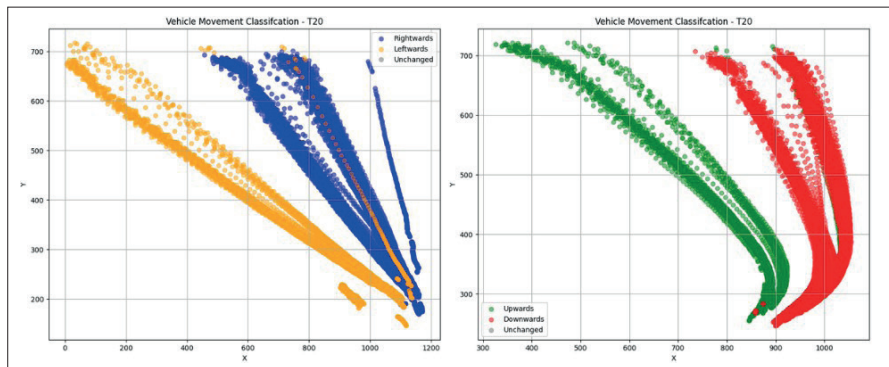
Figure 11 - T12 Classification Without and With the 'Unchanged Filter' On.



The remaining data is split in two: one for each way. The process conducted to the data in discussion here, resulted in:

- **T12:** Leftwards (T12 L) and Rightwards (T12 R)
- **T20:** Leftwards (T20 L) and Rightwards (T20 R)
- **T34:** Leftwards (T34 L) and Rightwards (T34 R)
- **T47:** Downwards (T47 D) and Upwards (T47 U)
- **T58:** Leftwards (T58 L) and Rightwards (T58 R)
- **T80:** Leftwards (T80 L) and Rightwards (T80 R). T80 L is the 'one-lane' portion.

Figure 12 - T20 and T47 after the process.



As we can see on Figure 12, we already have outliers on this process, as there's a vehicle on an rightwards trajectory on the 'leftwards lane'. How to deal with cases like this will be discussed on the next section.

Lane Segregation

This section will constitute the bulk of the work and 'heavy lifting' done by our data processing method. If we seek to establish a method for identifying anomalous interactions between two vehicles, having a reliable way to establish the succeeding process that takes place on a particular lane seems essential. After all, the most basic process of interaction of two vehicles on a highway is the one of the "barrier" posed by the vehicle in front of another. This is what is

illustrated in Figure 1. Since one of our main goal here is to produce a method that is as universal as possible and ready to work on small road portions/camera definitions, we'll give preference to this analysis.

Clustering: method considerations

The quintessential problem Clustering Analysis is trying to respond to is: determining the organization of grouped data when only the observed values are known and no additional information is provided [Fraley and Raftery 1998]. This is exactly the challenge we face when trying to automatically establish on which lane a vehicle is driving based on tracking data of a Computer Vision process. Considering the goals of maximum universalizing and automation of the process that is at the core of this paper, we must search for solutions that are both easily implemented and easily iterated. The product of this paper will not be final: the real-life phenomenon we are dealing with is immensely complex and there's no one-final solution to it.

In light of this context, we side with [Stančin and Jović 2019][p. 977]: "The use of Python in the area of data Science has reached unprecedented levels, especially in the area of freely available tools and libraries". Within Python we find the Scikit-learn library, "the most popular Python library for machine learning"[Stančin and Jović 2019]. On it's original paper [Pedregosa *et al.*, 2011], Scikit-learn focus is promoted as not providing the most number of features, but provide them solidly implemented.

On Sci-kit learn we find, among others, two clustering methods solidly implemented: K-Means and DBScan. K-Means is one of the most simple and popular algorithms, as well as highly scalable [bin Waheed *et al.*, 2019]. One of the problems of the algorithm is that the number of clusters must be inputted "beforehand", which is most cases is an unknown variable [bin Waheed *et al.* 2019]. This will not be the case for many of our use applications of the algorithm. So we are picking one of, if not the, most popular and simple clustering method and having not to deal with its biggest downside.

DBScan is selected as the other option for, in addition to also being a popular [bin Waheed *et al.*, 2019, p.5110], covering grounds that K-Means does not. In the original paper where DBSCAN is proposed, [Ester *et al.*, 1996], DBScan thrives

in identifying outliers, one of the main concerns of the work being done here on this paper, non-linearity and does not need the input of number of clusters [bin Waheed *et al.*, 2019, p. 5110].

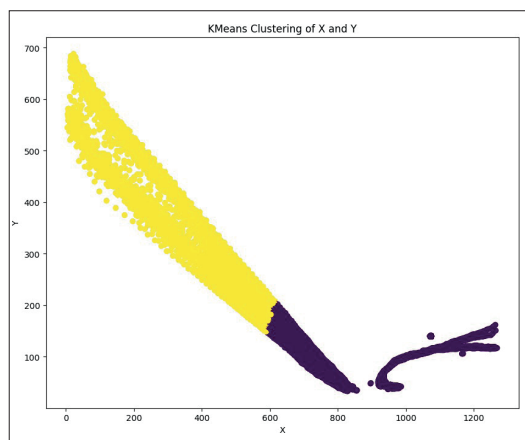
Before we move forward there is a fundamental decision to be made. What we are hoping to achieve is the classification of every single "valid" (non-outlying) ID in a lane, so that we can establish the interactions between it and the other vehicles on the same lane. For each ID we have a collection of points (X and Y coordinates) related to the center of the tracking box on said frame. So there's three ways to go about which lane is vehicle X on. It could be one answer for every active frame, e.g. "on frame X, ID Y was on the left lane, on frame Z it was on the right lane", and then we would have a continuum ranging from 'leftness' to 'rightness' in which to position each ID, e.g. "ID Y was 90% on the left and 10% on the right lane". We could establish that the car is either on the left or right if 100% of it's frame classifications are of the same label or that it is on an changing lane state if it has frames with both classifications. Lastly, we can maintain the one classification per ID approach but drop the 'purity' clause for the labels and establish classifications thresholds: "if more than 50% of the individuals boxes of an ID are classified as one specific lane, the ID will receive this label ID".

Since we want to provide an interaction list, the continuum approach is not work-able as the vehicles could not be identified as of one lane from which a succession order can be established. So we must go with one-ID-one-label approaches. The problem with the second approach is that it is likely to exclude a relevant fraction of the IDs, especially since near the vanishing point it will be harder to establish that ID's lane. So we are opting for the third approach and we'll be discussing on the upcoming sections how to setup the thresholds.

Problems With the Simplest Approach

So we established that we'll be using K-means and on K-means we can set the numbers of clusters that we want to divide our collection of data into. So what happens if we just apply K-means clustering with $k = 2$ on one of our single-way datasets?

Figure 13 - Ex.of K-Means applied to all of Way Coordinates.



As we can see on Figure 13, when we apply K-means ($k=2$) to the full set of values, what we get is a halfway point where the “average vehicle” would have produced half of its individuals tracking boxes: the vanishing point of the picture frame of the recording device will produce an effect of the objects staying longer on frame as a function of how far they are from the placement of the camera. We can also see that the rightmost part of the plot, which is a clear outlier was treated as a normal “clusterable” aggregate of points.

And since we can't set the numbers of clusters on DB Scan, it's clearly not a robust solution to be implemented here.

Outlier Removal

If the DBSCAN approach was not to be productive on the last section, we can use its outlying-detection-abilities now. We noticed that there were clear outliers on the Figure 13 plot. When applying DBSCAN, areas on the data map with a high concentration of points signify a cluster, whereas the ones with a low concentration suggest outliers or noise. [bin Waheed *et al.* 2019]. We can apply it to the data resulting from the previous orientation-segregation step, establish a main cluster on which the majority of points are located.

DBSCAN has several 'internal parameters' we can tweak in order to select for different results. The main two, the ones mainly discussed on its original publication, are the Eps-neighborhood of a point and the minimum number

of points required to form a cluster. Eps-neighborhood defines the maximum distance between two points to be considered neighbors. Low "eps" values will result in a greater number of clusters, as only points close together can be classified as being of the same cluster. The minimum number of points provides the baseline for a cluster size, there can be no clusters smaller than this set value. So with lower eps and minimum number values, there's a proliferation of clusters, and with high values for both parameters we see bigger and "less cohesive" clusters.

But we also have another two options on this step for 'filtering out' the outliers: We can consider an outlier and ID which has any point outside the biggest cluster or if all of its coordinates lie on 'secondary clusters'. It's a clear trade-off situation: we'll have a "clearer" remaining dataset with the first approach, but we might exclude a vehicle which could be in fact interacting on one of the main lanes of the highway.

Having run both approaches for every dataset and experimenting with higher and lower values for eps and "minimum values", let's expand on the trade-off considerations with the help of some visual aid.

- On Figure 14 we can see the problems of setting the parameters too high or too low. On the left side, the eps value was set to 30 and the minimum value set to 10. On the right side they were set to 10 and 1.
- Trajectories that on a visual analysis looked like they were "entirely" on 'secondary clusters' actually had an instance of a point on the majority cluster. This goes on the direction of the point we made previously: things get really fuzzy on the vanishing point. See Figure.
- The noise and non exclusion of IDs with any points on the 'secondary clusters' can create many cases of noise retention as we can see on Figure.

The most-exclusionary criteria is by no means perfect, it can exclude too much or even too little still, but it seems to be a more robust approach, especially keeping the interaction listing on mind. The data that 'does not make the cut' here will not simply be lost, but will be labeled as an outlier, which, from our analysis, seems to be a more productive approach than to have it take part in further steps towards the interactions listing.

As for the parameters, after testing for $\text{eps} = 10, 20$ or 30 and minimum value $1, 5$ or 10 , the middle ground made most sense. The parameters were so to: $\text{Eps} = 20$ and Minimum value $= 5$.

Figure 14 - Problem with too high values for the parameters and too low.

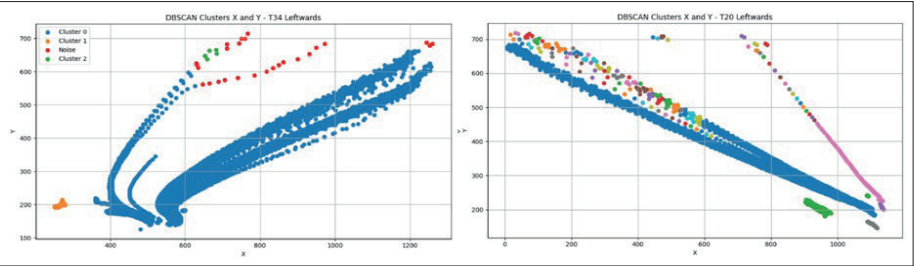
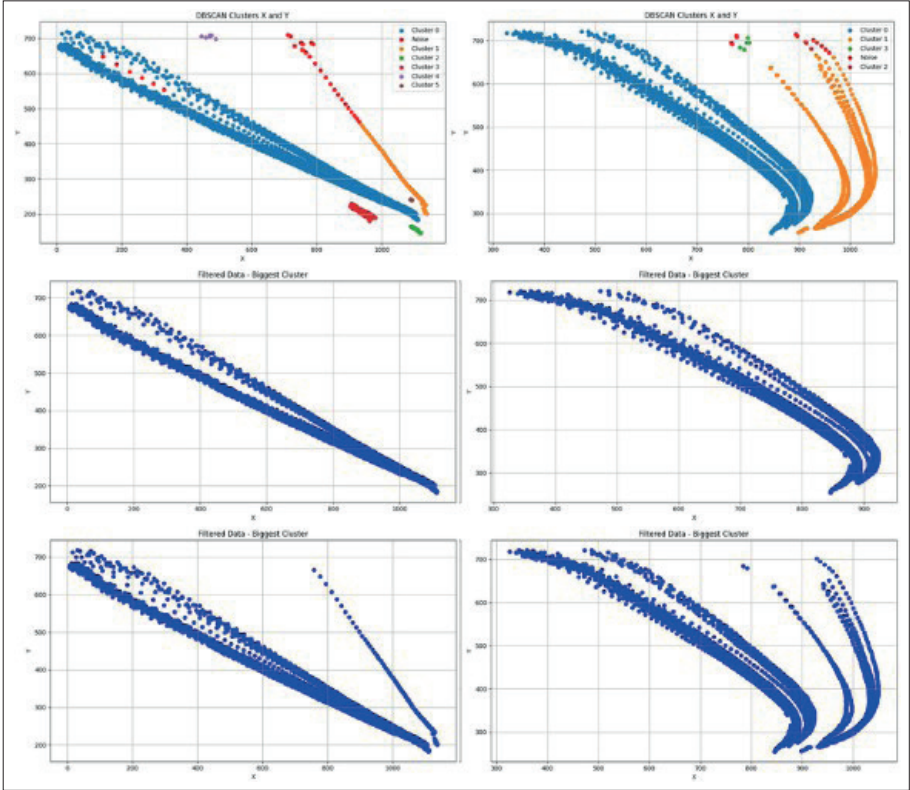


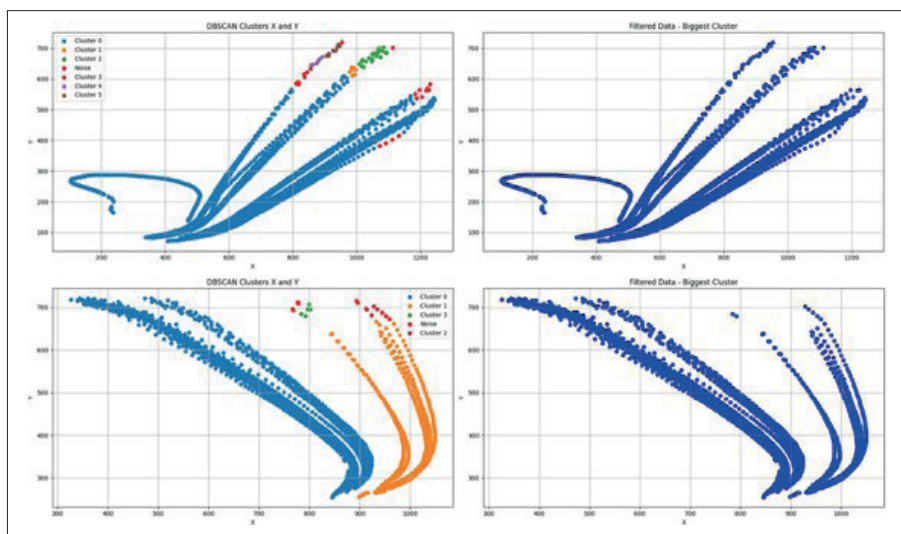
Figure 15 - When "clear outliers" have Main-Cluster instances.



Finding the Dividing Linear Function

Even if K-Means and DBSCAN can't reliably identify the two lanes on the full dataset from the full plot of the coordinates of selected non-outlier IDs, this does not mean that we can't use them. Returning and expanding on the observation that there are differences between the concentration of points depending on how close or far away that portion of the screen is from the camera, we can postulate that: the boxes on the edges of the screen are either the moment that the vehicle enters the screen or the moment it leaves.

Figure 16 - When not adopting the most exclusionary criteria leaves too much noise.



This observation, as trivial as it is, will open automation possibilities. Take Figure 17 where we plot T12 Leftwards' IDs First and Last Frames. We can see that where the vehicles entered the screen, the First frames, since it was further away from the camera, there's a lesser degree of dispersion of the coordinates in the 2D space. Where they left, Last Frames, as it was closer to the camera, there's a greater degree of spacial distribution.

What we propose is as follows: we want a straight line that would divide the two lanes we see on the plane. This line would "cut" the plane where the data is contained dividing the coordinate points into three separate categories. We would have points on one side of the line, points on the another side and the points

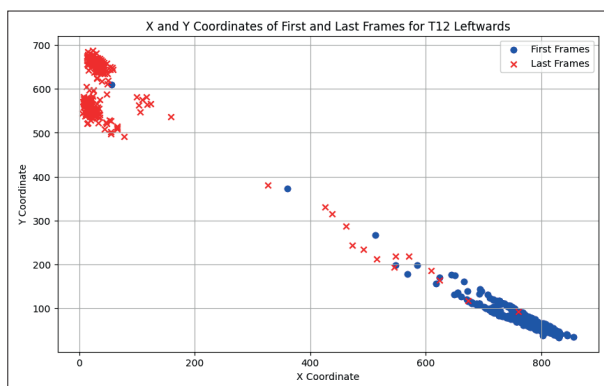
exactly on it. With this new data, we could, then, carry out our previously defined approach and classify the vehicles as being part of one or another lane.

The "line" we want here is synonym with an a linear function. If we have two points in space we would already have enough to produce one. The question now be-comes: how to establish these two points?

We'll always have, as we previously established, differences in point-coor-dinates dispersion between the group of first and the group of last frames. As we see on Figure 17, because of this, when we consider a plot of X and Y coordinates for the first and last frames of the vehicles, we'll see three clusters emerge: Two for moment of greater dispersion (either the group of first or last frames) and one for the moment of least.

Another thing we must ask when looking to the plot of Figure 17: why there are points on either the start or the end of a vehicle's trajectory on the middle portion of the screen? This a technical consideration that must be dealt with in the Computer Vision processing of the images. But since we have these points, and we must have a way to deal with them assuming that the technology behind the computer vision processes won't be perfect.

Figure 17 - Plot of the First and Last Coordinates T12 Leftwards.



We must return to the main goal of this paper as a north: what we want is to provide a full listing of interactions between same-lane vehicles on a highway. Assuming that what happened was either a mislabeling of a already identified vehicle as a new one or the first ID'ing of a previously untracked vehicle and not the mislabelling of a non-entity as an object of concern, we must chose to

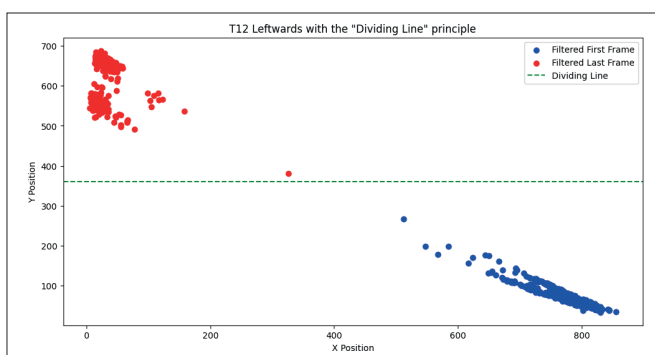
maintain these instances in the dataset. They can incur on relevant interactions with another vehicles, especially if we assume that this mislabelling can be the result of an anomalous behavior to begin with.

This problem is the direct result of the outliers regarding the first and last frame grouping we discussed two paragraphs ago. What we are trying to determine here is the linear function that would divide the plane and "represent" the diving line across both lanes. We don't want to drop these IDs from the analysis. We could implement a process to temporally suppress them in order to be able to form successfully the three clusters we would like to obtain. The following process will be adopted: we get the maximum and minimum value of Y for the first and last frames, we set a "dividing line" on the halfway point across and we select and operate with only the points located on the majority which has a bigger N. We can see this at work on Figure 18.

Once we have the "treated data" we can set to establish the linear function that will act as the conduit for the lane segregation. It will follow these steps:

- Identify the center of the clusters located on the "most disperse" area.
- Establish the half-way point between the two. (Point of Interest 1)
- Identify the center of the cluster on which the points of the "least disperse" area are located. (Point of Interest 2)
- Establish the linear function that contains both Pol 1 and Pol 2.

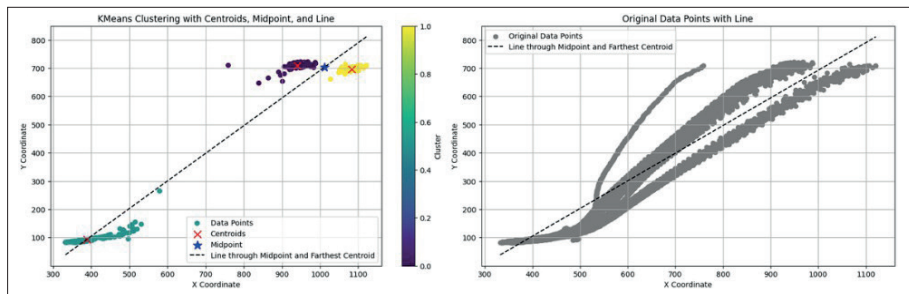
Figure 18 - T12 Leftwards when the dividing line is applied.



The process is illustrated on Figure 19. The linear function correctly splits the plane in two given the conditions provided, but we can see why this solution is not reliable as it is: it would only work for a straight path or if there was a large enough gap between the two lines.

We also tried establishing a process of reliably identifying one lane situations, as is the case with T80 Leftwards. We tried running DBSCAN on the resulting data after the lane diving processes, but the process was sensible to the points that remained close to the line. We must opt for excluding the dataset from further analysis. With this we restrict the application of the method only for highways ways that have two lanes.

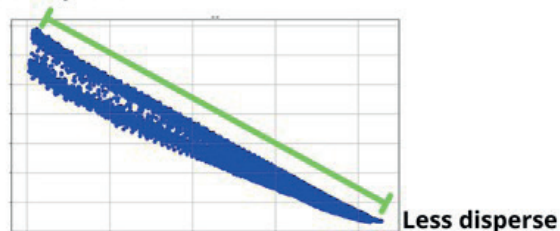
Figure 19 - First Attempt at establishing the dividing linear function.



The non-straightness problem

Figure 20 - Illustration of the 'continuum of dispersion'.

More disperse



A linear function mimics functionally a lane diving line if the road which originated the behaviors captured on the data was also straight. It fails to do so if the path delineated by the road is not straight, as we saw on Figure 19. But it does not "fail equally" on both ends of the coordinates plot. The upper half of the

plot, is correctly divided. If we assume that this is a function of the previously discussed delta of dispersion (Figure 20), and that the lane identification is easier when the vehicles' coordinates are more dispersed, a process can be set to classify each vehicle based on its coordinates on this portion of the plot.

The continuum of dispersion will be also a continuum between (saved outliers) the vehicles' first and last frames. Which will be each pole will vary depending on the camera placement and the lane's orientation. We'll establish the following process:

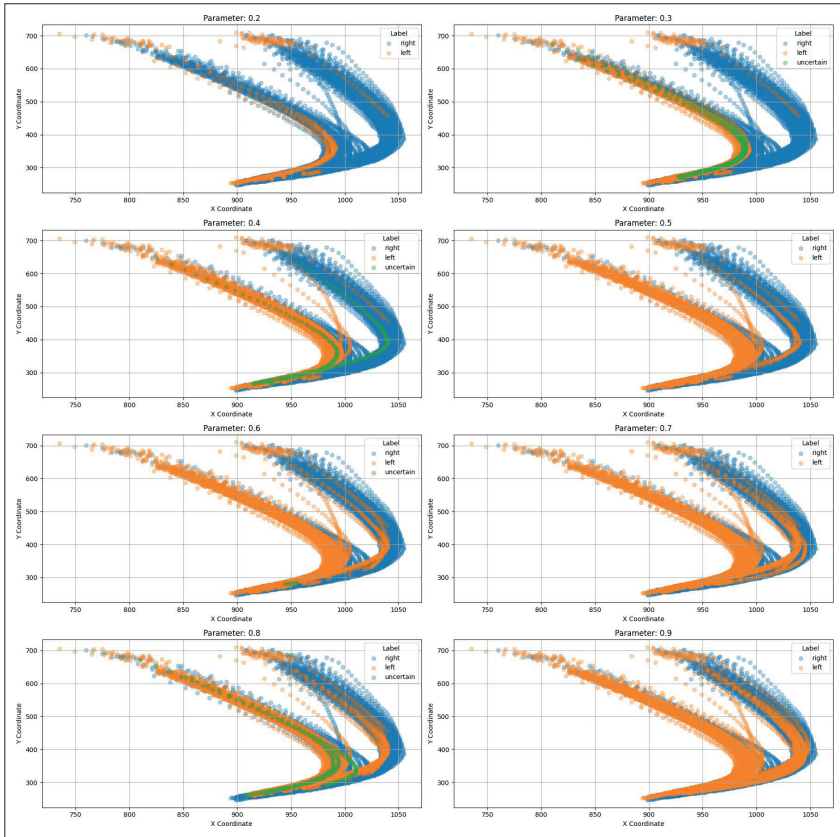
- There's a dispersion check between the aggregate of coordinates formed by the groups of first and last frames for the given dataset.
- The frames will be ordered going from the most dispersed moment (first or last) to the least dispersed (first or last)
- We'll set a parameter that will filter out only said portion of each vehicle's frames,
- e.g. if the first frames are more dispersed and we set the parameter of 0.5, the filtered dataset will contain only the first half of each individual vehicle frame collection.
- The same process of the previous section we'll be executed, but only on this filtered data.

This proved out to be a much more efficient approach, as we can see on Figure 21 where a previously impossible lane segregation was achieved with Parameter 0.5.

Setting parameters and how to select for clusterizations

Previously, our discussions of the quality of the clustering process have been based mostly on a visual check between what we know to be the road configuration from the visual aid we were provided with and checks based on knowing the vehicles that are being tracked on a given frame from the pixel coordinates + frame identification. This won't be enough if we want to setup a process that will automatically rank the clustering done by different parameters.

Figure 21 - Illustration of the 'continuum of dispersion'



On Scikit-learn we find 16 'Clustering Metrics' [Pedregosa *et al.* 2011]. We also find sub-sub-section 2.3.11: Clustering Performance Evaluation, on it we find three metrics that are described as to be used when the ground truth class assignments is not know: Silhouette Coefficient, Calinski-Harabasz Index and Davies-Bouldin Index.

Silhouette Coefficient: "calculated by taking into account the mean intra-cluster distance a and the mean nearest cluster distance b for each data point" [Shahapure and Nicholas 2020, p.747]. A higher Silhouette Coefficient score would point to more well defined clusters. The value returned is then, an average of all every points score.

Calinski-Harabasz Index: "based on a ratio of between cluster scatter matrix (BCSM) and within cluster scatter matrix (WCSM)". This will result, then, that the score will be higher for well separated and dense clusters [Pedregosa *et al.* 2011].

Davies-Bouldin Index: compares the distance between and the size of the clusters. It tends to identify "clusters which are far from each other and compact". It makes little sense to use it for our purposes since there will only be one distance between two clusters to measure.

So, we are left with the Silhouette Coefficient and Calinski-Harabasz Index. We would want both to be the highest score as possible for the final two lane classification operated on the dataset. This classification will constitute of: after establishing the dividing linear function, the filtered points we'll be classified following similar criteria to the orientation classification: if it has more than 50% of values as one lane, it'll be labeled as that lane, else it will be labeled as 'undefined'. Each ID will have a classification that will be applied to the full dataset.

We'll try three ways of selecting parameters based on them:

- Prioritize the biggest value for the Silhouette Coefficient, using the Calinski-Harabasz as a tiebreaker.
- Do the same, but now prioritizing the Calinski-Harabasz score.
- Selecting for the one with the highest "average score ranking", e.g. if a parameter had the second highest SC and the second Highest CH Score, it will take priority over the highest scoring SC but 4th best scoring CH Score. Ties will go to the one with the highest SC value.

The three approaches will be tested on the 11 datasets (excluding the single-lane one). The parameters will be 0.2 - 0.9 (with increments of 0.1), implemented using numpy's `linspace(0.2, 0.9, 8)`. The chosen parameter will be compared to the full plot of the 8 parameters and the lane classifications resulting from it. Then the selected parameter will be judged with regards to: being able to correctly segregate the lanes and being the "best" parameter compared to the other 8. The best parameter will be the one(s) which can more adequately identify both lanes and reduce to the maximum the number of "undetermined" vehicles.

RESULTS:

The labels are:

- **C:** The best parameter was able to correctly identified both lanes.
- **B:** The chosen parameter was the best at identifying both lanes
- **O:** The chosen parameter wasn't able to segregate both lanes.
- **-:** There wasn't a single, selected or not, parameter that was able to correctly segregate both lane.

The results are as following (Table 2):

The ranked choice was able to select the best parameter every time that a selective algorithm was able to do it. There were four times were at least on parameter was able to arrive at the lane segregation, but neither of the sorting method were able to select for it. On T12 R the process wasn't able to segregate the lanes.

The listing of interactions will proceed with the Ranked method to select for it's parameters on the lane segregation step, but it is not a fully functional solution. Here it opens a main pathway of inquiry for further works.

Table 2 - The Ability of the Parameter Selecting Processes to Obtain the Best Parameter.

Tabular Data	Silhouette Score	Calinski-Harabasz	Ranked
T12 L	C	C + B	C + B
T12 R	-	-	-
T20 L	C + B	C + B	C + B
T20 R	C + B	C + B	C + B
T34 L	O	O	O
T34 R	C + B	C + B	C + B
T47 U	O	O	O
T47 D	O	O	O
T58 L	O	O	O
T58 R	C	O	C + B
T80 R	C + B	C + B	C + B
Final Score	10	10	12

For the Road Portions that there was a parameter which that was able to segregate both lanes but wasn't selected by the selecting algorithm, the best parameter will be inputted so that the interactions analysis can proceed.

Listing Interactions

Now that we have each remaining vehicle classified in lanes we can produce a simple script that will produce the following steps:

- First it will enumerate every vehicle accordingly to the order it appears on the frame for that lane: the vehicle that has the lowest frame value for its first frame on a lane will have the lane order ID as 1, the second lowest, 2, etc. If there is a tie (which is a bad thing to happen) the vehicle with lowest ID will take priority.
- Then it will list every interaction between consecutive vehicles active on the screen at the same time. This could be a two vehicle interaction, or an N vehicle interaction. If we have say, 4 consecutive cars on the same lane active at the same time, the labels being, 20, 21, 22 and 23.

On Table 3 is the first 10 row of interactions listing for T12 L:

Table 3 - Consecutive Interactions for 4 same lane vehicles active on the same frames

Lane	Lane Oder IDs	Start Frame	Last Frame
left	(1,2)	3	42
left	(1,2, 3)	35	42
left	(2, 3)	35	69
left	(2, 3, 4)	35	69
left	(3, 4)	67	69
left	(3, 4, 5)	82	92
left	(3, 4, 5, 6)	88	92
left	(4, 5)	82	96
left	(4, 5, 6)	88	96
left	(4, 5, 6, 7)	94	96

This is the main result of the method that was here discussed. From the original computer vision tabular data, with the IDs of each vehicle, the center of the tracking boxes that were attributed to it as an coordinate and the respective

frame of that box, we were able, through successive, steps of data transformation and outlier removal we were able to establish the interactions between the vehicles on a given portion of the highway. From here, as discussed before, a myriad of other data obtained from the same set of data can be merged into utilizing the Lane + Lane Order ID. On the next section it will be provided an example of this, that will also be used to evaluate the accuracy of the interaction-detection.

Identifying Anomalous Interactions - An Use Case

Onto the interactions listing above will be merged into: the average distance between the vehicles involved in the interaction, if it is an interaction of more than two, the average distance will be the average of the distance between consecutive vehicles and the average speed of each vehicle during it's full route on the given highway portion.

The resulting table, for the interactions listed above, is shown on Table 4:

Table 4 - Consecutive Interactions for 4 same lane vehicles active on the same frames.

Lane	Lane Oder IDs	Start Frame	Last Frame	Average Distance	Avg Speed: ID1	Avg Speed: ID2	Avg Speed: ID3	Avg Speed: ID4
left	(1,2)	3	42	307	20.7	13.5	None	None
left	(1,2, 3)	35	42	376	20.7	13.5	16.1	None
left	(2, 3)	35	69	294	13.5	16.1	None	None
left	(2, 3, 4)	35	69	403	13.5	16.1	16.4	None
left	(3, 4)	67	69	403	16.1	16.4	None	None
left	(3, 4, 5)	82	92	367	16.1	16.4	9.2	None
left	(3, 4, 5, 6)	88	92	326	16.1	16.4	9.2	7
left	(4, 5)	82	96	307	16.4	9.2	None	None
left	(4, 5, 6)	88	96	157	16.4	9.2	7	None
left	(4, 5, 6, 7)	94	96	174	16.4	9.2	7	10.7

We'll provide an use case of this resulting data that will follow:

- Produce a column that calculates the average of the average speeds of the vehicles subetected to the listed interaction.
- The interactions that are on the bottom 20% of average distance and on the top 20% of "Average Average Speed" will be identified.

- The following interactions will be visually identified on the respective video footage. The interactions will be rated in three ways: they correctly identify consecutive same-lane vehicles, if they correctly identify two "visually fast" vehicles and if they identify two "visually close" vehicles.

This will be applied for two road portions: T12 Leftwards, as an example of a straight, well defined road portion, and on T47 Downwards, as example of a non-straight road portion that had didn't have the best parameter chosen by the selective algorithm.

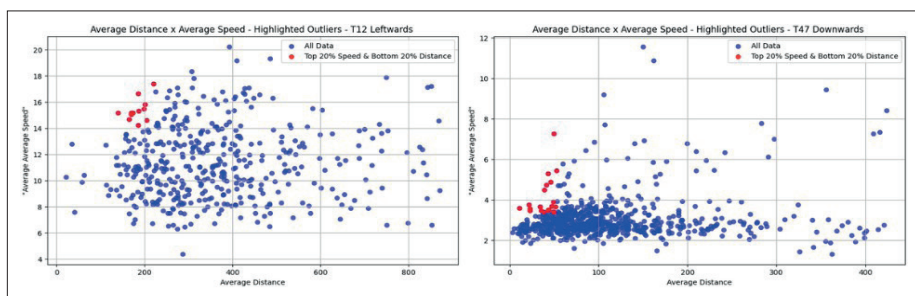
T12 Leftwards:

- 12 Anomalous Interactions were identified.
- 9 were between two vehicles and 3 between 3. These last, were between pairs of vehicles which interactions were already identified as anomalous.

T47 Downwards:

- 17 Anomalous Interactions were identified.
- 9 were between two vehicles, 8 between three vehicles and 3 between four. As with T12 L, all the anomalous 3 or 4-vehicles interactions were also classified as an anomalous pair.

Figure 22 - T12L and T47D Outlier Plot.



Every interaction, both on T12 Leftwards and T47 Downwards was indeed one, that from a visual analysis, evolved two consecutive-same-lane cars going at relatively high speeds to the other cars on the highway. All of the ones outlined

on T47D were indeed close, and 9 of the 12 on T12L were also close. We can see illustrations of the interactions captured on Figure 23.

Figure 23 - Example of Interaction Captured by the Listing + Selected Criteria.



RESULT ANALYSIS AND DISCUSSION

The method indeed functions as a way of listing interactions between same-lane vehicles on a given portion of a highway based on tracking data provided as tabular data with IDs for each vehicle. It seems to perform well for road portions that follow a straight lane configuration but not so much for curved pathways or with data that has too few Ns or too many path-outliers. We were able to establish a process in which we can deal with this non-straight roads, by applying different parameters to deal with the dispersion's discontinuity that we'll be a result of the perspective caused on the camera. But the parameter must be chosen "visually" as we weren't able to reliably fully automate it.

Our research was performed on very specific data as all the footage that produced the tabular data was from a period of 10 minutes. With a bigger N of vehicles on the highway the clustering process may not work with too many points of data. We also were forced to discard the one lane highway as we couldn't reliably find a way of identifying it, so our method is, in its currently state, only applicable to highways with two lanes.

Future Works

Future works delineated for the work done here are as follows:

- Further experimentation with different clustering techniques for outlier treatment.
- Further testing with clustering techniques for lane segregation.
- Find a way of establishing a check for single lane ways.
- Application of the designed method on more datasets, with new configurations and different computer vision techniques.
- Examination of a way to incorporate full tracking boxes into the process.
- Application of the method on a "real-time" setting.

REFERENCES

- Andrade, R. M. d. (2022). Vehicles behaviour detection from drones and surveillance images.
- bin Waheed, U., Al-Zahrani, S., and Hanafy, S. M. (2019). *Machine learning algorithms for automatic velocity picking: K-means vs. DBSCAN*, pages 5110–5114. Society of Exploration Geophysicists.
- CNT, C. N. d. T. (2023). Painel cnt de consultas dinâmicas dos acidentes rodoviários. CTB, C. d. T. B. (2021). Código de trânsito brasileiro.
- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Pfitzer, D. W. and Salmon, J. K., editors, *Second International Conference on Knowledge Discovery and Data Mining (KDD'96). Proceedings of a conference held August 2-4*, pages 226–331.
- Fraley, C. and Raftery, A. E. (1998). How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588.

Mount, J. and Zumel, N. (2019). *Practical data science with R*. Simon and Schuster. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Santosh, K. K., Dogra, D. P., and Roy, P. P. (2020). Anomaly detection in road traffic using visual surveillance: A survey. *ACM Computing Surveys*, 53(6):1–26.

Shahapure, K. R. and Nicholas, C. (2020). Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748.

Stančin, I. and Jović, A. (2019). An overview and comparison of free python libraries for data mining and big data analysis. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 977–982.

Čulík, K., Kalašová, A., and Štefancová, V. (2022). Evaluation of driver's reaction time measured in driving simulator. *Sensors (Basel)*, 22(9):3542.

MODELO DE INTELIGÊNCIA ARTIFICIAL PARA PREDIÇÃO DO PREÇO DO BITCOIN UTILIZANDO ANÁLISE DE SENTIMENTO DE DADOS DO *TWITTER*

Douglas Lessmann

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Emilia Colonese Carrard

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This work explores sentiment analysis of Twitter data with the goal of predicting the closing price of Bitcoin. A total of 4 million tweets were analyzed between the periods of 02/06/2021 and 03/05/2023. Among the methods used for sentiment analysis, two lexicon-based methods (VADER and TextBlob) and a pre-trained neural network model with a transformer architecture (RoBERTa) were analyzed. In terms of sentiment analysis, the tweets were classified as positive, negative, or neutral. The results show that prediction models that utilized both Bitcoin's historical price and the collected sentiments showed better results than models based solely on historical price.

RESUMO

Este trabalho aborda a análise de sentimento de dados do Twitter com o objetivo de prever o preço de fechamento do Bitcoin. Foram analisados 4 milhões de tweets entre os períodos de 06/02/2021 e 05/03/2023. Dentro dos métodos utilizados para a análise de sentimento, foram utilizados dois métodos léxicos (VADER e TextBlob) e um modelo de rede neural pré-treinado com arquitetura de transformers (RoBERTa). Em termos de análise de sentimento, os tweets foram classificados em: positivos, negativos e neutros. Os resultados obtidos revelam que os modelos de predição que utilizaram o preço histórico do Bitcoin e os sentimentos coletados possui melhores resultados que os modelos apenas baseados em preço histórico.

INTRODUÇÃO

Contextualização

Atualmente, com os avanços nos meios de comunicação, torna-se possível compartilhar informações em um nível jamais visto. De acordo com a pesquisa “*Data Never Sleeps 10.0*” (2021) realizada pela empresa DOMO, a cada minuto, em média, 5,9 milhões de consultas são feitas no Google, 500 horas de vídeos são carregadas no YouTube, 66 mil fotos são compartilhadas no Instagram, 347 mil tweets são compartilhados no Twitter e 231 milhões de e-mails são enviados. Além disso, de acordo com a Statista (2020) - uma empresa de dados globais com estatísticas, relatórios e análises sobre mais de 170 indústrias - o total previsto de dados para ser criado, capturado, copiado e consumido globalmente em 2023 é de 120 zettabytes, projetado para chegar em 181 zettabytes em 2025. Com este avanço no nível de informação disponível devido a criação e popularização da internet, tornou-se possível utilizar métodos de análise de sentimento das pessoas com relação às suas opiniões publicadas online [Pang, B., & Lee, L. 2008].

Os modelos de análise de sentimento podem ser utilizados também como forma de capturar os movimentos do mercado. Aplicações para prever o índice de mercado *Dow Jones Industrial Average* (DIJA) com base em publicações feitas por usuários no Twitter (*tweets*) mostra que a opinião compartilhada pelos usuários nas redes é correlacionada com o indicador econômico [Bollen *et al.*, 2011]. No Brasil, há evidência de correlação entre o índice BOVESPA por meio de notícias econômicas o impacto do sentimento das notícias e publicações em redes sociais, sendo que modelos complementados com a análise de sentimento superam estatisticamente *baselines* como o modelo randômico *Keep Trend* [Araújo Júnior, 2016]. Em estudo sobre flutuações cambiais para prever a taxa de câmbio entre dólares estado-unidenses e a rúpia paquistanesa com base em análise de sentimentos através de dados do Twitter possibilitou uma precisão de 82,14% para um modelo de classificação com objetivo de prever duas classes: aumento na taxa de câmbio e redução na taxa de câmbio [Naeem, S. *et al.*, 2021].

O *Bitcoin* é uma moeda virtual que possibilita pagamentos eletrônicos e tem sido usada cada vez mais desde sua criação em 2009. Ele foi construído em cima de um *log* de transações que permite transações irreversíveis e um histórico

público de histórico de transações [Böhme, Rainer *et al.* 2015]. Além disso, uma pesquisa feita em 2020 pela Hartford Steam Boiler (HBS), uma companhia de seguros com especialização em risco cibernético, e conduzida pela *Zogby Analytics* aponta que 36% dos pequenos e médios estabelecimentos nos Estados Unidos aceitam criptomoedas como forma de pagamento.

Analizando a produção científica brasileira no campo de finanças, umas das lacunas identificadas pelos autores para futuros trabalhos no campo de finanças [Magalhães-Timotio, J. G., & Barbosa, F. V. 2021] é o “efeito de notícias no comportamento dos preços de ativos no mercado de capitais”. O estudo proposto busca estudar, compreender e quantificar o impacto das publicações no Twitter nos preços de fechamentos diários do *Bitcoin*.

Problema de Pesquisa

O problema abordado nesta pesquisa busca aprofundar a produção científica brasileira de finanças explorando sobre análise de sentimento com o *Bitcoin*, dado que modelos de inteligência artificial com base em dados de notícias e redes sociais têm mostrado desempenho superiores aos métodos clássicos de análise. Considerando que o uso dos dados disponíveis no Twitter pode ajudar a ter uma melhor previsão das flutuações no valor da moeda, o problema enunciado é **“Como construir um modelo com base em análise de sentimento de *tweets* para prever variações no preço do *Bitcoin*?”**.

Objetivo Geral

O objetivo deste trabalho é “construir um modelo para prever variações no preço do *Bitcoin* com base em análise de sentimento de *tweets*” visando superar a precisão de modelos gerados por métodos clássicos estudados na literatura e utilizados pelo mercado.

Objetivos Específicos

- Coletar os conteúdos brutos contidos nos *tweets* através dos dados disponíveis na internet;

- Processar os conteúdos brutos dos *tweets* com o objetivo de caracterizar os sentimentos contidos neles;
- Utilizar os sentimentos coletados dos *tweets* para prever flutuações no preço do *Bitcoin*;
- Avaliar os resultados obtidos com um modelo *baseline* que utilize os métodos clássicos descritos.

TRABALHOS RELACIONADOS

Na análise de sentimento, diversos métodos foram desenvolvidos com o objetivo de alcançar melhores resultados ao capturar opiniões e sentimentos. Entre eles, temos os métodos léxicos, que são modelos que dependem da criação de palavras opinião, que são listas de classificação do que cada palavra representa. Por exemplo, “lindo” e “bonito” são palavras com opinião positiva. O processo de anotação dessas palavras pode ser feito de diversas maneiras, mas consomem bastante tempo [Liu; Bing, 2010].

O modelo VADER (Valence Aware Dictionary for sEntiment Reasoning) é uma ferramenta de análise de sentimento refinado com o objetivo de capturar sentimento em redes sociais. Leva em conta expressões utilizadas em redes sociais e adicionando regras de comunicação que ressaltam sentimentos e emoções (como pontos de exclamações adicionais em frases ou palavras em maiúsculo). Quando aplicado a diferentes contextos como textos de redes sociais, avaliações de filmes e produtos, ou notícias, o desempenho do VADER se aproxima do desempenho de humanos [Hutto, C. J. & Gilbert, Eric. 2014].

O TextBlob é uma biblioteca com ferramentas para análise de sentimento em Python. Além de fornecer funções para processamento de texto, ele também possui a capacidade de calcular a polaridade dos textos, permitindo a classificação em sentimento positivo, negativo ou neutro. [Aljedaani, W. *et al.*, 2022].

Como alternativa aos métodos léxicos expostos anteriormente, também existem modelos construídos com base em transformers, uma arquitetura apresentada em 2017 pelo artigo “Attention is all you need” e depois aprofundado para diversos ramos do aprendizado profundo que utiliza dados sequenciais. [Chollet, F. 2021]. O modelo RoBERTa (Robustly Optimized BERT Pretraining Approach) é um aprimoramento do modelo BERT. O modelo BERT implementa

uma melhoria com relação a outros modelos grandes de linguagem ao permitir que os tokens das frases sejam vistos de formas bidirecionais, diferente de outros modelos que apenas processam o texto da esquerda para direita [Devlin, J. *et al.*, 2019]. Entretanto, foi observado que um treinamento mais longo e com mais dados produzia melhores resultados para o modelo, sendo assim foi desenvolvida a versão RoBERTa [Liu, Yinhan *et al.*, 2019].

Modelos teóricos de finanças comportamentais já apontavam uma relação importante entre sentimento dos compradores e estimação de retornos pelas ações [Barberis, N., Shleifer, A. & Vishny, R. W. 1997]. Atualmente, com o avanço dos modelos de análise de sentimento e da disponibilidade de dados online, testes empíricos foram desenvolvidos. Para a predição de indicadores financeiros baseados em análise de sentimento, estudos anteriores evidenciam a existência de uma relação entre o índice de mercado Dow Jones Industrial Average (DIJA) e os sentimentos em redes sociais [Bollen, J. *et al.*, 2011].

Além do impacto de publicações em redes sociais, os sentimentos extraídos de notícias também mostram correlação com preço de ações de empresas como IBM e Twitter e, quando complementado com redes sociais, possibilitam modelos melhores [Khan, W. *et al.*, 2020].

No âmbito das criptomoedas, estudos apontam que há uma relação não linear significativa entre o índice de alegria do Twitter quando comparado com o preço das moedas: *Bitcoin*, *Ethereum*, *Ripple*, *Litecoin* e *Monero*. [Naeem, M. A. *et al.*, 2022].

DESENVOLVIMENTO

O estudo utiliza como base de dados *tweets* extraídos entre os períodos de 06/02/2021 e 05/03/2023 que possuem “#Bitcoin” ou “#btc”, disponibilizado na plataforma Kaggle, na base de dados nomeada *Bitcoin Tweets*. Ao todo, a base é constituída de 4,8 milhões de publicações, além de conter 13 colunas:

- *user_name*: Nome do usuário que fez a publicação;
- *user_location*: Localização informada no perfil do usuário;
- *user_description*: Descrição informada no perfil do usuário;
- *user_created*: Quando o usuário ingressou na plataforma;

- *user_followers*: Quantos seguidores o usuário possui;
- *user_friends*: Quantos amigos o usuário possui;
- *user_favourites*: Quantos favoritos o usuário possui;
- *user_verified*: Indicativo se o usuário possui uma conta verificada;
- *date*: Data com horário de quando o Tweet foi criado;
- *text*: Texto contido no Tweet;
- *hashtags*: Todas as hashtags contidas no texto do Tweet;
- *source*: Meio pelo qual usuário publicou o Tweet (Twitter Web App, Twitter for Android, etc.)
- *is_retweet*: Indicativo de a publicação é um *retweet* (republicação de uma postagem feita por outro usuário)

Ao todo, das 4.859.174 publicações, foram aplicados métodos de limpeza nos dados com o objetivo de permanecer apenas com a informação mais limpa e significativa possível. O primeiro passo foi descartar dados onde o campo *date* não correspondia a uma data, removendo assim 66 observações. Após este filtro, foi necessário identificar o idioma dos textos das publicações. Sendo assim, foi aplicado o modelo pré-treinado FastText, um modelo de código aberto desenvolvido pelo Facebook com o objetivo de identificar de forma eficiente o idioma de textos contendo mais de 157 idiomas. Com isso, o modelo identificou 4.136.565 como sendo Tweets em inglês. Estes foram mantidos na base, os demais descartados. Ao fim, foram removidos os tweets de usuários sem seguidores, já que essas publicações não seriam visualizadas por outras pessoas, e os *retweets*, devido ao fato de representarem uma duplicação da postagem original.

Com isso, a base final ficou com 4.075.759 de publicações. As colunas mantidas para análise foram a data e o texto da publicação. A coluna de texto foi processada para remover os símbolos de *hashtags*("#"), menções("@"), links e outros caracteres especiais. Além disso, foram convertidos para texto minúsculo.

Análise de sentimento

Com o objetivo de extrair o sentimento contido nos tweets, foram utilizados três modelos distintos, sendo dois modelos léxicos (Textblob e VADER) e

um modelo de aprendizado profundo baseado na arquitetura de transformers (RoBERTa). Neste trabalho, a implementação do modelo TextBlob ocorreu através da biblioteca TextBlob, o modelo VADER através da biblioteca de processamento de linguagem natural em Python NLTK (Natural Language Toolkit) e para o modelo RoBERTa a implementação foi feita através da biblioteca de modelos pré-treinados *Hugging Face*, com uma versão refinada para análise de sentimento de Tweets.

Para cada tweet, foi aplicado individualmente os três métodos descritos na seção 2. Após o enriquecimento dos tweets, os dados foram agregados a nível de dia, somando o total de postagem negativas, positivas e neutras de acordo com cada um dos modelos. Os tweets foram representados como percentual sobre o total de tweets feitos no dia. Ou seja, cada uma das colunas de sentimentos foi dividida pelo total de tweets que houve no dia, com o objetivo de representar em percentual quanto cada emoção fazia parte do total de publicações feitas no dia.

Nos gráficos abaixo, podemos observar que os modelo VADER possui maior presença de neutros, enquanto o TextBlob possui maior percentual de tweets classificados como positivos. Com relação aos negativos, TextBlob e RoBERTa possuem resultados muito similares, tendo diferença mais acentuada com relação a quantidade de neutros. Em geral, podemos observar um comportamento mais moderado para a classificação dos resultados obtidos com o modelo RoBERTa.

Gráfico 1 - Evolução do percentual de Tweets positivos.

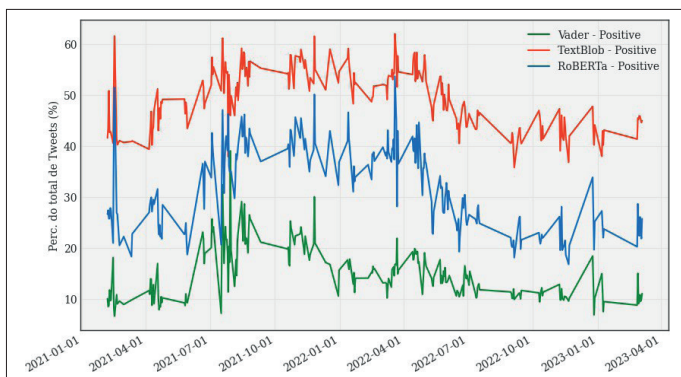


Gráfico 2 - Evolução do percentual de Tweets negativos.

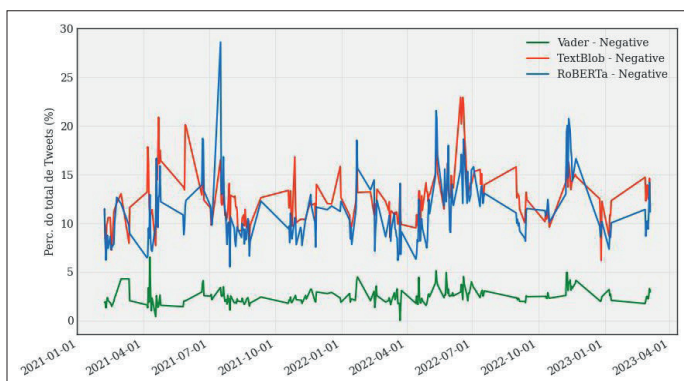
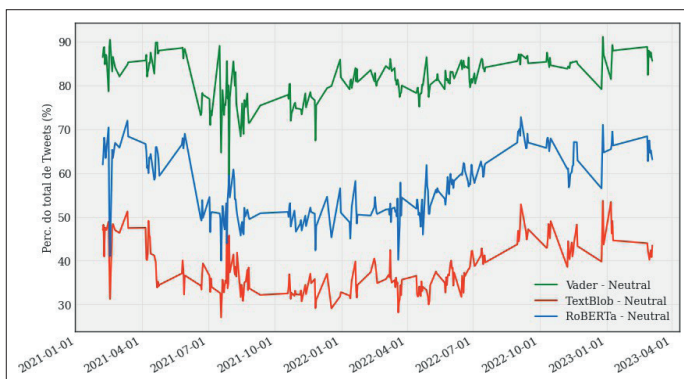


Gráfico 3 - Evolução do percentual de Tweets neutros.



Predição do valor do *Bitcoin*

Além dos dados da rede social, também foram extraídas as cotações de fechamento do *Bitcoin* para todos os dias presentes no período analisado. Os dados financeiros foram coletados de 01/01/2021 até 05/03/2023. Com isso, foi feito o cruzamento entre a cotação do *Bitcoin* nos dias de referência e as emoções. Ao todo, são 793 dias de dados, porém nem todos os dias possuem Tweets extraídos, provavelmente ocasionado por alguma limitação na fonte de Tweets utilizadas. Sendo assim, utilizamos como base apenas os dias em que possuíamos tweets, resultando em um total de 230 dias de observações. Destas observações, 200 dias foram usadas como amostra de treino (dias com dados disponíveis entre

06/02/2021 e 13/10/2022) e 30 como amostra de teste (dias com dados disponíveis entre 14/10/2022 e 05/03/2023).

Gráfico 4 - Evolução do preço do BTC.



ANÁLISE E DISCUSSÃO DE RESULTADOS

Para a análise da relação entre sentimento dos tweets e preço do *Bitcoin*, foram feitas duas análises:

1. Teste de causalidade de Granger;
2. Treinamento de modelos não lineares para prever o preço do *Bitcoin*.

Teste de causalidade de Granger

No teste de causalidade de Granger, necessitamos que nossa variável objetivo seja estacionária. Ao tomar a primeira diferença do preço de fechamento do *Bitcoin*, chegamos a uma série estacionária a um nível crítico de 99% no Teste de Dickey-Fuller aumentado.

O teste de Granger consiste na comparação entre duas séries temporais com o objetivo de testar a hipótese de que a série independente causa a série dependente. [Granger, C. W. J. 1969]. Para isso, foram criadas diversas regressões lineares com o seguinte formato:

$$Vbtc_t = \alpha + \sum_{i=1}^n \beta_i Vbtc_{t-i} + \mu_i \quad (1)$$

$$Vbtc_t = \alpha + \sum_{i=1}^n \beta_i Vbtc_{t-i} + \sum_{i=1}^n \gamma_i X_{t-i} + \mu_i \quad (2)$$

Onde $Vbtc$ representa a variação no preço da bitcoin (preço em T – preço em $T-1$) e X representa as variáveis relacionadas ao sentimento. Nos resultados obtidos, não encontramos nenhuma relação de causalidade significativa. Os menores p-valor ocorrem para o sentimento negativo com duas defasagens para o TextBlob e para os sentimentos negativos e neutros para o modelo RoBERTa. Entretanto, só poderíamos aceitar a hipótese de causalidade com um valor crítico baixo (90%). Com um valor crítica alto, de 95%, não encontramos relação de causalidade entra as variáveis e a variação do preço da bitcoin. Na tabela abaixo podemos consultar todos os valores obtidos.

Tabela 1 - Resultado dos p-valores obtidos no Teste de Granger.

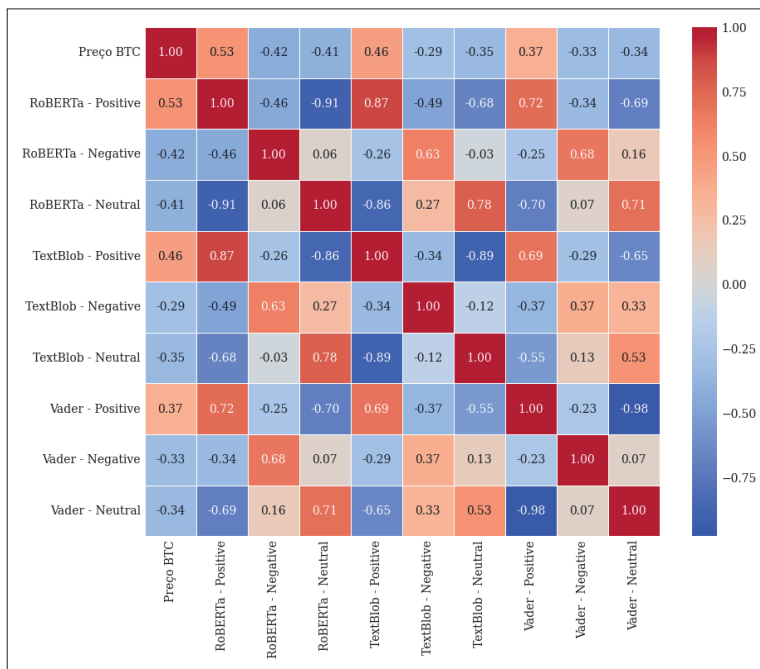
		Defasagens						
Modelo	Sentimento	1	2	3	4	5	6	7
VADER	Positivo	0,53	0,32	0,99	0,89	0,91	0,56	0,36
VADER	Negativo	0,32	0,25	0,76	0,47	0,84	0,25	0,39
VADER	Neutro	0,32	0,12	0,74	0,66	0,29	0,19	0,21
TextBlob	Positivo	0,58	0,13	0,75	0,76	0,36	0,21	0,25
TextBlob	Negativo	0,15	0,06*	0,52	0,38	0,32	0,25	0,13
TextBlob	Neutro	0,88	0,23	0,98	0,80	0,46	0,26	0,27
RoBERTa	Positivo	0,22	0,40	0,93	0,96	0,46	0,25	0,26
RoBERTa	Negativo	0,82	0,09*	0,40	0,56	0,66	0,43	0,29
RoBERTa	Neutro	0,17	0,09*	0,75	0,63	0,32	0,20	0,22

Entretanto, uma das limitações do teste de Granger deve-se ao fato de que o modelo explora a relação linear entre as variáveis. Para isso, podemos aprofundar a análise e treinar modelos não lineares para compreender se a adição de variáveis de sentimento produz melhores resultados de estimação.

Quando analisamos o coeficiente de correlação de Pearson entre as variáveis de sentimento dos diversos métodos com o preço do *Bitcoin*, notamos correlações lineares fracas. A maior correlação positiva ocorre entre sentimentos positivos do modelo RoBERTa (RoBERTa – Positive; correlação 0,53) e a maior correlação negativa ocorre com sentimentos negativos do modelo RoBERTa

(RoBERTa - Negative; correlação - 0,42). Abaixo podemos observar todos os coeficientes de correlação calculados:

Gráfico 5 - Mapa de correlação entre variáveis.



Treinamento de modelos não lineares para prever o preço da bitcoin

Como método para o treinamento de modelos não lineares foi utilizado o modelo LightGBM, um modelo de árvores de decisão com boosting de gradiente. Além disso, em certos casos, consegue performances superiores a outros métodos como XGBoost [Ke, Guolin *et al.*, 2017].

Ao utilizar modelos de árvores, podemos obter resultados além da relação apenas linear entre as variáveis independentes e o preço da bitcoin. Para isso, foram treinados 10 modelos distintos nos dados de treinamento e avaliados sobre a amostra de teste. Os 10 modelos foram configurados por:

Tabela 2 - Configuração de variáveis dos modelos.

ID do modelo	Variáveis
Baseline	Preço do <i>Bitcoin</i>
Vader 1	Preço do <i>Bitcoin</i> + Sentimento: Positivo
Vader 2	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo
Vader 3	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo, Neutro
TextBlob 1	Preço do <i>Bitcoin</i> + Sentimento: Positivo
TextBlob 2	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo
TextBlob 3	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo, Neutro
RoBERTa 1	Preço do <i>Bitcoin</i> + Sentimento: Positivo
RoBERTa 2	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo
RoBERTa 3	Preço do <i>Bitcoin</i> + Sentimento: Positivo, Negativo, Neutro

Para cada um dos modelos, foram utilizadas 3 defasagens de cada variáveis. Sendo assim, no modelo Baseline, tentamos prever o preço do *Bitcoin* no período T utilizando o preço do *Bitcoin* no período T-1, T-2 e T-3. Em cada modelo, as métricas de avaliação utilizadas foram o erro absoluto percentual médio (MAPE) e o percentual de casos em que o modelo acertou o sentido da flutuação: se capturou que foi um aumento/diminuição do preço, ambas aplicadas a amostra de teste.

Tabela 3 - Resultados obtidos em cada experimento.

ID do modelo	MAPE	% Acertos variação
Baseline 1	16,24%	43%
Vader 1	17,57%	43%
Vader 2	17,83%	43%
Vader 3	17,64%	47%
TextBlob 1	16,32%	43%
TextBlob 2	16,06%	47%
TextBlob 3	16,02%	47%
RoBERTa 1	16,18%	47%
RoBERTa 2	16,09%	50%
RoBERTa 3	16,22%	50%

Com o objetivo de prever preços, podemos ver que os modelo TextBlob 3 possui o menor MAPE: 16,02%. Já o modelo RoBERTa 2 possui o melhor resultado no acerto da variação: 50% e resultado de MAPE de 16,09%, próximo ao resultado obtido com o TextBlob 3. Estes resultados também confirmam a intuição que o teste de Granger havia apontado com os p-valores, de que as emoções extraídas

com TextBlob e RoBERTa possuem relação mais forte com o preço, do que as emoções obtidas com o VADER. Além disso, ambas versões destacadas possuem resultados superiores ao modelo que utiliza apenas os preços históricos.

Sendo assim, podemos verificar que os sentimentos publicados no Twitter possibilitam melhores resultados de previsão com relação ao preço do *Bitcoin*, indicando uma possível relação entre a variável dependente e as variáveis independentes. Além disso, o método RoBERTa, teve o melhor resultado na métrica de acertos na variação, mas em termos de MAPE, o método léxico TextBlob conseguiu resultados superiores. Podemos concluir que por mais os métodos léxicos sejam mais antigos, apresentaram bons resultados e sua facilidade de implementação com relação a modelos mais complexos como RoBERTa torna-se um ponto positivo adicional para sua adoção.

CONCLUSÃO

Este trabalho buscou analisar os diferentes métodos para análise de sentimento em dados de rede social com o objetivo de melhor modelar o comportamento de preços do *Bitcoin*. Podemos concluir que a inclusão da análise de sentimentos resultou em melhores predições. Entretanto é importante ressaltar algumas limitações deste estudo.

A primeira limitação ocorre na fonte de dados. Hoje, com a mudança de Twitter para X, as API's gratuitas de consultas não possibilitam mais realizar as coletas de dados tweets de outros usuários. Sendo assim, foi necessário utilizar uma base produzida por um terceiro, a qual possuía dias faltantes no período de análise. Além disso, este estudo limita-se aos dados de apenas uma rede social, entretanto existem outras fontes de informação online que podem impactar na percepção do mercado sobre um ativo, como sites de notícias e fóruns. As publicações coletadas são aquelas com as *hashtags* btc e Bitcoin. Esta limitação também pode impactar a análise porque exclui o sentimento de outros tweets que não possuem essas marcações e poderiam impactar em flutuações de mercado.

Outra ocorrência comum em redes sociais são posts façam menções a links que levam a outros textos e artigos, que podem conter opiniões mais fortes que apenas a publicação que divulga estes links, ou até mesmo imagens postadas que possuem emoções mais fortes que o título do texto. Por este motivo, é possível

que tweets considerados neutros, por apenas levarem a outra publicação em outro site, tenham algum sentimento atrelado a esta publicação. Além disso, as impressões das publicações (quantas vezes foi vista e quantas curtidas teve) é algo que, se levado em consideração, enriqueceria a análise.

Após explorar as diferentes técnicas de análise de sentimento, modelos léxicos (TextBlob e Vader) e modelos de aprendizado profundo (RoBERTa), podemos observar o modelo com melhor MAPE utilizava os todos os sentimentos (negativos, positivos e neutros) obtidos com o TextBlob, com resultado de 16,02%. Entretanto, no objetivo de capturar flutuações, o modelo com os sentimentos positivos e negativos extraídos através do método RoBERTa apresentou melhores resultados e teve um MAPE bastante próximo ao obtido com o TextBlob, com resultado de 16,09%. Sendo assim, podemos considerar os resultados do modelo RoBERTa como os sentimentos que mais enriqueceram o poder de predição do valor do *Bitcoin*.

Com base na análise gráfica da evolução do sentimento, os resultados do modelo RoBERTa se mostram mais moderados. O modelo VADER, que teve maior presença de neutros, resultou na pior performance quando comparado aos outros dois métodos. Já o método o TextBlob possui maior percentual de tweets classificados como positivos, ao passo que no modelo RoBERTa possui valores mais equilibrados entre as classes. Sendo assim, podemos concluir que capturar as polaridades dos tweets faz com que ocorram melhores predições. Os sentimentos mais moderados do modelo RoBERTa, poderiam ser um fator que ajuda a explicar porque o modelo obteve melhores resultados.

REFERÊNCIAS

- Aljedaani, W. *et al.* (2022). Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. *Knowledge-Based Systems*, 255.
- Barberis, N. Shleifer, A. & Vishny, R. W. (1998). A Model of Investor Sentiment. *Journal of Financial Economics*, 49(3), 307–343.
- Araújo Júnior, J. G. (2016). Previsão de setores e índice Bovespa por meio de notícias econômicas e suas repercussões em redes sociais. Páginas 90–106.

- Böhme, Rainer. Christin, Nicolas. Edelman, B. & Moore, T. (2015) Bitcoin: Economics, Technology, and Governance, *Journal of Economic Perspective*, 29(2), 213-238.
- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 2-6.
- Bollerslev, T., Chou, R. Y., & Kroner, K. F. (1986). ARCH modeling in finance. *Journal of Econometrics*, 52(1-2), 34-36.
- Coelho, L. dos S., Santos, A. A. P., & Costa Jr., N. C. A. da. (2008). Podemos prever a taxa de câmbio brasileira? Evidência empírica utilizando inteligência computacional e modelos econométricos. *Gestão & Produção*, 15(3), 642-645.
- Chollet, F. (2021). Deep learning with Python. 2. ed. São Paulo., 336-341.
- Delvin, J. *et al.* (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- DOMO (2022). Data Never Sleeps 10.0. Recuperado de: <https://www.domo.com/data-never-sleeps>.
- Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3), 424-438.
- Hutto, C.J.; Gilbert, Eric. (2014) A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM)*, 216-225.
- Ke, Guolin *et al.* (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, 30, 3146-3154.
- Liu, Bing. (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*. v. 2, 627-666, 2010.
- Liu, Yinhan *et al.* (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.
- Magalhães-Timotio, J. G., & Barbosa, F. V. (2021). A produção científica brasileira no campo de finanças. *Revista de ciências da Administração*, 22(58), 39-53.
- Naeem, M. A. *et al.* (2022). Does Twitter happiness sentiment predict cryptocurrency? *Journal of Financial Studies*. 10 (2), p. 123-140.
- Naeem S., Khan Mashwani W., Ali A., Irfan Uddin M., Mahmoud M. *et al.* (2021) Machine learning-based usd/pkr exchange rate forecasting using sentiment analysis of twitter data. *Computers, Materials & Continua*, 67(3), 3451-3459.
- Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 1-6.
- Statista. (2022). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. Disponível em: <https://www.statista.com/statistics/871513/worldwide-data-created/>.

SISTEMA DE NEGOCIAÇÃO DE DÓLAR FUTURO UTILIZANDO REGRESSÃO LOGÍSTICA

Ramon Vitor Malaquias Serra

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)
mserra.ramon@gmail.com

Elton Felipe Sbruzzi

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This paper presents a logistic regression model designed to predict intraday price variations in mini U.S. dollar futures. Utilizing a time series that incorporates the Relative Strength (RSL) indicator, the model aims to optimize buy and sell decisions in daily trading operations. The study covers the collection and preprocessing of historical data, implementation of the RSL, and model training. Backtesting results indicate that higher probability thresholds and longer periods lead to greater cumulative returns and profit factors. The model outperforms the Buy and Hold strategy under specific conditions, highlighting its potential as a tool for automated trading.

RESUMO

Este trabalho apresenta um modelo de regressão logística para prever variações intradiárias nos preços dos minicontratos futuros de dólar. Com base em uma série temporal que inclui o indicador Relative Strength (RSL), o modelo busca otimizar decisões de compra e venda em operações diárias. O estudo inclui a coleta e o pré-processamento de dados históricos, implementação do RSL, e treinamento do modelo. Resultados de backtesting mostram que filtros de probabilidade mais altos e períodos mais longos resultam em maior retorno acumulado e fator de lucro. O modelo supera a estratégia de Buy and Hold em condições específicas, demonstrando potencial como ferramenta de negociação automatizada.

INTRODUÇÃO

Contextualização

O mercado de renda variável no Brasil é dinâmico e crucial para a economia, pois considerando apenas o ano de 2022 a bolsa movimentou R\$ 7,46 trilhões [B3, 2023]. Nesse cenário, os minicontratos futuros de dólar se destacam, sendo amplamente utilizados por investidores institucionais e pessoas físicas para hedge cambial e especulação. Devido à sua alta liquidez e volatilidade, esses contratos atraem especuladores, especialmente na negociação intradiária, que permite aproveitar pequenas variações de preço e minimiza a exposição a riscos durante a noite.

Nos últimos anos, o uso de modelos matemáticos e algoritmos de negociação automatizada tem crescido exponencialmente, impulsionando uma transformação significativa no mercado financeiro. Esses modelos, baseados em técnicas quantitativas e análise técnica, permitem identificar padrões no comportamento dos preços e, assim, desenvolver estratégias mais eficazes para maximizar lucros. Nesse contexto, este trabalho buscou o desenvolvimento de um modelo que identifica as oportunidades de variação de preço do minicontrato futuro de dólar em operações intradiárias, utilizando abordagens modernas de modelagem preditiva, de forma similar a revisão conjunta de análise técnica e quantitativa demonstrada por Pan [2003].

Problema de Pesquisa

No decorrer da década de 2010 a Bolsa de Valores brasileira quintuplicou a quantidade de investidores pessoa física [Souza 2021]. No entanto, apesar do mercado financeiro estar se tornando cada vez mais acessível ao público em geral, ainda é extremamente desafiador para uma pessoa física competir com instituições financeiras, que possuem infraestrutura, acesso a informações em tempo real e profissionais altamente capacitados. Portanto, desenvolver estratégias de negociações robustas e lucrativas a longo prazo é uma tarefa complexa.

O objetivo desta pesquisa foi demonstrar um método para previsão da variação dos preços do dólar futuro e obter resultados positivos utilizando a regressão

logística. A escolha dessa técnica se justifica por sua eficácia comprovada na previsão de tendências de preços em mercados financeiros, como destacado por Gong and Sun [2009], que desenvolveram um modelo de regressão logística para prever a direção dos preços das ações com alta precisão. Segundo os autores, a regressão logística se destaca por sua simplicidade e interpretabilidade, além de permitir a utilização de variáveis que são facilmente obtidas a partir de informações diárias de negociação, tornando-a uma abordagem adequada para investidores individuais. O modelo também demonstrou ser menos complexo e mais preciso em comparação com outras metodologias, como redes neurais, o que a torna ideal para aplicações em ambientes de negociação rápida e dinâmicos, como o mercado de minicontratos futuros de dólar.

Objetivo Geral

O objetivo deste estudo foi desenvolver um modelo preditivo baseado em regressão logística, capaz de prever variações intradiárias significativas nos preços dos minicontratos futuros de dólar. Para isso, foram utilizadas como features do modelo uma série temporal que incorpora um indicador de força relativa (RSL) e um critério de saída baseado no tempo, visando otimizar as decisões de compra e venda em operações diárias e maximizar a rentabilidade.

Objetivos Específicos

Coletar e pré-processar dados históricos dos preços dos minicontratos futuros de dólar:

- Utilizar uma base proprietária com dados de preço do minicontrato futuro de dólar de maio de 2012 a setembro de 2023.
- Criar rótulos de compra e venda na base de dados e definir o período de treinamento.
- Realizar o tratamento dos dados para adequação ao modelo preditivo.

Implementar o indicador de Relative Strength (RSL) como variável preditora no modelo:

- Calcular o RSL com base na metodologia proposta por Levy [1967].
- Integrar o RSL à série temporal para utilização como feature no modelo de regressão logística, empregando um período de janela deslizando pré-definido.

Desenvolver e treinar o modelo de regressão logística para prever as classes compra ou venda:

- Utilizar técnicas de balanceamento de classes e treinar a regressão logística utilizando os dados pré-processados.
- Avaliar as métricas do modelo e ajustar seus parâmetros para otimização, se necessário.

Definir e aplicar critérios de saída baseados no tempo para operações de compra e venda:

- Estabelecer critérios de saída baseados no tempo que otimizem as operações diárias.

Analisar a performance do modelo e comparar com outras abordagens:

- Realizar backtests para avaliar o desempenho do modelo preditivo.
- Comparar os resultados com outras estratégias de negociação para validar a eficácia do modelo.
- Comparar os resultados com diferentes valores de filtro de probabilidade da saída do modelo.

DESENVOLVIMENTO

Coleta e pré-processamento de dados

Descrição da base de dados

A base de dados foi adquirida através da empresa Nelogica e contém os dados de preço de abertura, fechamento, máxima, mínima e volume financeiro referentes ao ativo minicontrato de dólar (WDO). O período de dados utilizados foi de 30/05/2012 a 18/09/2023 e a granularidade dos dados foi de um minuto.

Criação de Rótulos de Compra e Venda

Como o modelo de regressão logística é um modelo de classificação supervisionado é necessário que a base esteja rotulada, portanto foram definidas as seguintes regras de rotulação:

- Coletado o valor de fechamento do período n .
- Subtraído o fechamento do período $n + 10$ com o fechamento n .
- Se o resultado for positivo, é criada uma coluna Entrada com o valor 1 em n .
- Caso contrário, o valor da coluna Entrada é definido como 2 na posição n .

Os valores da coluna Entrada representam 1 como uma operação de compra e 2 como de venda. A figura 1 exemplifica o resultado dessa rotulação.

Figura 1 - Gráfico de Candlestick com marcações de compra e venda.



Definição do período de treinamento, teste e backtest

Com base em testes preliminares, foi definido o período de 30/05/2012 a 30/05/2014 para o treinamento e teste do modelo. Dentro desse intervalo, os dados foram divididos em 70% para treinamento e 30% para teste. O restante da base de dados, que abrange o período após 30/05/2014, foi reservado para o backtest. Foi dada preferência aos períodos mais recentes para o backtest, visando avaliar a capacidade do modelo em ser aplicável a novos dados. Essa divisão garantiu que uma parcela suficiente de dados fosse utilizada para o treinamento, assegurando que o modelo tivesse uma capacidade de aprendizado robusta, com uma avaliação eficaz durante o teste e o backtest.

Tratamento dos dados

Foi realizada uma análise para verificar a existência de dados faltantes na base, e constatou-se que todos os dados estavam completos. Para a criação do indicador Relative Strength (RSL), foram adicionadas três colunas ao dataset, correspondentes à média aritmética dos valores de fechamento dos últimos 8, 20 e 200 períodos. Esses intervalos foram escolhidos com o objetivo de fornecer uma perspectiva de curto (8 períodos), médio (20 períodos) e longo prazo (200 períodos) da movimentação dos preços.

Implementação do indicador relative strength (RSL)

Cálculo do RSL

O indicador Relative Strength (RSL) foi calculado seguindo a metodologia descrita por Levy [1967], que propôs o uso de relações entre os preços atuais e históricos para avaliar a força relativa de um ativo em relação a outros. Especificamente, o cálculo do RSL envolveu a criação de uma razão entre o preço de fechamento atual de um ativo e a média aritmética de seus preços de fechamento ao longo de um determinado número de períodos anteriores.

Integração do RSL no modelo

Após o cálculo do RSL para os períodos de 8, 20 e 200, foram realizados testes adicionais com diferentes intervalos para a janela deslizante utilizada na criação da série temporal. Com base nesses testes, foi definido um período de 30 minutos como o mais adequado. Como cada registro na base de dados corresponde ao intervalo de 1 minuto, o modelo considera os valores de cada um dos últimos 30 minutos como features individuais, permitindo capturar a dinâmica temporal detalhada do ativo.

Desenvolvimento e treinamento do modelo de regressão logística

Definição do modelo

O modelo preditivo utilizado neste trabalho é uma regressão logística, escolhida pela sua capacidade de lidar com problemas de classificação binária, como o proposto neste estudo. A configuração do modelo foi implementada utilizando a biblioteca scikit-learn, amplamente reconhecida e utilizada em aplicações de machine learning.

- **Variáveis Independentes (Features)**

As variáveis independentes, ou features, correspondem aos valores derivados da série temporal, incluindo os cálculos do Relative Strength (RSL) para períodos de 8, 20 e 200, aplicados em uma janela deslizante de 30 minutos. Cada um desses cálculos de RSL, em diferentes intervalos de tempo, foi tratado como uma feature separada, permitindo ao modelo capturar as dinâmicas temporais do ativo financeiro.

- **Variável dependente (target)**

A variável dependente, ou target, é a classe binária que indica a ação a ser tomada: "compra" ou "venda". Essa variável foi derivada a partir de uma estratégia de rotulagem, onde os dados históricos de preço foram analisados para determinar o momento mais apropriado para executar uma operação de compra ou venda.

- **Configuração do modelo**

O modelo de regressão logística foi inicializado com os parâmetros padrão do scikit-learn, sendo treinado com uma divisão do conjunto de dados em 70% para treinamento e 30% para teste, realizada utilizando a função `train_test_split`. Essa divisão foi feita com a intenção de avaliar a capacidade de generalização do modelo.

Técnicas de Balanceamento de Classes

O balanceamento de classes foi necessário devido ao desbalanceamento observado entre as classes de compra e venda dos dados. Especificamente, a quantidade de registros na classe “compra” correspondia a aproximadamente 36,3% dos dados, enquanto a classe “venda” correspondia a cerca de 63,7%. Esse desbalanceamento poderia levar o modelo a favorecer a classe majoritária (venda), resultando em um desempenho inferior na predição das operações de compra.

Para mitigar esse problema, foi aplicada a técnica de oversampling utilizando o método `RandomOverSampler` da biblioteca `imblearn`. O `RandomOverSampler` aumenta o número de instâncias da classe minoritária (compra) ao replicar aleatoriamente os registros existentes até que o número de instâncias se iguale ao da classe majoritária. Essa técnica foi escolhida por sua simplicidade e eficácia em situações em que o desbalanceamento de classes pode prejudicar o aprendizado do modelo.

Treinamento do Modelo

O processo de treinamento do modelo de regressão logística foi realizado utilizando o conjunto de dados balanceado previamente, conforme descrito na seção anterior. Para garantir a capacidade de generalização do modelo, os dados foram divididos em conjuntos de treinamento e teste, utilizando a função `train_test_split` da biblioteca `scikit-learn`. Especificamente, 70% dos dados foram alocados para o treinamento, enquanto os 30% restantes foram reservados para teste. Essa divisão foi feita de maneira aleatória, mas com um `random_state` fixo (42) para garantir a reprodutibilidade dos resultados.

O modelo de regressão logística foi escolhido devido à sua simplicidade e eficácia em problemas de classificação binária, como a decisão entre compra e venda. A implementação foi realizada com os parâmetros padrão do scikit-learn, e o treinamento foi conduzido utilizando o conjunto de dados de treinamento.

Para avaliar o desempenho do modelo, foram utilizadas várias métricas de classificação, aplicadas ao conjunto de dados do teste. As métricas escolhidas são amplamente reconhecidas e recomendadas para problemas de classificação, conforme descrito na documentação oficial do scikit-learn [scikit-learn: Model Evaluation 2024]:

Acurácia: Mede a proporção de previsões corretas em relação ao total de casos avaliados, fornecendo uma visão geral do desempenho do modelo. É definida como a razão entre o número de previsões corretas e o número total de previsões.

- **Precisão (weighted):** Reflete a proporção de previsões corretas para cada classe, ponderada pelo número de instâncias em cada classe. A precisão é especialmente importante em cenários onde as classes são desbalanceadas, pois evita que a performance seja dominada pela classe majoritária.
- **Recall (weighted):** Avalia a capacidade do modelo de identificar todas as instâncias relevantes (verdadeiros positivos) para cada classe, também ponderada pelo número de instâncias em cada classe. O recall é crucial quando o objetivo é minimizar falsos negativos.
- **F1 Score (weighted):** Representa a média harmônica entre precisão e recall, ponderada pelo número de instâncias em cada classe. O F1 Score é uma métrica que combina a precisão e o recall, proporcionando uma única medida que reflete tanto a exatidão quanto a abrangência do modelo.

Critérios de Saída Baseados no Tempo

Definição dos Critérios de Saída

Os critérios de saída das operações de compra e venda foram definidos com base em um período pré-estabelecido. Para a análise dos resultados, foram considerados diferentes intervalos de tempo: 5, 7, 10, 15, 20 e 30 minutos. Esses

intervalos foram selecionados para avaliar o impacto do tempo de permanência em cada operação na rentabilidade e eficácia das estratégias de negociação.

Análise de Performance do Modelo

Backtesting

Para o backtesting, foi utilizada uma base de dados que não fez parte do treinamento do modelo, abrangendo o período de 30/05/2014 a 18/09/2023. O objetivo foi avaliar a eficácia do modelo em condições de mercado que não foram previamente vistas. As métricas utilizadas para analisar os resultados incluem o retorno acumulado em pontos, a taxa de acerto (Win Rate), o fator de lucro (Profit Factor), e o Payoff Ratio.

O retorno acumulado em pontos mede o ganho ou perda total gerado pelo modelo ao longo do período de backtest. Ao calcular esse retorno, foi considerado um custo médio de 0,5 pontos por operação, refletindo os custos de transação que impactam diretamente na rentabilidade final.

Adicionalmente, foram consideradas as métricas de taxa de acerto (Win Rate), fator de lucro (Profit Factor), e Payoff Ratio, conforme sugerido por Wong [2019]. A taxa de acerto, que mede a proporção de operações lucrativas em relação ao total de operações, é calculada pela fórmula:

$$\text{Taxa de Acerto (Win Rate)} = \frac{\text{Número de Operações Lucrativas}}{\text{Número Total de Operações}} \times 100$$

O fator de lucro é a razão entre o total de lucros e o total de perdas, sendo calculado pela fórmula:

$$\text{Fator de Lucro (Profit Factor)} = \frac{\text{Total de Lucros}}{\text{Total de Perdas}}$$

Por sua vez, o Payoff Ratio compara o lucro médio das operações vencedoras com a perda média das operações perdedoras, sendo definido pela seguinte fórmula:

$$\text{Payoff Ratio} = \frac{\text{Ganho Médio por Operação Lucrativa}}{\text{Perda Média por Operação Perdida}}$$

Essas métricas são fundamentais para avaliar a consistência e a eficácia da estratégia. Elas foram escolhidas para fornecer uma avaliação abrangente da performance do modelo, considerando tanto o retorno absoluto quanto a capacidade do modelo de resistir a perdas significativas durante o backtesting. A metodologia segue as práticas recomendadas por Wong [2019], garantindo que o backtest forneça uma visão realista do desempenho da estratégia em cenários de mercado não observados anteriormente.

Comparação com Outras Estratégias

Para estabelecer uma base de comparação, foi calculada a razão entre o retorno acumulado em pontos dos modelos testados e o retorno obtido pela estratégia de buy and hold. Essa abordagem permitiu avaliar a eficiência dos modelos desenvolvidos em relação a uma estratégia passiva de mercado, fornecendo uma medida clara de desempenho relativo.

Comparação de Filtros de Probabilidade

Nos testes, utilizou-se a função `predict_proba` da biblioteca `scikit-learn` para que o modelo retornasse à probabilidade associada a cada uma das classes de compra e venda. A partir dessas probabilidades, foi realizado um experimento para determinar o limiar mínimo de probabilidade a ser considerado em cada operação. Foram testados limiares de 52%, 53%, 54%, 55%, com uma análise detalhada dos resultados para cada um desses valores.

ANÁLISE E DISCUSSÃO DOS RESULTADOS

Resultados das Métricas do Modelo

Acurácia, Precisão, Recall e F1 Score

O modelo de regressão logística apresentou uma acurácia de 51,61%, indicando que pouco mais da metade das previsões foram corretas. A precisão ponderada foi de 51,69%, sugerindo que o modelo identificou corretamente as operações de compra e venda em pouco mais da metade dos casos previstos. O recall ponderado foi de 51,61%, mostrando que o modelo capturou cerca de 51,61% das operações reais.

O F1 Score ponderado, que combina precisão e recall, foi de 51,31%, refletindo uma performance equilibrada, mas modesta, do modelo. A matriz de confusão mostrou que o modelo teve dificuldades em fazer distinção entre compra e venda, com 15.653 previsões corretas para compras e 21.047 para vendas, mas com uma quantidade significativa de erros em ambas as classes.

O relatório de classificação detalhou que o modelo foi ligeiramente mais eficaz em prever vendas do que compras, com uma precisão de 52% para compras e 51% para vendas.

Métricas do Modelo na Base de Backtesting

O modelo de regressão logística apresentou uma acurácia de 51,97%, indicando que pouco mais da metade das previsões realizadas durante o backtesting foram corretas. Comparado com a acurácia de 51,61% obtida nos dados de treinamento e teste, esse resultado sugere uma estabilidade no desempenho do modelo, com uma leve melhora.

A precisão ponderada foi de 51,75%, o que é ligeiramente inferior à precisão de 51,69% observada anteriormente. Isso indica que o modelo manteve uma capacidade consistente de identificar corretamente as operações de compra e venda, mesmo em um ambiente de mercado não visto anteriormente.

O recall ponderado foi de 51,97%, praticamente equivalente ao valor de 51,61% obtido nos dados de treinamento e teste. Esse resultado mostra que o modelo conseguiu capturar uma proporção similar de operações reais, sugerindo uma estabilidade na sua capacidade de reconhecimento.

O F1 Score ponderado, que combina precisão e recall, foi de 51,83%, comparado com 51,31% nos dados de treinamento e teste. Essa ligeira melhora no F1 Score indica que o modelo teve um desempenho um pouco mais equilibrado no backtesting, mas ainda reflete uma performance geral modesta.

A matriz de confusão para o backtesting revelou que o modelo teve dificuldades semelhantes em distinguir entre compras e vendas, com 245.958 previsões corretas para compras e 381.650 para vendas. No entanto, assim como no treinamento, houve um número significativo de previsões incorretas: 306.373 compras foram classificadas como vendas, e 273.572 vendas foram classificadas como compras.

O relatório de classificação detalhou que o modelo foi ligeiramente mais eficaz em prever vendas do que compras durante o backtesting, similarmente ao que foi observado nos dados de treinamento. A precisão foi de 51% para compras e 52% para vendas, indicando uma consistência na tendência do modelo de ser ligeiramente mais preciso em prever vendas.

Resultados do Backtesting

Desempenho Geral no Backtesting

Tabela 1 - Tabela com resultado do backtest.

Probabilidade Corte	Período	Retorno Acumulado	Custo	Taxa de Acerto	Fator de Lucro	Payoff Ratio	Modelo Buy_Hold
52,00%	5	-5184,0	17062,0	51,08%	0,95	0,91	- 1,97
	7	-1821,5		51,89%	0,98	0,91	- 0,69
	10	992,5		52,23%	1,01	0,92	0,38
	15	2799,0		52,63%	1,02	0,92	1,06
	20	3885,5		52,97%	1,02	0,91	1,47
	30	4421,0		52,82%	1,02	0,91	1,68
53,00%	5	3740,5	3811,5	54,04%	1,15	0,98	1,42
	7	6484,5		54,91%	1,22	1,01	2,46
	10	9048,0		55,91%	1,27	1,00	3,43
	15	12449,5		56,83%	1,32	1,00	4,72
	20	16411,0		57,78%	1,38	1,01	6,23
	30	20428,0		58,54%	1,41	1,00	7,75

Probabilidade Corte	Período	Retorno Acumulado	Custo	Taxa de Acerto	Fator de Lucro	Payoff Ratio	Modelo Buy_Hold
54,00%	5	3780,0	1206,0	56,86%	1,47	1,12	1,43
	7	5707,5		58,15%	1,65	1,19	2,17
	10	8076,0		60,63%	1,82	1,18	3,06
	15	11467,5		63,16%	2,05	1,19	4,35
	20	13933,5		64,74%	2,16	1,18	5,29
	30	18058,0		67,31%	2,40	1,17	6,89
55,00%	5	2922,0	514,5	57,87%	1,87	1,36	1,11
	7	3965,0		60,00%	2,03	1,35	1,50
	10	5237,5		62,64%	2,21	1,32	1,99
	15	6966,5		65,07%	2,46	1,32	2,64
	20	9008,5		67,12%	2,79	1,37	3,42
	30	11628,5		73,00%	3,16	1,17	4,41

Os resultados do backtesting, apresentados na Tabela 1, mostram que o desempenho do modelo melhora à medida que a probabilidade de corte e o período aumentam. Para uma probabilidade de corte de 0,52, o retorno acumulado variou de -5184 a 4421 pontos, com um Fator de Lucro entre 0,9465 e 1,021. Quando a probabilidade foi ajustada para 0,55, o retorno acumulado atingiu até 11628,5 pontos, com um Fator de Lucro máximo de 3,16. Esses resultados indicam que a estratégia se torna mais eficaz e robusta com operações de maior probabilidade e em períodos mais longos, oferecendo um melhor equilíbrio entre risco e retorno.

Análise de Sensibilidade aos Filtros de Probabilidade

A Figura 2 evidencia o impacto dos diferentes filtros de probabilidade nos resultados financeiros durante o backtesting. Com o aumento do filtro de probabilidade de corte de 0,52 para 0,55, observou-se uma melhora significativa nos indicadores de desempenho, como o retorno acumulado e o Fator de Lucro. Por exemplo, para um período de 30, o retorno acumulado aumentou de 4421 para 11628,5 pontos, enquanto o Fator de Lucro subiu de 1,021 para 3,16.

Além disso, o custo operacional também desempenha um papel crucial na análise de sensibilidade. Na medida em que o filtro de probabilidade aumenta, o número de operações pode diminuir, o que, por sua vez, reduz o impacto dos custos operacionais no resultado final. Isso fica mais visível na comparação dos

custos operacionais entre os diferentes filtros, onde, por exemplo, o custo operacional caiu significativamente ao passar de um filtro de 0,52 para 0,55, resultando em um aumento líquido de retornos após a retirada dos custos.

Essa abordagem de ajustar os filtros de probabilidade permite não apenas otimizar o retorno e o Fator de Lucro, mas também minimizar os efeitos adversos dos custos operacionais, promovendo uma estratégia de negociação mais eficiente e potencialmente mais lucrativa.

Comparação com Outras Estratégias

Comparação com Buy and Hold

Conforme a Figura 2, o modelo supera a estratégia de Buy and Hold principalmente em filtros de probabilidade mais altos (0,54 e 0,55) e em períodos mais longos. Nessas condições, o retorno acumulado e o Fator de Lucro do modelo são superiores, indicando melhor desempenho. Por outro lado, com filtros mais baixos (0,52), a estratégia de Buy and Hold se mostra mais eficaz, especialmente em períodos curtos, onde o modelo apresenta retornos acumulados inferiores. Isso sugere que o modelo funciona melhor em cenários onde é possível aplicar filtros de probabilidade.

Vantagens e Desvantagens das Estratégias Testadas:

O modelo de negociação testado apresenta várias vantagens em comparação com a estratégia de Buy and Hold, especialmente no contexto de gestão ativa e em ambientes de alta volatilidade. Uma das principais vantagens do modelo é sua capacidade de gerar retornos ajustados ao risco, especialmente quando são aplicados filtros de probabilidade que ajudam a otimizar as decisões de entrada e saída, conforme discutido na seção anterior. Além disso, o modelo oferece a possibilidade de reinvestimento dos lucros, permitindo a obtenção de retornos compostos (juros sobre juros), o que pode amplificar os ganhos ao longo do tempo.

Por outro lado, as desvantagens incluem a sensibilidade aos custos operacionais, que podem impactar significativamente a rentabilidade líquida. A necessidade de ajustes constantes e a complexidade envolvida na implementação e manutenção do modelo também podem ser vistas como desafios, especialmente

em comparação com a simplicidade da estratégia de Buy and Hold. Além disso, o modelo pode ser mais vulnerável em condições de mercado extremamente adversas, onde a volatilidade pode superar as previsões e levar a perdas maiores.

Assim, enquanto o modelo oferece um maior potencial de retorno e capacidade de adaptação às condições de mercado, ele também exige uma gestão mais ativa e um controle rigoroso dos custos operacionais para maximizar sua eficácia.

Interpretação dos Resultados

Conformidade com as Expectativas

Os resultados do modelo de regressão logística se alinham em grande parte com as expectativas iniciais, especialmente ao aplicar filtros de probabilidade mais altos (0,54 e 0,55), onde o modelo gerou retornos acumulados significativos e um Fator de Lucro elevado. Esses sucessos confirmam a eficácia do modelo em prever variações nos preços dos minicontratos futuros de dólar e otimizar operações.

Entretanto, o modelo apresentou limitações em cenários com filtros mais baixos (0,52) e períodos de operação mais curtos, onde não superou a estratégia de Buy and Hold.

Na prática, o modelo mostra-se promissor, mas sua aplicação bem-sucedida requer monitoramento contínuo e ajustes para garantir consistência nos resultados ao longo do tempo.

Implicações Práticas

O modelo de regressão logística mostrou-se eficaz para traders que buscam otimizar operações em minicontratos futuros de dólar, especialmente com filtros de probabilidade mais altos, que melhoram a performance. Em contextos reais, o modelo pode identificar oportunidades intradiárias de compra e venda, permitindo o crescimento composto dos ganhos. No entanto, exige gestão ativa e controle rigoroso dos custos operacionais para maximizar a rentabilidade. Assim, é uma ferramenta útil, mas que requer atenção para garantir resultados consistentes.

Limitações e Possíveis Melhorias

O modelo de regressão logística apresenta algumas limitações que podem ter influenciado os resultados. Uma limitação significativa é a simplicidade inerente ao modelo, que pode não capturar completamente a complexidade dos movimentos do mercado. Além disso, a sensibilidade aos custos operacionais pode reduzir a rentabilidade líquida.

Outra limitação é o uso de uma única variável preditora, o RSL, que embora útil, pode não ser suficiente para uma análise abrangente. A abordagem de rotulagem baseada em uma única regra de fechamento também pode ser simplista, não refletindo todas as dinâmicas do mercado.

Para melhorar o modelo, futuros estudos poderiam explorar a inclusão de variáveis adicionais, como indicadores de volume ou volatilidade, além de testar diferentes algoritmos de machine learning que possam capturar padrões mais complexos. A implementação de estratégias de ajuste dinâmico de filtros de probabilidade, adaptadas às condições de mercado em tempo real, também poderia aprimorar a eficácia do modelo.

Considerações Finais sobre os Resultados

Os resultados deste estudo destacam a eficácia da regressão logística como uma ferramenta preditiva para operações intradiárias no mercado de minicontratos futuros de dólar. Observou-se que a aplicação de filtros de probabilidade mais elevados e períodos mais longos resultou em melhorias significativas no retorno acumulado e no fator de lucro, conforme ilustrado pela Figura 2 (com o melhor fator de lucro) e Figura 3 (com o maior resultado acumulado). Esses achados contribuem para a literatura existente ao demonstrar que modelos preditivos podem ser adaptados com sucesso para otimizar estratégias de negociação em mercados voláteis.

Figura 2 - Resultado acumulado com filtro de 55 % e período de 30.

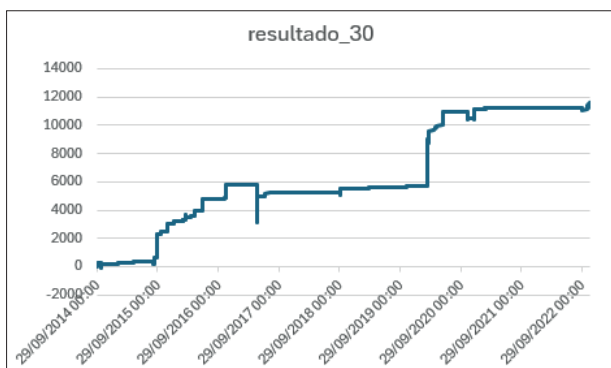


Figura 3 - Resultado acumulado com filtro de 53 % e período de 30.



Em futuros estudos, sugere-se a aplicação do modelo a diferentes ativos para verificar sua capacidade de generalização. Além disso, explorar outras técnicas de machine learning e ajustar os parâmetros do modelo pode ajudar a superar as limitações identificadas e aprimorar os resultados obtidos. A próxima etapa natural seria expandir o escopo da pesquisa, testando o modelo em um conjunto mais amplo de ativos e condições de mercado para avaliar sua robustez e aplicabilidade em diferentes contextos de trading.

CONCLUSÃO

Este estudo demonstrou que a regressão logística pode ser uma ferramenta eficaz para a previsão de variações intradiárias nos preços dos minicontratos futuros de dólar, oferecendo uma abordagem quantitativa para a tomada de decisões de compra e venda. Através do backtesting, verificou-se que o modelo,

quando ajustado com filtros de probabilidade mais altos, é capaz de superar a estratégia de Buy and Hold, especialmente em períodos mais longos e em condições de maior volatilidade.

As análises realizadas indicam que o modelo possui um potencial significativo para melhorar a rentabilidade em estratégias de negociação automatizada, especialmente quando aplicado com rigor na gestão de custos operacionais e em um contexto de gestão ativa. No entanto, as limitações identificadas, como a sensibilidade a custos e a necessidade de ajustes frequentes, sugerem que há espaço para melhorias.

Futuras pesquisas devem focar na aplicação do modelo a diferentes ativos para testar sua generalização, bem como explorar outras metodologias de machine learning para otimizar ainda mais os resultados. Com isso, espera-se que este estudo contribua para o avanço das práticas de negociação automatizada e ofereça insights valiosos para traders e investidores no mercado de minicontratos futuros.

REFERÊNCIAS

B3 (2023). Balanço Anual 2022: Resultados e Desempenho do Mercado de Renda Variável. Bolsa de Valores do Brasil. São Paulo. Disponível em: https://www.b3.com.br/pt_br/noticias/mercado-de-renda-variavel-movimenta-r-7-46-trilhoes-na-b3-em-2022.htm#:~:text=05%2F01%2F2023-,Mercado%20de%20renda%20vari%C3%A1vel%20movimenta%20R%24%207%2C46,trilh%C3%B5es%20na%20B3%20em%202022. Acesso em: 01 ago. 2024.

Gong, J. and Sun, S. (2009). A New Approach of Stock Price Prediction Based on Logistic Regression Model. International Conference on New Trends in Information and Service Science, Beijing, China, 2009, pp. 1366-1371, doi: 10.1109/NISS.2009.267.

Levy, R. A. (1967). Relative Strength as a Criterion for Investment Selection. The Journal of Finance, 22(4): 595–610.

Pan, H. (2003). A Joint Review of Technical and Quantitative Analysis of Financial Markets Towards A Unified Science of Intelligent Finance. Hawaii International

Conference on Statistics and Related Fields, Hawaii, USA. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a3675e58337501ec1554d6c89fd91eca583db26b>.

Souza, A.C.S. (2021). Análise da relação risco-retorno dos principais métodos de tomada de decisão ao alcance de investidores pessoa física na negociação de ações de empresas do mercado financeiro brasileiro. Araraquara, 2021 45 p.

Wong, L. (2019). Introductory Backtesting Notes for Quantitative Trading Strategies. Hong Kong University of Science and Technology. Disponível em: https://cse.hkust.edu.hk/~rossiter/independent_studies_projects/stocks_sector_random_forest/backtest_notes.pdf. Acesso em: 06 ago. 2024.

scikit-learn: Model Evaluation. Documentação Oficial do scikit-learn. Disponível em: https://scikit-learn.org/stable/modules/model_evaluation.html. Acesso em: 10 ago. 2024.

INTEGRAÇÃO DE MACHINE LEARNING E SISTEMAS AUTOMATIZADOS NA AVALIAÇÃO DE IMÓVEIS

Luciano Ventura

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)
avaliador.com@gmail.com

Filipe Alves Neto Verri

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This study applies machine learning techniques to enhance real estate appraisal, focusing on more accurate and automated predictive models. Through careful variable selection and advanced result validation, models such as LGBMRegressor and HistGradientBoostingRegressor demonstrated greater accuracy in predicting property values compared to traditional models. The study considered data on location, property characteristics, and socioeconomic variables, along with transformations to reduce the influence of outliers. Methods like stepwise, random forest, and PLS-SEM were used to identify the most relevant variables, resulting in robust and generalizable models. As a key contribution, this work proposes a standardized methodology for data collection and analysis in the real estate sector, applicable across different regions and various property types. The study also highlights the potential of web scraping to automate data collection, promoting greater efficiency and accuracy in the appraisal process.

RESUMO

Este trabalho aplica técnicas de machine learning para aprimorar a avaliação imobiliária, com foco em modelos preditivos mais precisos e automatizados. Através da seleção criteriosa de variáveis e da validação avançada de resultados, modelos como LGBMRegressor e HistGradientBoostingRegressor demonstraram maior precisão na previsão de valores de imóveis, em comparação com modelos tradicionais. O estudo considerou dados de localização, características dos imóveis e variáveis socioeconômicas, além de transformações para reduzir a influência de outliers. Métodos como stepwise, random forest, e PLS-SEM foram empregados para identificar as variáveis mais relevantes, resultando em modelos robustos e generalizáveis. Como contribuição, propõe-se uma metodologia padronizada para coleta e análise de dados no setor imobiliário, aplicável em diferentes regiões e para diversos tipos de imóveis. O trabalho destaca ainda o potencial do web scraping para automatizar a coleta de dados, promovendo mais eficiência e precisão no processo de avaliação.

INTRODUÇÃO

A incorporação da Inteligência Artificial (IA) em variados setores tem se mostrado revolucionária, elevando patamares de eficiência e precisão. No domínio da engenharia de avaliações, a IA surge como ferramenta potencial para refinar e inovar práticas já estabelecidas.

Contextualização

Grande parte da riqueza das famílias está concentrada em imóveis, desempenhando um papel central na acumulação de capital e na segurança financeira (Relatório do Banco Mundial, *The Changing Wealth of Nations* 2018).

“O valor de uma casa ou imóvel é determinado pelo número de pessoas que desejam o local; pelo número de outros locais semelhantes no mercado ao mesmo tempo, dos quais o comprador pode escolher; e pelas obstruções na maneira de usar o local conforme desejado para seu uso” (Horace F. Clark, in *Appraising the Home*, McMichael's *Appraising Manual*, 1951).

A engenharia de avaliações tem evoluído significativamente, consolidando-se como uma área essencial no Brasil e globalmente, com a modernização das técnicas tradicionais impulsionada pela rápida evolução tecnológica. No Brasil, as avaliações seguem a norma ABNT NBR 14653, complementada por normas internacionais como *International Valuation Standards* (IVS), as *European Valuation Standards* (EVS) e o *Red Book*, além das recomendações do *Banco Central do Brasil* e do *Financial Stability Board*.

Atualmente, cerca de 10 mil profissionais estão cadastrados no IBAPE, além de muitos outros que atuam no setor público, privado e judicial.

Problema de Pesquisa

A manifestação escrita de uma avaliação se dá por meio de um laudo de avaliação, que é o *‘relatório com fundamentação técnica e científica, elaborado por profissional da engenharia de avaliações, em conformidade com a ABNT NBR 14653, para avaliar o bem’* (ABNT NBR 14653-1:2019).

O laudo deve ser isento e fundamentado. Ele é utilizado para certificar instituições financeiras sobre o valor de imóveis em financiamentos, ações judiciais, garantias, hipotecas, e alienações fiduciárias. Também é exigido pelo poder público em desapropriações, servidões, cálculo de IPTU e ITBI, inventários e avaliações patrimoniais, entre outras finalidades.

A norma de avaliação de imóveis urbanos apresenta dois métodos principais de tratamento de dados:

- **Tratamento por fatores**
- **Tratamento científico**

Também permite o uso de ferramentas como regressão espacial, análise envoltória de dados e redes neurais artificiais. O método mais utilizado é o comparativo direto de dados de mercado, com modelos de regressão linear. Softwares específicos ajudam os engenheiros a elaborar os laudos.

Entretanto, os normativos ainda não abrangem todas as técnicas e possibilidades oferecidas pela ciência de dados. Isso destaca a necessidade de constante atualização profissional, além de revisões das normas para acompanhar as inovações tecnológicas e melhorar a eficiência dos processos. A correta coleta, armazenamento e uso de dados, junto com novas ferramentas analíticas como aprendizado de máquina e inteligência artificial, são cruciais para melhorar essas práticas.

A falta de padronização na coleta e registro de dados causa retrabalho, perda de informações e atrasos no atendimento das demandas. Além disso, os avaliadores lidam com pressões para cumprir prazos, o que compromete a obtenção de informações precisas, completas e atualizadas. Essas inconsistências geram divergências nos valores calculados entre os profissionais, elevando o risco de avaliações imprecisas e resultando em valores que não refletem adequadamente o mercado.

Foram coletados dados de profissionais com mais de 20 anos de experiência no segmento, onde foram observadas divergências nas nomenclaturas das variáveis e na relevância das informações. A falta de compartilhamento de dados entre avaliadores e a ausência de padronização na coleta limitam significativamente o desenvolvimento dessa etapa do processo de avaliação.

Objetivo Geral

Estudar o levantamento de dados de mercado para desenvolver uma base sólida na coleta e modelagem de dados imobiliários, com o objetivo de criar sistemas que aprimorem e automatizem a coleta e análise de informações no setor.

As fontes frequentemente contêm dados estruturados e não estruturados, como informações de imóveis, fotos, detalhes das imobiliárias e mapas (WIRTH; FROZZA, 2013). Os laudos de avaliação são baseados em variáveis qualitativas (categóricas), quantitativas (numéricas), dicotômicas ou do tipo “proxy”, como definidas pela ABNT NBR 14653-2.

As fontes frequentemente contêm dados, tanto estruturados quanto não estruturados, como informações do imóvel, fotos, detalhes da imobiliária e mapas com a localização do imóvel (WIRTH, R.; FROZZA, 2013).

Os laudos de avaliação são baseados em variáveis (features) que, na engenharia de avaliação, são categorizadas como qualitativas (categóricas), quantitativas (numéricas), variáveis dicotômicas ou dummy, e variáveis do tipo ‘proxy’. Estas últimas são utilizadas para substituir variáveis de difícil mensuração, presumindo-se que mantenham uma relação pertinente, obtida por meio de indicadores publicados ou inferidos em outros estudos de mercado (ABNT NBR 14653-2).

Grande parte dessas informações, que são vitais para avaliações precisas, acabam dispersas em inúmeros locais e em diferentes formatos, tornando a consolidação, análise e reutilização desses dados uma tarefa complexa.

Problema de pesquisa: Como centralizar, estruturar e otimizar o acesso e uso desses dados dispersos, de modo a melhorar a eficiência e precisão dos laudos de avaliação?

Objetivos Específicos

Este trabalho tem como objetivo realizar um estudo de caso que pode ser implementado em uma ou mais cidades e adaptado para diferentes tipologias imobiliárias. Busca explorar e validar ferramentas modernas, incentivar o compartilhamento de dados e auxiliar na padronização das variáveis a serem coletadas, com potencial para uso em softwares já existentes, que facilitam o trabalho dos profissionais. Além disso, visa desenvolver uma base para o uso de ferramentas

como web scraping e outras técnicas emergentes, otimizando todo o processo de coleta, análise e modelagem de dados, possibilitando, ao final, laudos com maior precisão e agilidade.

Contribuições

As principais contribuições deste trabalho incluem:

- ✓ Incentivo a padronização e compartilhamento dos dados;
- ✓ Apresentação de algoritmos e técnicas modernas para tratamento e modelagem de dados;
- ✓ Apresentar critérios para a seleção de dados, além de métodos eficazes de tratamento e análise de dados;
- ✓ Desenvolvimento de uma nova metodologia de seleção de variáveis, utilizando ranqueamento com base em múltiplos critérios de análise;
- ✓ Estudo de caso para validação do modelo proposto, demonstrando aplicabilidade;
- ✓ Proposta de ferramentas para automatização da coleta de dados, como web scraping, visando otimizar o processo de análise.

METODOLOGIA

Os dados foram coletados de diversas fontes e organizados em tabelas, com as variáveis ajustadas conforme os pesos fornecidos pelos profissionais. Scripts em Python foram desenvolvidos e executados com o auxílio do ChatGPT (versão GPT-4o) e do Gemini 1.0, utilizando a plataforma Google Colaboratory. O processo foi realizado em um computador com Windows 11, processador Intel Core i7, 8GB de RAM. Para a geração dos mapas, utilizou-se o Google Earth para visualizar a localização dos elementos pesquisados.

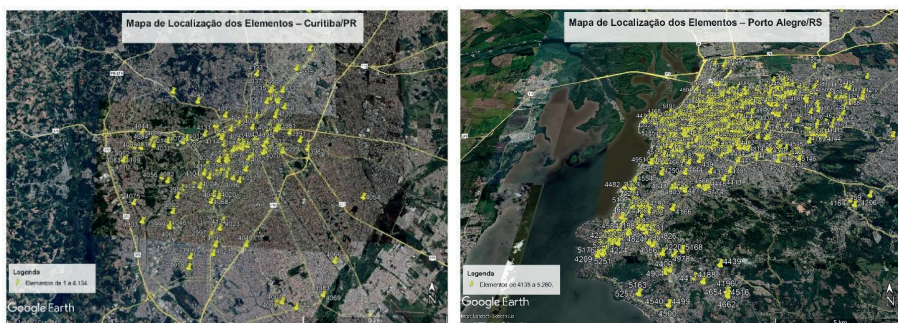
Dados

Localização

Foram coletados **5.260 registros de apartamentos** localizados em **Curitiba, Paraná, e Porto Alegre, Rio Grande do Sul**. Alguns dados já possuíam coordenadas geográficas, enquanto outros precisaram ser geocodificados com base no endereço do imóvel. Após esse processo, os dados foram utilizados para gerar um arquivo KML, que pode ser aberto no Google Earth, exibindo os pontos geográficos (latitude e longitude) dos imóveis coletados.

A seguir, são apresentados os mapas de localização dos elementos da amostra, uma informação comumente incluída nos laudos de avaliação. Dependendo do tipo e finalidade do laudo, essa etapa pode ser crucial no processo de avaliação.

Figuras 1 e 2 - Mapa de localização dos elementos da amostra – Curitiba/PR e Porto Alegre/RS.



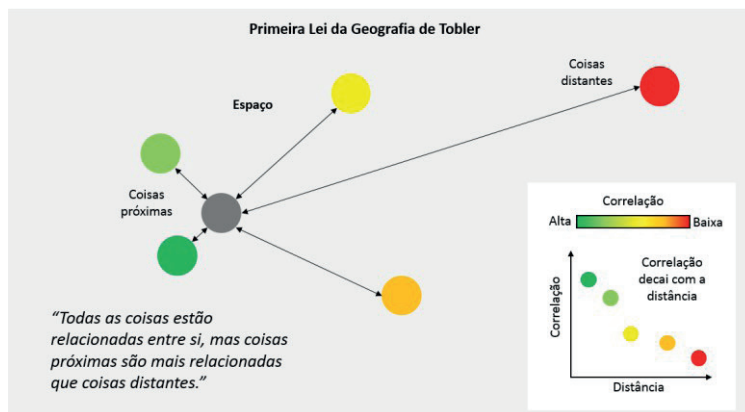
Seleção dos Dados

Localização

Como a maior parte dos dados dos últimos cinco anos se concentrava na cidade de Curitiba, optamos por seguir a análise apenas com essa cidade, totalizando **4.134 registros**. A seleção inicial dos dados foi fundamentada na Primeira Lei da Geografia, conforme ilustrado na Figura 3, formulada por Waldo Tobler em 1970, que afirma: *"Tudo está relacionado com tudo o mais, mas coisas mais próximas estão mais relacionadas do que coisas distantes"* (Tobler, W. R., 1970, A Computer Movie Simulating Urban Growth in the Detroit Region). Dessa

forma, foram utilizados apenas os dados de Curitiba, devido à maior quantidade de informações disponíveis.

Figura 3 - Mapa da Primeira Lei da Geografia de Tobler -te: <https://mapasabertos.com/wp-content/uploads/2020/08/fig01-1.png>.



Período da Pesquisa

Os dados de Curitiba foram coletados entre junho de 2020 e março de 2024, enquanto os de Porto Alegre abrangem de novembro de 2000 a julho de 2024. Conforme as normas, é imprescindível utilizar **dados contemporâneos** à data da avaliação. Quando não for possível, os preços devem ser atualizados por meio de consulta direta à fonte, e, caso isso seja inviável, por índices de pesquisas de mercado. Optamos por utilizar apenas os dados dos últimos cinco anos e corrigir a variável dependente (**Y=Valor Unitário**) pelo índice IPCA da FGV.

Filtragem dos Dados

Os dados foram tratados para garantir que as variáveis estivessem dentro de intervalos adequados, eliminando valores discrepantes. Foram inseridos comandos no início da seleção para ajustar os valores mínimos e máximos de cada uma das variáveis selecionadas, permitindo a filtragem de imóveis mais similares. Após esse filtro, restaram **4.121 dados** e **32 variáveis**, que serão usadas nas próximas etapas de análise e modelagem.

Detecção e Remoção de *Outliers*

Outliers foram identificados e removidos usando o método **Interquartile Range** (IQR). Após a remoção de 207 **outliers** na coluna Y e 176 na coluna Y1, o **DataFrame** foi reduzido para **3.854 registros**. Esse processo é essencial para melhorar a precisão dos modelos preditivos, eliminando valores atípicos que poderiam distorcer a análise.

O IQR é calculado como a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1). Valores fora do intervalo definido por: $Q1 - 1.5 \times IQR$, $Q3 + 1.5 \times IQR$.

Variáveis

A seleção adequada das variáveis é tão importante quanto a escolha e coleta dos dados. No processo de avaliação de imóveis, não há uma solução única, destacando a importância de abordagens que auxiliem os profissionais nessa etapa. O ajuste das variáveis melhora a acurácia dos modelos preditivos, eliminando dados que distorçam as projeções e garantindo que o modelo se baseie em informações representativas. Esse processo é essencial para assegurar a confiabilidade e precisão dos resultados

Tabela 1 - Variáveis Utilizadas.

Variável	Rótulo	Crescimento Esperado	Descrição
A	Código_imóvel		Código de referência do imóvel
B	Endereço		Endereço do imóvel
C	Complemento		Complemento da informação do imóvel
D	Bairro		Bairro onde se localiza o imóvel
E	Cidade		Cidade onde se localiza o imóvel
F	Estado		Estado onde se localiza o imóvel
G	Link de referência do imóvel		Link de referência da pesquisa do imóvel
H	Informante		Nome do informante do imóvel
I	Telefone		Telefone de contato do informante
J	Latitude		Coordenada da amostra conforme a característica geodésica decimal, representada em graus (°).
K	Longitude		Coordenada da amostra conforme a característica geodésica decimal, representada em graus (°).

Variável	Rótulo	Crescimento Esperado	Descrição
Y	Valor Unitário / Preço Unitário		Valor ou Preço de Venda / Área Privativa Principal Construída
Y1	Valor Unitário / Preço Unitário - Atualizado		Valor ou Preço de Venda / Área Priv. Principal - Atualizado para a data da modelagem, através de índices de variação
X1A	Data do Evento	(+) positivo	Período de contagem: mensal. Tem como referência inicial o dado mais antigo
X3	Renda IBGE 2010	(+) positivo	Proxy para localização. Faixa salarial por bairro, fonte: IBGE-Censo Demográfico 2010
X11	Área Privativa Principal Construída	(-) negativo	Área da unidade autônoma de uso exclusivo
X12	Idade Real/Estimada	(-) negativo	Aproximação da idade real do imóvel
X17A	Equipamentos do Condomínio - Elevador (Dicotômica)	(+) positivo	Identifica se o empreendimento possui ou não elevador: 0- sem elevador e 1- com elevador
X32	Estado de Conservação	(+) positivo	Escala de conservação: 1(Ruim) a 7(Novo)
X32B	Vida Total	(-) negativo	Estimativa da vida útil total de um imóvel em anos
X32C	Idade em % da Vida útil	(-) negativo	Idade do imóvel aparentada dividida pela vida total estimada
X32D	Fator 'K' Ross-Heidecke	(-) negativo	Valor retirado da Tabela de Ross-Heidecke utilizado no cálculo da depreciação do imóvel
X32E	CUB-Depreciado-R\$	(+) positivo	Custo Unitário Depreciado por metro quadrado, considerando-se a idade aparente e estado de conservação do imóvel
X36A	Total de Vagas	(+) positivo	Quantidade total de vagas de estacionamento
X43	Benfeitorias da Residência - Dormitórios (Qtd.)	(+) positivo	Quantidade de dormitórios presentes no imóvel
X47A	Banheiros	(+) positivo	Total de banheiros da unidade, incluindo banheiros sociais, lavabos e suítes
X49	Número de Pavimentos do Prédio	(+) positivo	Quantidade total de pavimentos do prédio
X50	Andar do Apartamento	(+) positivo	Pavimento onde se localiza o apartamento
X51	Número de Apartamentos no Andar	(-) negativo	Quantidade de apartamentos existentes no mesmo andar
X53	Custo Unitário		Padrão do CUB adotado
X54	CUB-R\$	(+) positivo	Custo Unitário Básico por metro quadrado de construção referente ao projeto-padrão
X57	Padrão	(+) positivo	Padrão construtivo e de acabamentos do imóvel

Transformações nas variáveis dependentes e independentes:

Transformações foram aplicadas nas variáveis dependentes e independentes para melhorar a acurácia do modelo. As variáveis dependentes passaram pelas seguintes transformações:

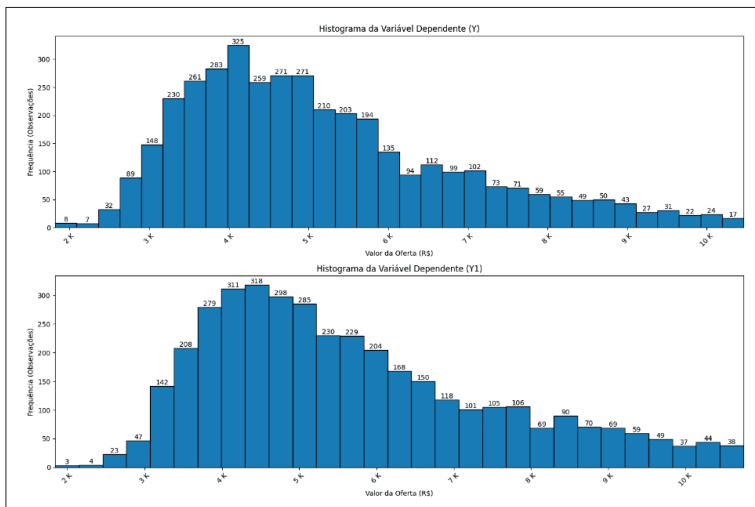
- **Y (Valor Unitário Original):** sem transformação, logaritmo natural e inverso da raiz quadrada.
- **Y1 (Valor Unitário Ajustado):** sem transformação, logaritmo natural e inverso da raiz quadrada.

Para a variável independente **X11 (Área Privativa Principal Construída)**, foi aplicado o logaritmo natural para reduzir a influência de outliers e estabilizar a variabilidade dos dados, tornando o modelo mais robusto. Além disso, variáveis categóricas com poucos valores foram ajustadas, e colunas numéricas foram convertidas para categóricas, garantindo variação suficiente para análises e modelagem.

Histograma das Variáveis Dependentes (Y e Y1)

A visualização das distribuições das variáveis dependentes Y e Y1 foi realizada por meio de histogramas. O objetivo principal dessa análise é entender a frequência dos valores de oferta de imóveis, tanto no **valor unitário (Y)** quanto no **valor unitário atualizado (Y1)**.

Figuras 4 e 5 - Histogramas das Variáveis Valor Unitário (Y) e Valor Unitário Atualizado (Y1)



Essa visualização é essencial para identificar a concentração de valores de oferta em determinados intervalos e refletir tendências do mercado imobiliário,

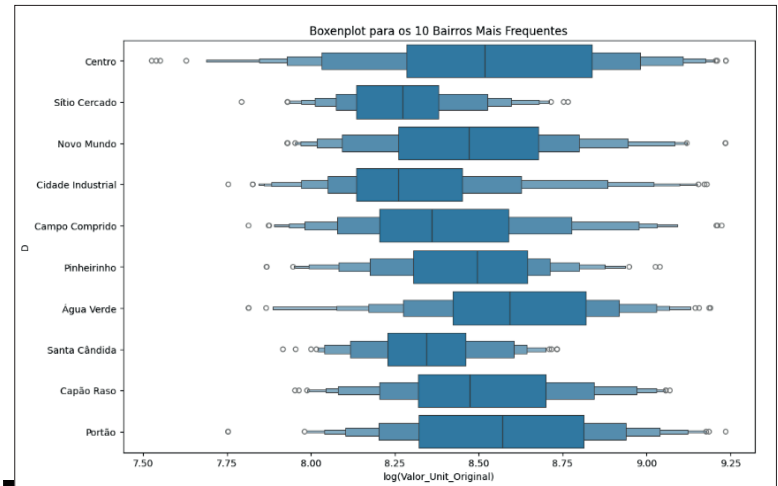
além de ajudar a detectar **outliers** e a necessidade de ajustes nos dados para aprimorar os modelos preditivos.

Análise Gráfica com Boxenplot

A técnica de boxenplot foi utilizada para uma análise detalhada da distribuição das variáveis dependentes e sua relação com as variáveis independentes. O boxenplot, uma variação do boxplot, oferece uma visão mais precisa da densidade dos valores e possíveis outliers em grandes volumes de dados. Foi aplicado para visualizar a distribuição das variáveis dependentes Y e $Y1$ em suas diferentes transformações (Y , $\log(Y)$, $1/Y^1/2$), em função dos bairros (variável categórica D), permitindo identificar variações de preços entre as regiões.

Para melhorar a clareza da análise, foi realizada a filtragem para exibir apenas os 10 bairros mais frequentes na amostra, evitando distorções de regiões com poucos dados e facilitando a visualização das principais tendências.

Figura 6 - Grafico Boxenplot de $\log(Y)$ em função dos 10 bairros mais frequentes da amostra.



A utilização de boxenplot com essa abordagem foi essencial para identificar discrepâncias na distribuição de valores e para observar como os preços dos imóveis variam entre os bairros mais representativos, fornecendo insights importantes para a modelagem preditiva.

Análise de Correlações Bivariadas - Critério V1

A **análise de correlações bivariadas** foi aplicada para avaliar a relação entre pares de variáveis e identificar possíveis redundâncias ou atributos com menor impacto no modelo preditivo. A soma das correlações de cada variável foi calculada para fornecer uma visão geral de sua associação com as demais. Com base nesses resultados, as variáveis foram ordenadas e ranqueadas, priorizando aquelas com maior potencial de contribuição ao desempenho do modelo. Foram gerados gráficos de calor (Figura 7) e de barras (Figura 8) para facilitar a visualização das correlações, destacando as variáveis mais relevantes.

Figura 7 - Matriz de Correlação Bivariada - Critério V1.

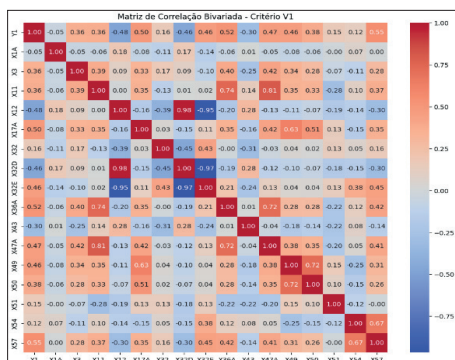
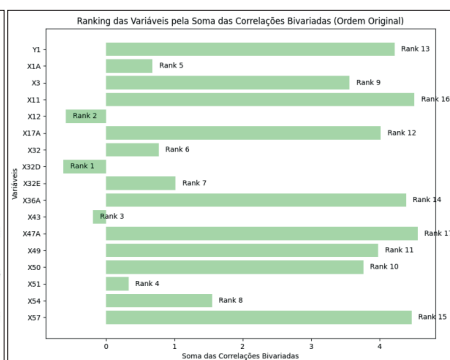


Figura 8 - Ranking das Variáveis pela Soma das Correlações Bivariadas (Ordem Original).



Para facilitar a leitura e comparação com outras análises, foi elaborado um gráfico de barras que classifica as variáveis da menor à maior correlação. O gráfico, complementando o heatmap, mostra a magnitude da associação de cada variável com as demais. Embora o objetivo não seja diretamente excluir variáveis com correlações mais baixas, o ranking ajuda a avaliar o impacto de cada variável no conjunto de dados. A análise das correlações bivariadas (Critério V1) oferece uma visão detalhada das inter-relações entre as variáveis, permitindo escolhas informadas sobre sua relevância para o modelo preditivo, sem um critério rígido de exclusão.

Regressão *Forward-Backward Stepwise* - Critério V2

A técnica de **Regressão Stepwise** foi aplicada, combinando os métodos **forward** e **backward**. A Eliminação Recursiva de Atributos (RFE) com Regressão Linear foi utilizada para selecionar as variáveis mais relevantes. O gráfico de barras gerado exibe o ranking das variáveis com base em sua importância para o modelo.

Portanto, as variáveis com barras maiores são consideradas menos relevantes e podem ser candidatas à exclusão, especialmente ao se buscar um modelo mais simples e eficiente.

Eliminação Recursiva de Atributos com *Random Forest* - Critério V3

A técnica **Eliminação Recursiva de Atributos (RFE)** foi aplicada em combinação com o **Random Forest** para selecionar variáveis. O modelo atribuiu um ranking de importância, mostrado no gráfico de barras, ajudando a identificar as variáveis mais influentes para o desempenho do modelo.

PLS-SEM via *Bootstrapping* - Critério V4

A técnica **PLS-SEM com bootstrapping** foi utilizada para ranquear as variáveis de acordo com sua relevância no modelo. O gráfico de barras apresenta o ranking das variáveis, facilitando a seleção dos atributos mais importantes. A variável com ranking 1 é considerada a mais importante para o modelo, enquanto valores maiores indicam menor relevância.

Florestas Aleatórias - Critério V5

A técnica de **Florestas Aleatórias** foi aplicada para avaliar a importância das variáveis. O gráfico de barras mostra as variáveis classificadas de acordo com sua relevância, destacando as mais influentes no modelo preditivo.

Resumo dos Gráficos - Critérios V1 a V5

A Figura 9 apresenta um resumo visual que combina todos os gráficos de barras e a matriz de correlação em um layout de fácil visualização. Esse conjunto

de gráficos oferece uma visão geral do processo de análise de variáveis, destacando as técnicas e resultados de cada abordagem aplicada, o que facilita a interpretação e a compreensão dos dados.

Figura 9 - Todos os Gráficos de Análise de Variáveis (Critérios V1 a V5).

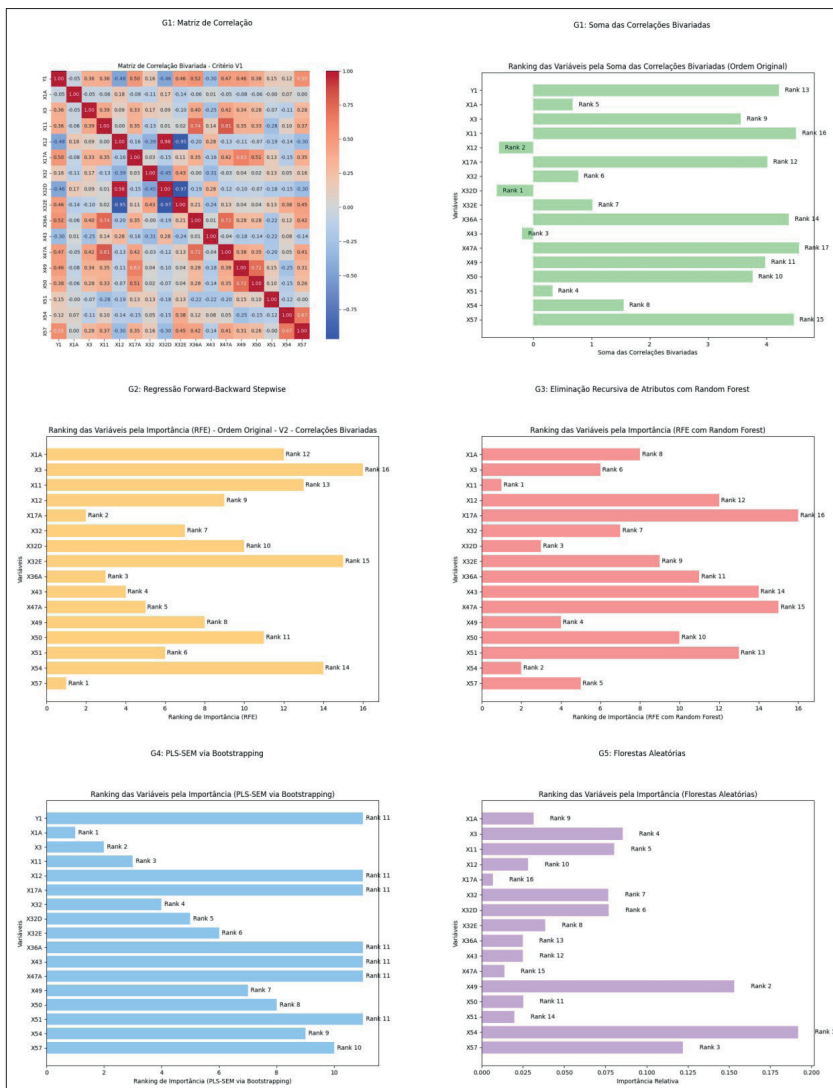
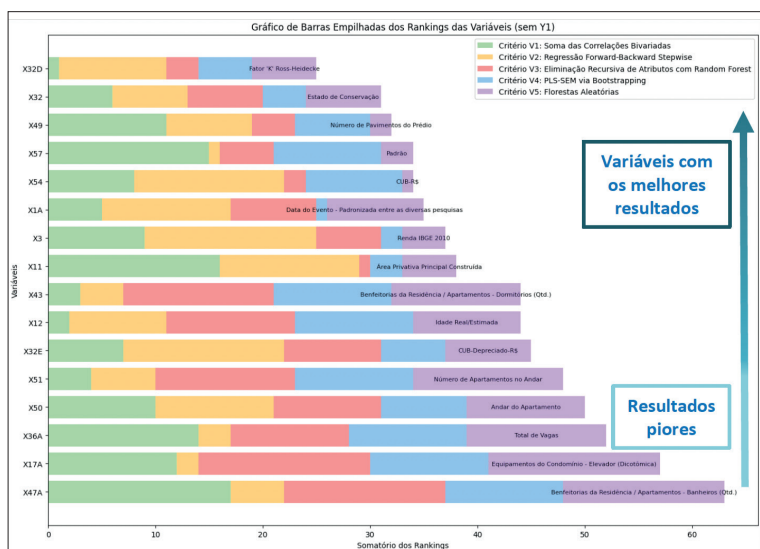


Gráfico de Barras Empilhadas dos Rankings das Variáveis (V1 a V5)

A Figura 10, apresentada a seguir, compara os rankings das variáveis de acordo com cinco critérios distintos, auxiliando na identificação de padrões de relevância e na otimização da seleção de variáveis no modelo preditivo.

Cada critério avalia a importância das variáveis dentro de um contexto preditivo, permitindo observar como diferentes métodos atribuem relevância a cada variável. Essa comparação é essencial para identificar padrões de relevância e embasar decisões sobre quais variáveis devem ser mantidas ou excluídas no modelo preditivo final. A análise empilhada fornece uma visão clara das variáveis mais influentes em cada critério, contribuindo para a otimização do desempenho do modelo por meio da seleção dos atributos mais significativos.

Figura 10 – Gráfico de Barras Empilhadas dos Rankings das Variáveis (Critérios V1 a V5).

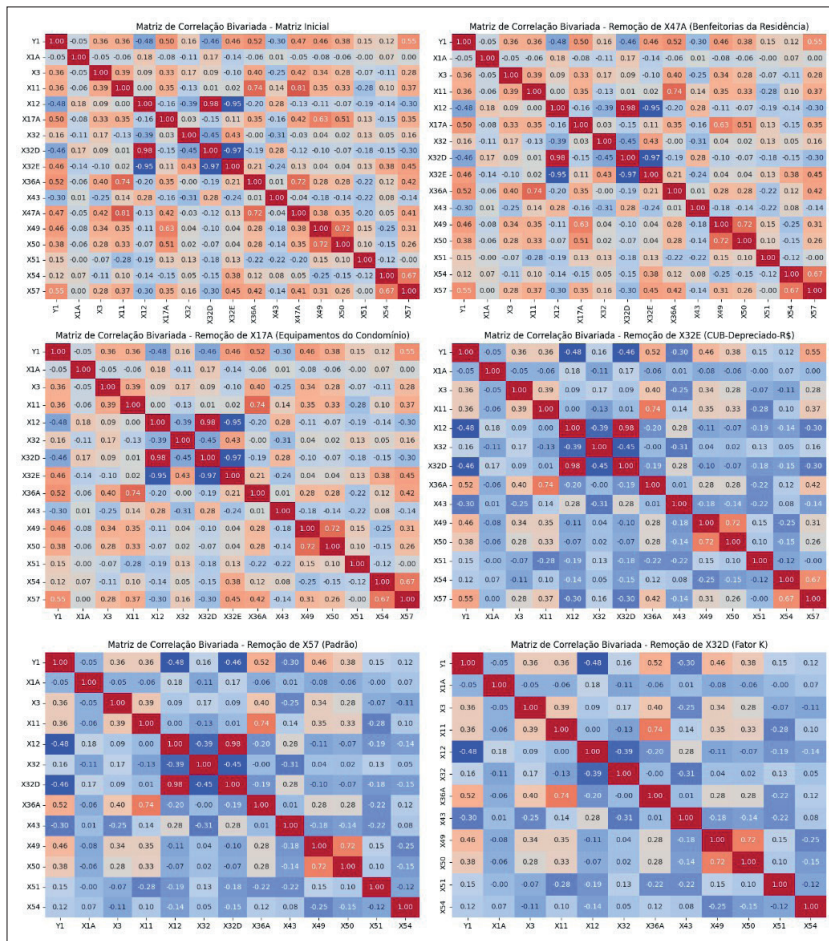


Exclusão das variáveis:

Com base no gráfico de barras empilhadas dos rankings das variáveis, foram removidas aquelas com menor relevância ou alta correlação entre si. A remoção foi realizada de forma progressiva, recalculando a matriz de correlação e visualizando os impactos por meio de gráficos de calor (heatmaps). As variáveis selecionadas para remoção foram: X47A, X17A, X32E, X57, X32D.

Após a remoção da terceira variável, o gráfico de calor, conforme mostrado na Figura 11, apresentou uma aproximação mais visível à cor azul, indicando correlações menores. A partir desse ponto, não foram observadas variações significativas. Portanto, optou-se por interromper a exclusão de variáveis após a remoção de X47A, X17A e X32E, uma vez que os resultados se mostraram satisfatórios.

Figura 11 - Simulações de remoção das variáveis selecionadas pelo autor.



Após a exclusão das variáveis, o processo foi finalizado e os testes estatísticos subsequentes foram realizados para validação dessa etapa. O DataFrame final contém as seguintes 15 variáveis remanescentes:

Variáveis restantes:

- D: Bairro.
- Y: Valor Unitário (R\$/m²).
- Y1: Valor Unitário, atualizado.
- X1A: Data do evento.
- X3: Renda média, IBGE (2010).
- X11: Área privativa principal (m²).
- X32: Estado de conservação (escala de 0 a 7).
- X32D: Fator 'K' Ross-Heidecke.
- X36A: Número de vagas.
- X43: Número de dormitórios.
- X49: Número de pavimentos do prédio.
- X50: Andar do apartamento.
- X51: Número de apartamentos no andar.
- X54: Custo Unitário Básico (CUB) de construção (R\$).
- X57: Padrão de acabamento do imóvel (escala de 1 a 3).

Processo de Validação dos Cálculos e Resultados

A validação dos cálculos de seleção de dados e variáveis foi realizada utilizando múltiplos modelos de regressão, como **XGBoost** e **LightGBM**, além de uma ampla gama de algoritmos regressivos, com o objetivo de maximizar a precisão das previsões. Os dados, após tratamento para remoção de outliers, foram divididos em **80% para treino** e **20% para teste**. O método **LazyRegressor** foi empregado para gerar um ranking dos melhores algoritmos com base no coeficiente de determinação ajustado (Adjusted R-Squared) e no erro quadrático médio (RMSE).

As transformações logarítmicas aplicadas às variáveis dependentes resultaram em diferenças significativas em relação aos dados não transformados, devido à alteração na escala dos dados. Assim, os resultados dessas variáveis não podem

ser comparados diretamente com os dados originais e devem ser analisados separadamente. Para interpretar os resultados no formato original, uma possibilidade seria aplicar a transformação exponencial inversa, retornando os dados à sua escala inicial.

Os melhores modelos, como `LGBMRegressor` e `HistGradientBoostingRegressor`, destacaram-se tanto nas variáveis originais quanto nas transformadas. A seguir, está a **tabela 2** com a lista completa dos algoritmos de regressão testados:

Tabela 2 - Lista completa dos algoritmos de regressão testados.

<code>AdaBoostRegressor</code>	<code>KNeighborsRegressor</code>	<code>OrthogonalMatchingPursuitCV</code>
<code>BaggingRegressor</code>	<code>KernelRidge</code>	<code>PassiveAggressiveRegressor</code>
<code>BayesianRidge</code>	<code>Lars</code>	<code>PoissonRegressor</code>
<code>DecisionTreeRegressor</code>	<code>LarsCV</code>	<code>RANSACRegressor</code>
<code>DummyRegressor</code>	<code>Lasso</code>	<code>Ridge</code>
<code>ElasticNet</code>	<code>LassoCV</code>	<code>RidgeCV</code>
<code>ElasticNetCV</code>	<code>LassoLars</code>	<code>SGDRegressor</code>
<code>ExtraTreeRegressor</code>	<code>LassoLarsCV</code>	<code>SVR</code>
<code>ExtraTreesRegressor</code>	<code>LassoLarsIC</code>	<code>TransformedTargetRegressor</code>
<code>GammaRegressor</code>	LinearRegression	<code>TweedieRegressor</code>
<code>GaussianProcessRegressor</code>	<code>LinearSVR</code>	<code>XGBRegressor</code>
<code>GradientBoostingRegressor</code>	<code>MLPRegressor</code>	<code>LGBMRegressor</code>
<code>HistGradientBoostingRegressor</code>	<code>NuSVR</code>	
<code>HuberRegressor</code>	<code>OrthogonalMatchingPursuit</code>	

Essa diversidade de algoritmos, utilizados com suas configurações padrão, permitiu uma análise abrangente das variáveis, resultando em modelos robustos e adequados para diferentes contextos de previsão.

Melhor Modelo

O **`LGBMRegressor`** foi o melhor modelo para "Valor Unitário Original", com um **Adjusted R-Squared de 0,75**. O **`HistGradientBoostingRegressor`** se destacou nas variáveis transformadas, incluindo logaritmo e inverso da raiz quadrada, com coeficientes de 0,74 e 0,75. Ambos os algoritmos mostraram excelente capacidade preditiva, com o **`HistGradientBoostingRegressor`** demonstrando consistência em diferentes transformações e escalas.

A **tabela 3** apresenta os 10 melhores modelos, incluindo Regressão Linear, amplamente utilizada em avaliações imobiliárias.

Tabela 3 - 10 melhores modelos, incluindo Regressão Linear.

Model	Adjusted R-Squared	R-Squared	RMSE *	Time Taken **	Y (Valor Unit. Original), ou Y1 (Valor Unit. Ajustado) ***	frame
LGBMRegressor	0,75	0,76	867,99	0,12	Valor_Unitário_Original	models_1
HistGradientBoostingRegressor	0,75	0,75	876,1	0,36	Valor_Unitário_Original	models_1
HistGradientBoostingRegressor	0,75	0,75	0,16	0,37	log(Valor_Unit_Original)	models_2
GradientBoostingRegressor	0,75	0,75	883,02	0,6	Valor_Unitário_Original	models_1
LGBMRegressor	0,74	0,75	0,16	0,12	log(Valor_Unit_Original)	models_2
LGBMRegressor	0,74	0,75	945,28	0,11	Valor_Unitário_Ajustado	models_4
HistGradientBoostingRegressor	0,74	0,75	947,93	0,35	Valor_Unitário_Ajustado	models_4
HistGradientBoostingRegressor	0,74	0,74	0,16	0,37	log(Valor_Unit_Ajustado)	models_5
LGBMRegressor	0,74	0,74	0,16	0,12	log(Valor_Unit_Ajustado)	models_5
GradientBoostingRegressor	0,74	0,74	0,16	0,61	log(Valor_Unit_Original)	models_2
LinearRegression (Método de Regressão Linear)	0,64	0,64				

Observação:

* **RMSE:** Os valores apresentam variação devido à aplicação de diferentes transformações, como o logaritmo natural ou o inverso da raiz quadrada, para ajustar a escala das variáveis dependentes.

** **Time Taken:** Tempo de processamento de cada modelo.

*** • **Y (Valor Unit.Original):** sem transf., log.natural e inverso da raiz quad.; e • **Y1 (Valor Unit. Ajustado):**

Y: Valor unitário original, sem transformações.

Log(Y): Logaritmo natural do valor unitário original.

1/√Y: Inverso da raiz quadrada do valor unitário original.

Y1: Valor unit.ajustado após atualizações para a data de modelagem e que sofreu as mesmas transformações **Log(y1)** e **1/√Y1**.

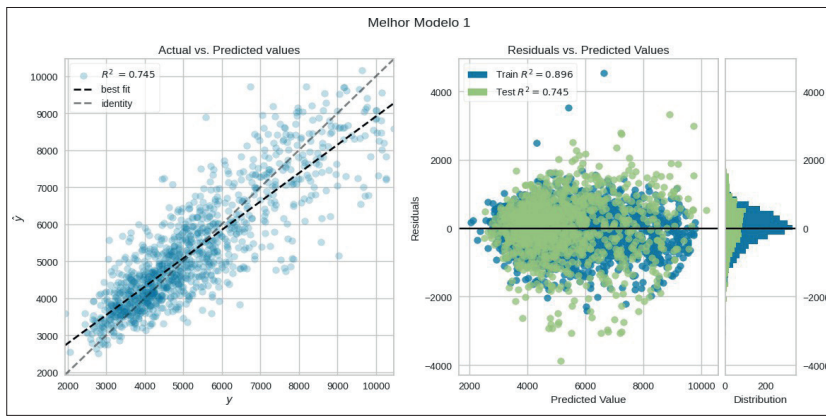
O gráfico a seguir, foi gerado com o modelo **LGBMRegressor**, considerado o melhor pelo critério R, e apresenta:

- **Gráfico de Valores Reais vs. Previstos:** Mostra a relação entre valores reais e previstos. Quanto mais próximos à linha de identidade, melhor o desempenho.

- **Gráfico de Resíduos:** Verifica se os resíduos (diferença entre valores reais e previstos) estão distribuídos aleatoriamente em torno de zero, indicando um bom ajuste.

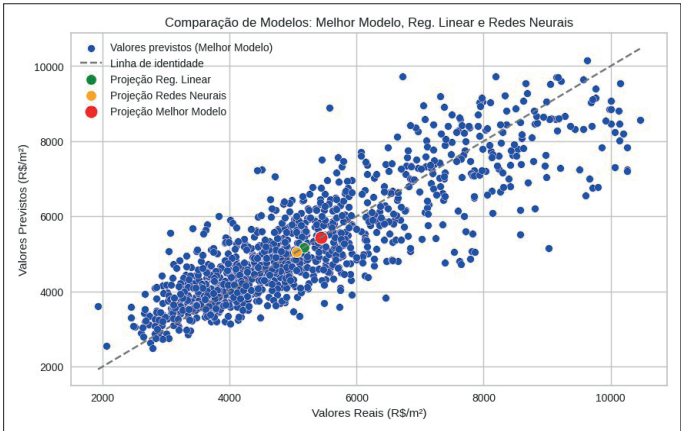
Esses gráficos ajudam a validar o desempenho do modelo e identificar problemas de ajuste.

Figura 12 - Gráfico gerado com o modelo LGBMRegressor, avaliado pelo critério R.



O gráfico a seguir ilustra a simulação de uma projeção de avaliação feita com três modelos: **LGBMRegressor (Melhor Modelo)**, **Regressão Linear** e **Redes Neurais**. A relação entre os valores reais e previstos é exibida, com a linha de identidade representando a previsão ideal. Cada modelo foi testado para comparar sua precisão preditiva com base nos dados fornecidos, considerando o valor médio das características de cada variável.

Figura 13 - Simulações de remoção das variáveis selecionadas pelo autor.



RESULTADOS

Diversas técnicas modernas mostraram-se eficazes para os engenheiros de avaliações, otimizando o tratamento de dados e liberando tempo para outras etapas fundamentais da atividade, como as vistoria e análise da documentação.

Foram aplicadas metodologias distintas no tratamento e seleção de dados imobiliários, com uma abordagem estruturada para a seleção de variáveis, baseada em técnicas combinadas, que visam melhorar a precisão e a eficiência do processo de avaliação.

O modelo preditivo desenvolvido, com o uso de técnicas de machine learning, especialmente o LGBMRegressor, apresentou alta capacidade de previsão do valor unitário dos imóveis, com um coeficiente de determinação ajustado de 0,75. O HistGradientBoostingRegressor também teve desempenho consistente, principalmente em variáveis transformadas. A análise gráfica dos resíduos e dos valores previstos indicou boa aderência à linha de identidade e distribuição aleatória dos resíduos, validando a eficácia do LGBMRegressor como o melhor modelo.

Os modelos adotados provaram ser estatisticamente superiores aos atualmente utilizados nas avaliações, que se baseiam principalmente em modelos de regressão linear.

Mesmo com pesquisas conduzidas por diferentes profissionais e com critérios variados de coleta de dados, foi possível identificar um processo de seleção

de dados e variáveis que resultou em modelos com desempenho satisfatório, demonstrando a consistência e a adequação dos resultados obtidos.

Através do compartilhamento dos dados e uma padronização de pesquisas, a tendência seriam resultados ainda mais satisfatórios e avaliações mais precisas.

Foram identificadas as seguintes variáveis mais relevantes para a predição dos valores dos apartamentos. As variáveis Bairro (D), Valor Unitário (Y), Valor Unitário - Atualizado (Y1), Data do Evento - Padronizada (X1A), Renda IBGE 2010 (X3), Área Privativa Principal Construída (X11), Estado de Conservação (X32), Fator 'K' Ross-Heidecke (X32D), Total de Vagas (X36A), Benfeitorias - Dormitórios (X43), Número de Pavimentos do Prédio (X49), Andar do Apartamento (X50), Número de Apartamentos no Andar (X51), CUB-R\$ (X54) e Padrão (X57) foram fundamentais para a precisão do modelo adotado.

Essas variáveis impactaram significativamente o desempenho preditivo e foram essenciais para a projeção do valor dos imóveis.

TRABALHOS FUTUROS

Com base nos resultados obtidos, diversas frentes podem ser exploradas para futuros aprimoramentos:

- **Ampliação da Base de Dados:** Incorporar mais registros de imóveis, novas tipologias e variáveis, como vizinhança e infraestrutura urbana, para aumentar a robustez dos modelos preditivos.
- **Exploração de Novos Modelos:** Avaliar modelos mais complexos, como redes neurais profundas e aprendizado semissupervisionado, pode aprimorar a acurácia das previsões. Essas técnicas modernas são capazes de lidar com conjuntos de dados maiores e mais dinâmicos, proporcionando maior capacidade de generalização.
- **Ajustes nos Modelos Testados:** Os algoritmos foram utilizados com configurações padrão; entretanto, novos ajustes poderiam ser explorados para alcançar resultados ainda mais satisfatórios.
- **Implementação de Novos Testes Estatísticos:** Realizar testes adicionais para validar os modelos, incluindo a verificação das expectativas de crescimento e do comportamento de cada variável. Aplicar outras transformações relevantes nas variáveis e incorporar ensaios

consagrados na engenharia de avaliações, em conformidade com as orientações da ABNT NBR 14653.

- **Integração com Web Scraping:** Automatizar a coleta de dados para manter os modelos constantemente atualizados, sempre com a devida validação pelos profissionais para assegurar a qualidade das informações.
- **Desenvolvimento de Plataformas:** Criar uma plataforma para o compartilhamento de dados entre avaliadores, padronizando processos e otimizando a velocidade e precisão das avaliações.
- **Aplicação em Diferentes Contextos Geográficos:** Testar a metodologia em diversas regiões do Brasil e mercados imobiliários distintos.
- **Dados Não Estruturados:** Utilizar algoritmos para análise de dados não estruturados, como fotografias de imóveis, a fim de identificar padrões construtivos, estado de conservação, e até prever possíveis problemas para investidores ou compradores de imóveis.

A coleta e o compartilhamento de dados são fundamentais para melhorar o processo de modelagem e avaliação de imóveis, oferecendo maior segurança para profissionais, clientes e contratantes. Este serviço, de extrema relevância para a sociedade, lida diretamente com questões financeiras, econômicas e sociais, já que os imóveis são onde vivemos, trabalhamos, estudamos, realizamos a maior parte de nossas atividades e investimos grande parte de nossos recursos. Isso torna a precisão e eficiência das avaliações cruciais para o bem-estar e o desenvolvimento da sociedade.

REFERÊNCIAS

FEDERAL RESERVE BOARD. *Survey of Household Economics and Decisionmaking (SHED)*. 2023. Disponível em: <https://www.federalreserve.gov/consumerscommunities/shed.htm>. Acesso em: 28 out. 2024. CLARK, Horace F. Appraising the Home. In: McMICHAEL, Stanley L. *McMichael's Appraising Manual*. 6th ed. Chicago: The Macmillan Company, 1951.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. 14653-1: Avaliação de bens — Parte 1: Procedimentos gerais — Rio de Janeiro: ABNT, 2019.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. 14653-2: Avaliação de bens — Parte 2: Imóveis urbanos — Rio de Janeiro: ABNT, 2011.

Banco Mundial. (2018). The Changing Wealth of Nations 2018: Building a Sustainable Future. Washington, DC: World Bank.

WIRTH, R.; FROZZA, A. A. Sistema inteligente para análise de viabilidade de imóveis. *Revista Eletrônica de Iniciação Científica em Computação*, [S. l.], v. 12, n. 1, 2013. Disponível em: <https://sol.sbc.org.br/journals/index.php/reic/article/view/1006>. Acesso em: 1 nov. 2023.

TOBLER, W. R. A computer movie simulating urban growth in the Detroit region. *Economic Geography*, v. 46, n. 2, p. 234-240, 1970.

DIAS, Willamy Mamede da Silva. *Machine learning e a previsão de preços de terrenos em Brasília*. 2023. Dissertação (Mestrado Profissional em Economia) – Universidade de Brasília, Brasília, DF, 2023.

SOUSA NETO, José Nilo Alves de; LADEIRA, Marcelo. Value estimation of properties administered by the Brazilian Army using machine learning and spatial components. In: **XXII COBREAP - Congresso Brasileiro de Engenharia de Avaliações e Perícias**, São Paulo. Anais: **IBAPE**, 2023.

DIAS, Willamy Mamede da Silva. *Machine Learning e a Previsão de Preços de Terrenos em Brasília*. Mestrado Profissional em Economia. Brasília: **IDP - Instituto Brasileiro de Ensino, Desenvolvimento e Pesquisa**, 2023.

MODELO DE PREVISÃO PARA O MERCADO DE COMMODITIES AGRÍCOLAS CONSIDERANDO FATORES MACROECONÔMICOS E INDICADORES FINANCEIROS

Eduardo Ferreira Vieira D'Almeida

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)
eduardofvdalmeida@gmail.com

William Ordiales Oliveira

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)
williamordiales@hotmail.com

Elton Felipe Sbruzzi

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

INTRODUÇÃO

Contextualização

O mercado de commodities pode sofrer diversos tipos de impactos, no longo e curto prazo. No longo prazo, os mercados de commodities estão sujeitos a choques ou mudanças de tendência que vão desde catástrofes naturais e intervenções político-militares até mudanças estruturais no mercado. Esses choques tendem a ser de natureza irregular e causam mudanças abruptas nos preços, geralmente para níveis mais altos, mas às vezes para níveis mais baixos [Labys 2003].

No médio prazo, os fatores que chocam os mercados de commodities também podem ser de natureza política ou cataclísmica, mas tendem a estar mais relacionados às condições econômicas nacionais ou às próprias forças de mercado. Flutuações nas forças de mercado tendem a ser observadas nas condições de oferta e demanda e no equilíbrio do mercado subjacente. Flutuações nas condições econômicas nacionais, comumente observadas na forma de ciclos econômicos, podem causar mudanças na produção industrial e, portanto, na demanda mineral ou nas taxas de juros e, em última análise, nos investimentos em commodities. As variações nas condições climáticas induzem mudanças na oferta agrícola e, conseqüentemente, nos preços dos produtos [Labys 2003].

Há momentos em que quase todos os preços de commodities se movem juntos e esse comportamento demonstra a dificuldade de se ignorar os fatores macroeconômicos quando se busca os determinantes dos preços das commodities, porém as ligações entre a macroeconomia e a volatilidade nos mercados de commodities ainda não foram examinados extensivamente [Frankel *et al.* 2010].

A aplicação de métodos estatísticos aos mercados e preços de commodities tem sido limitada devido ao comportamento errático de preços e quantidades. No entanto, um campo muito rico de modelagem e previsão cresceu ao longo do tempo [Labys 2003].

A volatilidade dos mercados de commodities, representada pelo Commodity Research Bureau Index e pelo S&P Goldman Sachs Commodity Index, reage às notícias macroeconômicas dos EUA e da China, incluindo o emprego e a

produção econômica dos EUA, bem como as intenções de compra de fabricantes chineses [Smales 2017].

Certas variáveis macroeconômicas e financeiras (ou seja, a volatilidade da inflação, o VIX (Índice do Medo), o spread de retorno padrão e o spread TED) impulsionam a volatilidade mensal das commodities [Prokopczuk *et al* 2017].

No artigo de Gargano e Timmermann (2014), os autores analisam a capacidade de diversas variáveis macroeconômicas e financeiras de prever preços de commodities. O estudo se baseia em um extenso conjunto de dados que abrange o período de 1947 a 2010, e as previsões são feitas em diferentes horizontes de tempo (mensal, trimestral e anual). Entre as variáveis utilizadas, destacam-se: Taxas de câmbio de moedas de commodities, Produção industrial, Relação investimento-capital, inflação e taxa de desemprego.

Para modelar essas previsões, Gargano e Timmermann (2014) usaram tanto abordagens univariadas quanto multivariadas. No caso das previsões univariadas, cada variável foi testada individualmente em modelos de regressão simples, com a variável de interesse sendo o retorno das commodities. Já para as abordagens multivariadas, os autores utilizaram técnicas como regressão ridge, que ajusta os parâmetros para reduzir os erros de estimativa, e combinação de modelos (complete subset regressions), que combina previsões de diferentes conjuntos de variáveis para melhorar a precisão preditiva.

Já Ye *et al.* (2020), os autores investigam o impacto das previsões macroeconômicas sobre a volatilidade dos futuros de commodities na China. O estudo utiliza um modelo GARCH-MIDAS, que combina dados diários de volatilidade com previsões mensais de variáveis macroeconômicas, como crescimento do PIB, produção industrial, oferta de dinheiro (M2) e inflação, para 15 contratos futuros de commodities, incluindo setores de metais, agricultura e energia.

Problema de Pesquisa

O problema abordado nesta pesquisa relaciona-se com a dificuldade de previsibilidade do mercado de commodities agrícolas por conta de diversos fatores macroeconômicos e do mercado financeiro, no curto e longo prazo. Considerando-se que a Inteligência Artificial possa auxiliar na previsão do mercado de commodities agrícolas o seguinte problema de pesquisa foi enunciado "É

possível utilizar Inteligência Artificial para prever o preço das commodities agrícolas, considerando fatores macroeconômicos e indicadores financeiros?”.

Objetivo Geral

O objetivo desta pesquisa é construir um modelo preditivo dos preços de commodities agrícolas no curto prazo, onde o estudo de caso abrangerá um cenário particular, e será considerado a maior parte de fatores que envolvem suas respectivas oscilações, como preço do petróleo, inflação, desemprego, índice nacional de preços do consumidor e produção industrial.

Objetivos Específicos

Os objetivos específicos apresentam, de forma detalhada, as ações que se pretende alcançar, considerando o cenário econômico do Brasil e a commodity de soja produzida no país. Desta forma, os seguintes objetivos específicos foram identificados:

- Identificar e detalhar todos os requisitos necessários para a utilização do modelo preditivo;
- Elaborar uma estrutura contemplando todos os requisitos que seja compatível com o modelo preditivo;
- Avaliar o modelo de predição através da aplicação em um estudo de caso.

DESENVOLVIMENTO

Dados

Foram coletados 6 fontes de dados distintas, que são elas:

- Histórico indicador ESALQ/BM&FBOVESPA (Soja): Esta fonte oferece o histórico mensal do indicador de preço da soja, elaborado pelo CEPEA/ESALQ em parceria com a BM&FBOVESPA. O indicador ESALQ/BM&FBOVESPA reflete o preço da soja em nível de produtor no estado do Paraná, servindo como um importante referencial para o mercado brasileiro de soja.

- Produção Física Industrial por Seções Industriais - Fonte: Instituto Brasileiro de Geografia e Estatística (IBGE). Esta base de dados mensal da produção industrial física por setores oferece um panorama da evolução da produção de diferentes segmentos da indústria brasileira, como metalurgia, produtos alimentícios, e outros setores relevantes.
- Índice de Preços ao Consumidor Amplo (IPCA) durante o Plano Real - Fonte: IBGE. Este é o principal índice de inflação no Brasil, medido mensalmente desde o início do Plano Real. O IPCA acompanha a variação dos preços de uma cesta de bens e serviços.
- Índice Nacional de Preços ao Consumidor (INPC) durante o Plano Real - Fonte: IBGE. O INPC mede a variação de preços para famílias de baixa renda (1 a 5 salários mínimos) e também é calculado mensalmente desde o Plano Real.
- Taxa de Desocupação (Taxa de Desemprego) - Fonte: IBGE. Esta estatística mensal reflete o percentual da população economicamente ativa que está desocupada (desempregada), servindo como um importante indicador da saúde do mercado de trabalho.
- Soybean Prices - 45 Year Historical Chart - Fonte: Macrotrends. Informações diárias do preço da soja dos últimos 45 anos.

Base de Dados Principal

Considerando que os indicadores financeiros são dados mensais, foi necessário deixar a base de dados Soybean Prices em uma granularidade mensal, para isso foi estipulado o preço de fechamento da variável target, ou seja, o último dia do mês (Figura 1).

As variáveis que constituem a base de dados principal são: Country: País de análise (Apenas Brasil)

Year: Ano

Month: Mês

Oil Price: Preço do Petróleo Soybean Price: Preço da Soja

ESALQ/BM&FBOVESPA: Indicador do Preço da Soja Indústria de transformação:

Indústria extrativa:

Indústria geral:

INPC: índice nacional de preços ao consumidor

IPCA: índice de inflação

Taxa de desocupação: Taxa de Desemprego Date: Data

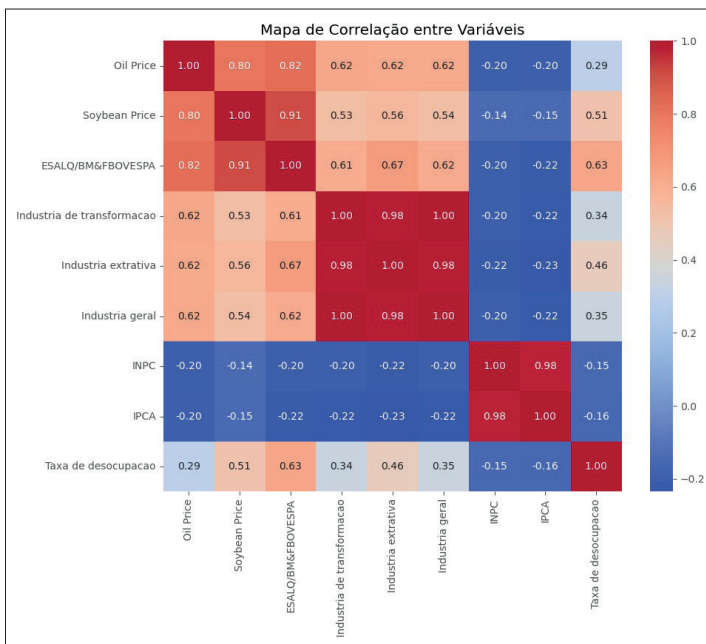
Figura 1 - Base de Dados Principal.

	Country	Year	Month	Oil Price	Soybean Price	ESALQ/BM&FBOVESPA	Industria de transformacao	Industria extrativa	Industria geral	INPC	IPCA	Taxa de desocupacao	Date
223	Brazil	2012	3	123.41	14.03	30.61	114.48887	116.06553	114.59866	0.0018	0.0021	0.08	2012-03-01

Análise Exploratória dos Dados

Foi realizado um mapa de correlação entre todas as variáveis (Figura 2):

Figura 2 - Mapa de Correlação entre as Variáveis.



De acordo com o mapa pode-se pressupor que o preço do petróleo e Soybean Price têm uma correlação positiva de 0.71, indicando que, em geral, os preços de petróleo e soja tendem a se mover na mesma direção, o que pode ser atribuído à interdependência entre os mercados globais de energia e agricultura, como na produção de biocombustíveis.

Soybean Price e ESALQ/BM&FBOVESPA têm uma correlação extremamente forte de 0.94, o que era esperado, dado que ambos medem o preço da soja, ainda que sob diferentes metodologias ou regiões.

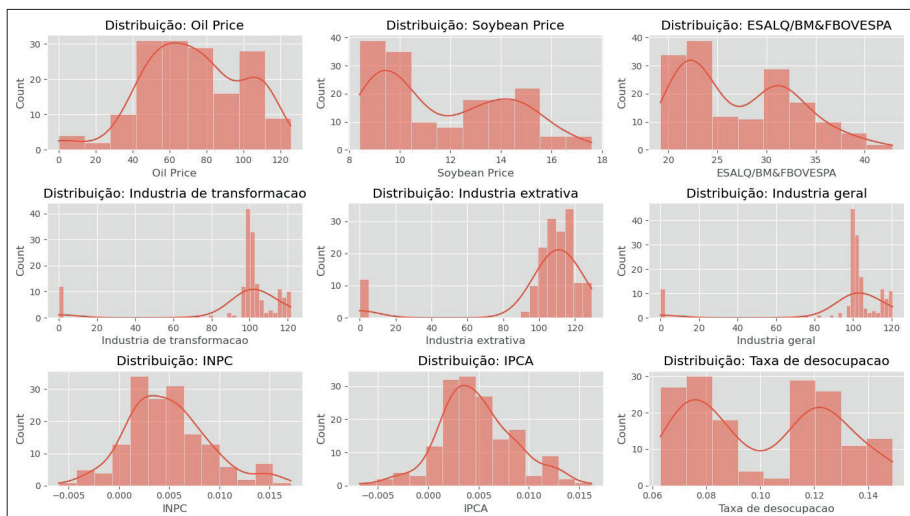
IPCA e INPC, os dois principais índices de inflação, apresentam correlações muito fracas com os preços das commodities, como Soybean Price (IPCA: 0.11, INPC: 0.13). Isso sugere que, no período analisado, a inflação não teve uma influência significativa direta sobre o comportamento dos preços das commodities.

A Taxa de Desocupação apresenta uma correlação negativa moderada com o Oil Price (-0.56), com o Soybean Price (-0.39) e com o índice ESALQ/BM&FBOVESPA (-0.38). Isso indica que, em períodos de aumento no desemprego, é provável que haja uma pressão de baixa nos preços dessas commodities, possivelmente devido à redução da demanda em função de uma economia enfraquecida.

Há uma forte correlação entre as variáveis de setores industriais, com valores próximos de 1.0 entre a Indústria de Transformação, a Indústria Extrativa e a Indústria Geral. Isso sugere que esses setores industriais se movimentam de forma muito alinhada, refletindo que a produção econômica é um fator determinante em várias frentes da indústria.

Também foram analisadas as distribuições de todas as variáveis presente nos dados como mostra a Figura 3.

Figura 3 - Distribuição das Variáveis.



Distribuição: Oil Price (Preço do Petróleo)

- A distribuição dos preços do petróleo apresenta uma leve assimetria positiva, com a maior concentração de valores entre 40 e 80 dólares por barril. A curva de densidade indica que há uma cauda mais longa à direita, sugerindo que os valores mais altos ocorrem com menor frequência. Isso reflete o comportamento dos preços do petróleo, que tendem a permanecer em faixas moderadas, com picos ocasionais devido a crises ou mudanças na oferta e demanda.

Distribuição: Soybean Price (Preço da Soja)

- O preço da soja tem uma distribuição assimétrica, com maior concentração entre 10 e 14 dólares por bushel. A curva de densidade mostra uma cauda à direita, indicando que há episódios menos frequentes de preços elevados, o que pode estar relacionado a fatores climáticos ou de mercado que afetam a oferta.

Distribuição: ESALQ/BM&FBOVESPA (Preço da Soja no Brasil)

- O indicador de preço da soja no Brasil (ESALQ/BM&FBOVESPA) também exibe uma distribuição assimétrica, com a maior concentração de valores em torno de 20 a 25. A curva de densidade evidencia picos em torno desses valores e uma diminuição progressiva para valores maiores. Isso pode refletir a sazonalidade da produção de soja no Brasil e as flutuações do mercado internacional.

Distribuição: Indústria de Transformação

- A produção da indústria de transformação mostra uma distribuição concentrada na faixa de 80 a 120, com uma cauda longa à esquerda. Isso sugere que a produção é relativamente estável em níveis elevados, mas ocasionalmente pode sofrer quedas bruscas, possivelmente devido a crises econômicas ou choques de demanda.

Distribuição: Indústria Extrativa

- A produção da indústria extrativa apresenta uma distribuição moderadamente concentrada entre 80 e 120, com um pico claro em torno de

100. A curva de densidade mostra que a produção desse setor tende a se manter em níveis elevados, mas há uma leve assimetria, indicando que, em alguns períodos, a produção pode ser significativamente menor.

Distribuição: Indústria Geral

- A produção industrial geral segue um padrão muito semelhante ao da indústria de transformação e extrativa, com a maior parte dos valores concentrada entre 80 e 120. Isso reflete uma tendência geral de estabilidade na produção industrial, com variações ocasionais.

Distribuição: INPC

- O Índice Nacional de Preços ao Consumidor (INPC) tem uma distribuição bastante simétrica, com a maior parte dos valores próximos de zero, indicando variações mensais muito pequenas no índice. Isso é esperado, uma vez que o INPC mede a inflação para famílias de baixa renda, que tende a ser moderada em períodos curtos.

Distribuição: IPCA

- A distribuição do Índice de Preços ao Consumidor Amplo (IPCA) também é simétrica e centrada em valores próximos de 0,005. Isso indica que a inflação medida pelo IPCA tende a variar de forma relativamente constante, sem grandes picos, mas com uma leve dispersão em torno do valor central.

Distribuição: Taxa de Desocupação

- A distribuição da taxa de desemprego é bimodal, com dois picos: um em torno de 0,08 e outro em torno de 0,12. Isso sugere que o desemprego apresenta diferentes regimes, possivelmente relacionados a períodos de crise econômica e recuperação. A curva de densidade também mostra uma queda acentuada após o segundo pico, indicando que taxas de desemprego mais elevadas são menos frequentes.

Após a análise exploratória, percebe-se que será necessário realizar um pré-processamento dos dados em termos de normalização das variáveis e selecionar apenas uma variável que diz respeito à produtividade das indústrias

por conta da forte correlação entre elas. A variável escolhida para representar a produtividade das indústrias foi a Indústria Geral.

Pré Processamento do Dados

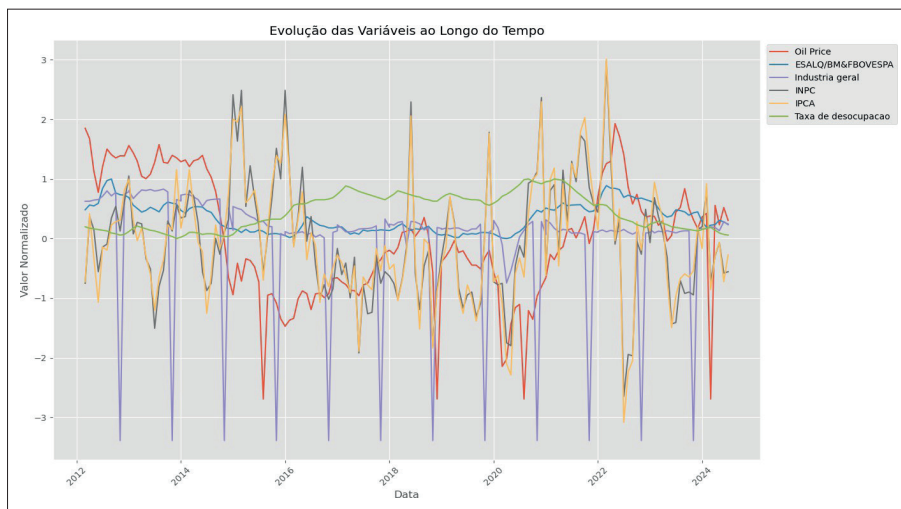
A escolha dos métodos de normalização e padronização é um passo fundamental no pré-processamento dos dados, especialmente em modelos preditivos e de machine learning. Abaixo estão as justificativas para o uso do Min-Max Scaling e da Padronização (Z-Score) nas variáveis selecionadas.

Para as variáveis, ESALQ/BM&FBOVESPA e Taxa de Desocupação foi aplicado o método do Min-Max Scaling. Já para as variáveis, Oil Price, Indústria Geral, INPC e IPCA foi utilizado o método Z-Score.

Min-Max Scaling foi aplicado a variáveis que naturalmente têm um intervalo limitado e cujas diferenças relativas dentro de uma faixa fixa são mais importantes do que desvios em relação à média (Figura 4).

Padronização (Z-Score) foi usada para variáveis com uma maior dispersão ou uma variabilidade considerável, permitindo que os dados fossem centralizados e ajustados à mesma escala, levando em conta a variabilidade das diferentes magnitudes das variáveis.

Figura 4



APLICAÇÃO DO MODELO DE REGRESSÃO LINEAR MULTIVARIADA

Foi aplicada uma Regressão Linear Multivariada para prever o preço da soja (Soybean Price) a partir de diversas variáveis independentes, utilizando a biblioteca Scikit-learn. A regressão linear multivariada é adequada quando se quer modelar a relação entre uma variável dependente (target) e várias variáveis independentes (preditores).

Os dados foram divididos em conjuntos de treino (80%) e teste (20%), utilizando a função `train_test_split`, com o parâmetro `random_state=42` para garantir reprodutibilidade.

Resultados Obtidos

Os resultados obtidos com a aplicação do modelo de regressão linear multivariada para a previsão dos preços da soja são os seguintes:

Erro Quadrático Médio (MSE): 1.0938. Esse resultado sugere que o modelo apresenta um nível aceitável de erro.

Coefficiente de Determinação (R^2): 0.8347. O valor de 0.8347 indica que aproximadamente 83,47% da variação nos preços da soja pode ser explicada pelos fatores incluídos no modelo preditivo. Esse resultado é considerado satisfatório, uma vez que demonstra que o modelo é capaz de capturar bem as flutuações nos preços da soja com base nas variáveis macroeconômicas fornecidas.

CONCLUSÃO

O modelo atual se baseia em variáveis macroeconômicas tradicionais, como preços do petróleo, inflação, produção industrial e taxa de desemprego. No entanto, algumas outras variáveis que influenciam o preço da soja podem ser incluídas no modelo:

Fatores Climáticos: Temperaturas extremas, precipitação, secas e outros eventos climáticos podem ter um impacto significativo nas safras de soja, afetando diretamente os preços. Dados climáticos regionais ou globais poderiam melhorar o modelo.

Dados Geopolíticos e Comerciais: A política internacional e restrições comerciais, como tarifas de exportação ou sanções, afetam os mercados de soja. Incluir dados relacionados a acordos comerciais, conflitos, ou até mesmo políticas governamentais de subsídio poderia enriquecer as previsões.

Índices de Volatilidade (VIX): O índice VIX, que mede a volatilidade do mercado financeiro, pode ser um bom indicador de períodos de instabilidade que afetam o mercado de commodities.

Índices de Demanda Mundial: A inclusão de dados sobre o consumo mundial de soja e produtos derivados poderia ajudar a capturar as tendências globais que influenciam o preço.

A variabilidade nos preços da soja pode não ser constante ao longo do tempo. A utilização de um modelo GARCH (Generalized Autoregressive Conditional Heteroskedasticity) poderia capturar a variabilidade condicional nos dados e melhorar as previsões (Heterocedasticidade).

O modelo de regressão linear multivariada deste trabalho apresentou um bom desempenho, com um R^2 de 0.83, indicando que o modelo explica bem a variação nos preços da soja com base nas variáveis utilizadas. Embora o MSE indique a presença de algum erro nas previsões, o modelo serve como um ponto de partida robusto para a previsão de preços de commodities com base em variáveis macroeconômicas e de mercado.

REFERÊNCIAS

BOLLERSLEV, Tim.(1986). Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, v. 31, n. 3, p. 307-327.

ENGLE, Robert F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica: Journal of the econometric society, p. 987-1007.

ENGLE, Robert F.; HONG, Che-Hsiung; KANE, Alex.(1990). Valuation of variance forecast with simulated option markets.

Frankel, J.A., and Rose, A.K. (2010). Determinants of agricultural and mineral commodity prices. In Inflation in an era of Relative Price Shocks. Reserve Bank of Australia. Gilbert, C.L. (2010). How to understand high food prices. Journal of Agricultural Economics, 61(2), 398-425.

Gargano, A., & Timmermann, A. (2014). Forecasting commodity price indexes using macroeconomic and financial predictors. *International Journal of Forecasting*.

GHYSELS, Eric; SINKO, Arthur; VALKANOV, Rossen.(2007). MIDAS regressions: Further results and new directions. *Econometric reviews*, v. 26, n. 1, p. 53-90.

C. LABYS, Walter. (2003) New directions in the modeling and forecasting of commodity markets. *Monde en développement*, n. 2, p. 3-19..

Smales, L.A. (2017). Commodity market volatility in the presence of US and Chinese macroeconomic news. *Journal of Commodity Markets*, 7, 15-27.

Ye, W., Guo, R., Deschamps, B., Jiang, Y., & Liu, X. (2020). Macroeconomic forecasts and commodity futures volatility. *Economic Modelling*.

APLICAÇÃO DE APRENDIZAGEM DE MÁQUINA PARA PREVER A CRONICIDADE EM PACIENTES COM CHIKUNGUNYA

Tiago Machado de Alencar

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

André Machado de Siqueira

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

chikungunya is an arboviral disease, which can develop into a chronic condition, leading to persistent joint pain and other debilitating symptoms over time. Clinical exams, sociodemographic variables, and specific questionnaires and forms can contribute to predicting chronicity in chikungunya. Multiple classification models were compared to develop a prediction of chronicity in individuals with chikungunya. The analyses suggest that factors such as joint pain, pain intensity, the number of painful joints, and age are the most important attributes in the early detection of chronicity in chikungunya. The main objective of this project is to apply the Random Forest algorithm to predict chronicity and classify the chronic and non-chronic cases of chikungunya.

RESUMO

A chikungunya é uma arbovirose, doença viral transmitida pelos mosquitos *Aedes aegypti* e *Aedes albopictus*, que pode evoluir para uma condição crônica, resultando em dor articular persistente e outros sintomas debilitantes ao longo do tempo. Exames clínicos, variáveis sociodemográficas, e questionários e formulários específicos podem contribuir para a previsão da cronicidade em chikungunya. Múltiplos modelos de classificação foram comparados para desenvolver uma previsão de cronicidade em indivíduos com chikungunya. As análises sugerem que fatores como dor articular, intensidade da dor, quantidade de juntas com dor e idade são os atributos mais importantes na detecção precoce da cronicidade em chikungunya. O principal objetivo deste projeto é aplicar o algoritmo Random Forest para prever a cronicidade e classificar os casos crônicos e não crônicos de chikungunya.

Palavras-chave: previsão de cronicidade em chikungunya; chikungunya; aprendizagem de máquina; saúde pública; autoML, random forest.

INTRODUÇÃO

chikungunya é uma doença viral transmitida aos seres humanos pelos mosquitos *Aedes aegypti* e *Aedes albopictus*, os mesmos vetores responsáveis pela dengue e pelo Zika vírus. Identificada pela primeira vez na África Oriental em 1952, já são conhecidos quatro principais genótipos do vírus chikungunya (CHIKV): o Asiático, o Oeste Africano, o Leste-Central Sul-Africano e o do Oceano Índico. No Brasil, a chikungunya foi introduzida em 2014 e desde então tem se tornado uma preocupação crescente de saúde pública, sendo desde 2016, o país epicentro das epidemias de chikungunya nas Américas, com 1.659.167 casos, o maior número relatado na região [8].

Este fato pode ocorrer devido ao fato de fatores climáticos e ambientais brasileiros, como a precipitação de chuva e clima influenciarem a dinâmica populacional do *Aedes aegypti* [20].

Esta doença é caracterizada por sintomas como febre alta de início súbito e dor articular severa, que muitas vezes acomete múltiplas articulações, sendo esta dor uma das principais marcas da doença. Além disso, outros sintomas comuns incluem dores musculares, cefaleia, náusea, fadiga e erupções cutâneas.

Estudos indicam que cerca de 30% a 40% dos pacientes infectados podem desenvolver sintomas persistentes, especialmente dor articular, que pode evoluir para uma condição crônica, semelhante àquela observada em doenças autoimunes, o que representa um impacto significativo na qualidade de vida dos pacientes (Roth *et al.*, 2014; Simon *et al.*, 2015). Uma revisão de literatura realizada em 316 estudos apontou que a febre foi o sintoma mais reportado, com 57,9% dos estudos, seguido de erupções cutâneas em 54,1%, dor de cabeça em 51,3% e artralgia e 47,8%. Destaca-se que na revisão sistemática a maior média de duração dos sintomas foi de artralgia, de 342 dias [18]. Uma meta análise realizada apontou 43,89% de indivíduos com chikungunya crônica no terceiro mês de sintomas [18]. A cronicidade da chikungunya tem um impacto substancial tanto nos pacientes quanto no sistema de saúde. Para os pacientes, a dor articular prolongada pode levar a limitações físicas, afetando significativamente sua capacidade de realizar atividades diárias e diminuindo sua qualidade de vida. Isso pode resultar em perda de produtividade e em complicações psicológicas, como depressão e ansiedade. Para o sistema de saúde, o tratamento contínuo e o acompanhamento

desses pacientes crônicos representam um desafio adicional, exigindo recursos consideráveis e aumentando a pressão sobre os serviços de saúde pública, especialmente em regiões já sobrecarregadas. Estudos apontam que a infecção por chikungunya crônica pode trazer efeitos na qualidade de vida dos indivíduos, afetando sua saúde mental e física por longos períodos. Prever a cronicidade da chikungunya é de extrema importância, pois permite que intervenções mais precoces e direcionadas possam ser aplicadas, melhorando o manejo clínico e potencialmente reduzindo a carga de morbidade associada à doença. Este estudo visa aplicar técnicas de aprendizagem de máquina para prever a cronicidade em pacientes com chikungunya, classificando os indivíduos crônicos com base nos dados do diagnóstico, com o objetivo de melhorar as estratégias de intervenção e manejo clínico, contribuindo para a redução dos impactos negativos da doença tanto para os pacientes quanto para o sistema de saúde.

Aprendizagem de Máquina em Saúde

Nos últimos anos, a aplicação de técnicas de aprendizagem de máquina (machine learning) na área da saúde tem ganhado destaque, principalmente pela sua capacidade de analisar grandes volumes de dados e identificar padrões complexos que seriam difíceis de detectar por métodos tradicionais. A aprendizagem de máquina tem sido utilizada para diversas finalidades, como a previsão de desfechos clínicos, a personalização de tratamentos, a detecção precoce de doenças e o aprimoramento de diagnósticos. Modelos de aprendizagem de máquina, como redes neurais e outros baseado em construções de Árvores de decisão como o Random forest e Extra Tree, têm demonstrado ser particularmente eficazes na predição de resultados baseados em variáveis clínicas e demográficas, permitindo intervenções mais direcionadas e potencialmente melhorando os resultados para os pacientes. A disponibilidade de grandes quantidades de dados médicos conduziu uma revolução no setor de saúde. Aproveitar as tecnologias de bancos de dados para compartilhar, analisar e processar esses dados tem imenso potencial para identificar padrões de tratamento e reduzir custos de saúde [25]. Iniciativas como o Advance Alert Monitor (AAM) da Kaiser Permanente e o SPOKE também têm demonstrado como a análise preditiva e o gerenciamento integrado de dados podem transformar a prática médica. O AAM, por exemplo, utiliza modelos preditivos avançados para reduzir a mortalidade e melhorar

a eficiência do atendimento hospitalar [16]. O SPOKE, por sua vez, integra milhões de conceitos biomédicos em uma rede de conhecimento, facilitando a descoberta de novos insights clínicos [17]. Este estudo também aborda os tipos mais comuns de aplicações utilizados na área médica, incluindo plataformas amplamente adotadas como o IBM Watson e o Google DeepMind. Essas ferramentas exemplificam o potencial da inteligência artificial na medicina, desde a personalização de tratamentos até a descoberta de novas terapias [1]. Machine learning tem se mostrado crucial no desenvolvimento de modelos preditivos para diagnósticos e tratamentos personalizados. Muniasamy *et al.* discutem como o deep learning está sendo aplicado na análise preditiva de dados de saúde, destacando sua eficácia em diagnósticos médicos, descoberta de medicamentos e personalização de tratamentos [19] Jiang *et al.* exploram o impacto da inteligência artificial, especialmente o machine learning e o processamento de linguagem natural (NLP), na melhoria do tratamento de várias doenças, com ênfase particular no cuidado com o acidente vascular cerebral [12]. No contexto das doenças infecciosas, a aprendizagem de máquina tem sido aplicada para prever a evolução de condições como a dengue, o Zika vírus e a própria chikungunya. Esses modelos têm se mostrado úteis para identificar fatores de risco que podem levar à cronicidade ou a complicações severas, auxiliando profissionais de saúde a tomar decisões informadas sobre o manejo dos pacientes. A capacidade de prever a cronicidade em doenças infecciosas não só ajuda a melhorar a qualidade do cuidado, mas também otimiza a alocação de recursos, uma vez que os pacientes de maior risco podem ser identificados e tratados de forma mais eficaz. Para atender o objetivo do estudo, de classificar indivíduos em crônicos ou não crônicos, trata-se de um aprendizado supervisionado e é necessário uma base de dados anotados, com o histórico da cronicidade de indivíduos e suas características, sociodemográficas e de saúde. Para isso foram utilizados os dados da Replick, Rede de Pesquisa Clínica e Aplicada em chikungunya.

METODOLOGIA

Replick - Rede de Pesquisa Clínica e Aplicada em Chikungunya

A Replick foi criada em 2016 com o objetivo de “promover a integração de expertise e métodos para realizar uma investigação abrangente das manifestações

clínicas da chikungunya, da resposta terapêutica e dos mecanismos de patogênese, a fim de produzir evidências que possam informar e auxiliar na formulação de políticas públicas de saúde” [9]. O estudo desenvolvido pela Replick é de coorte prospectivo, abrangendo as cinco macrorregiões do Brasil, o qual são coletados dados de indivíduos a partir de 18 anos que reportaram febre e artralgia, com suspeita de arbovirose [9]. Os dados dessa coorte incluem informações detalhadas sobre a história clínica dos pacientes, resultados laboratoriais, sintomas apresentados durante a fase aguda e crônica, e características demográficas. A coleta de dados é realizada através de entrevistas, exames clínicos e laboratoriais. Para o presente estudo, foram selecionados indivíduos positivos para chikungunya, que apresentaram informações na coleta d90 (90 dias após a confirmação da infecção).

Machine Learning

Machine Learning (ML) é uma subárea da inteligência artificial que se concentra na criação de algoritmos e modelos que permitem que os sistemas de computador aprendam e façam previsões ou decisões baseadas em dados. Em vez de serem programados explicitamente para realizar uma tarefa, os modelos de ML usam dados históricos para identificar padrões e construir regras que possam ser aplicadas a novos dados. Essa capacidade de aprender a partir de dados e o que torna o machine learning tão poderoso, permitindo que os sistemas melhorem continuamente seu desempenho à medida que recebem mais informações. As aplicações de ML abrangem diversas áreas, incluindo reconhecimento de fala, análise de imagens, sistemas de recomendação e previsão de doenças em saúde pública (Jordan Mitchell, 2015). Prever a cronicidade em indivíduos trata-se de aprendizagem de máquina supervisionada.

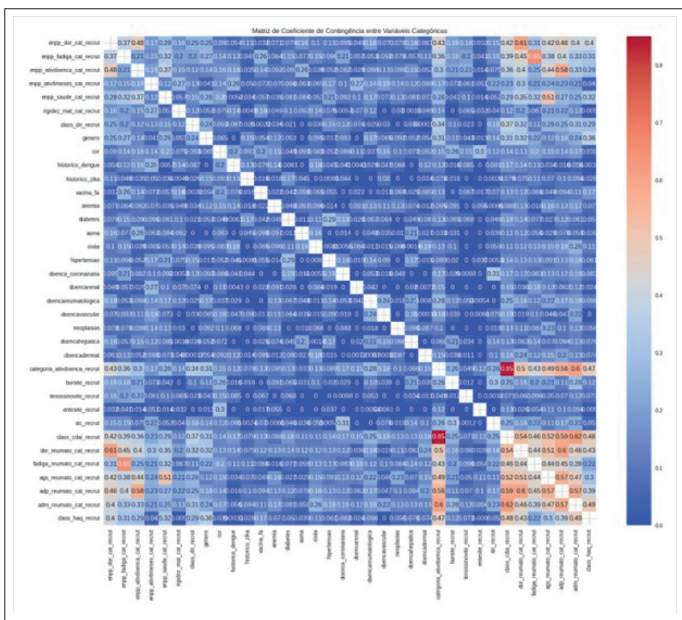
O aprendizado supervisionado é uma das abordagens mais comuns em machine learning, onde o modelo é treinado utilizando um conjunto de dados rotulado, ou seja, onde as entradas são associadas a saídas conhecidas. Durante o processo de treinamento, o algoritmo aprende a mapear as entradas para as saídas, minimizando o erro entre as previsões do modelo e as saídas reais. Uma vez treinado, o modelo pode ser aplicado a novos dados para fazer previsões (Bishop, 2006).

Para o estudo, foi realizado o Auto Machine Learning (AutoML), abordagem que automatiza etapas envolvidas no processo de criação de modelos de machine learning, como a seleção de algoritmos, a otimização de hiperparâmetros e a engenharia de características. Além de simplificar o processo, o AutoML também pode aumentar a eficiência e a rapidez com que os modelos são desenvolvidos. (He; Zhao; Chu, 2021).

Pré-processamento de Dados

A etapa de pré-processamento é um passo fundamental no processo de análise de dados e construção de modelos de machine learning. Ela envolve a preparação e a transformação dos dados brutos para que possam ser usados de forma eficaz pelos algoritmos de aprendizado. Essa etapa é essencial para garantir que os dados estejam limpos, consistentes e no formato adequado para a análise. Cabe ressaltar que o pré-processamento foi realizado nas bases de treino e teste, separadamente para evitar o vazamento de dados. O pré-processamento dos dados foi uma etapa crítica para garantir a qualidade e a eficácia dos modelos de aprendizagem de máquina. Inicialmente, foram selecionados os indivíduos com dados referentes ao d90 presentes, totalizando 289 indivíduos. Em seguida foram realizados testes Qui-quadrado com um nível de significância de 5%, para identificar se as variáveis explicativas apresentavam associação com a variável resposta (cronicidade). Todas as variáveis apresentaram p-valor inferior a 0,05, indicando que houve evidências para rejeitar a hipótese de que as variáveis não são associadas. Após o teste Qui-quadrado, foi realizada uma matriz de coeficientes de contingência entre as variáveis categóricas, para identificar possíveis multicolinearidades.

Figura 1 - Coeficientes de Contingência.



Fonte: Elaborada pelo autor.

Os coeficientes não apontaram associações fortes entre as variáveis e dessa forma foi optado tratar possíveis multico-linearidades dentro do pré-processamento do AutoML. As bases de dados foram divididas em treino e teste, 70% e 30% respectivamente. No processo de AutoML a base de treino ainda foi dividida em treino e validação, para possibilitar o teste de múltiplos modelos e tunar os parâmetros do modelo selecionado sem incorrer em vazamento de dados. Somente ao final da construção do modelo é que as métricas de avaliação foram checadas utilizando a base de testes. Para iniciar a construção dos modelos, foram realizadas as imputações em dados ausentes. Para as variáveis numéricas, utilizou-se a imputação pela média. Já para as variáveis categóricas, a imputação foi feita utilizando-se a moda. A codificação das variáveis categóricas foi realizada utilizando a técnica de *one-hot encoding* para aquelas com múltiplas categorias e a codificação binária para variáveis dicotômicas. Além disso, foi necessário balancear as classes da variável alvo, uma vez que a distribuição entre os pacientes que cronificaram (214, 74%) e os que não cronificaram (75, 26%) estava desequilibrada. Para isso, foram utilizados dois métodos, undersampling e

oversampling. Contudo, os resultados mostram-se mais satisfatórios utilizando o método de sobreamostragem da classe minoritária (*SMOTE - Synthetic Minority Over-sampling Technique*), que cria novas observações sintéticas na classe menos representada, equilibrando assim as proporções. As normalizações das variáveis foram realizadas utilizando zscore e foi realizado o cross-validation para avaliar o desempenho dos modelos de forma mais robusta e evitar problemas de overfitting.

Definição do modelo

Para avaliar o desempenho dos modelos, utilizou-se a validação cruzada em k-folds, onde os dados foram divididos em 10 subconjuntos, avaliando-se as métricas: acurácia, precisão, recall, F1-score e AUC. A acurácia fornece uma medida geral de quantas previsões o modelo acertou. A precisão indica a proporção de verdadeiros positivos em relação ao total de positivos previstos, enquanto o recall mostra a proporção de verdadeiros positivos em relação ao total de positivos reais. O F1-score é a média harmônica entre precisão e recall, fornecendo uma métrica balanceada que é particularmente útil em situações de classes desbalanceadas. A AUC (Área Sob a Curva) fornece uma visão geral do desempenho do modelo em todos os limiares de classificação, mostrando a relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos (1 - especificidade) para diferentes limiares de decisão do modelo. Para o presente estudo é muito importante acertar principalmente os casos positivos, para que os indivíduos possam ter o devido acompanhamento de forma antecipada. Essa característica levou a observar com mais critério o recall e a AUC.

RESULTADOS

O início da construção do modelo de classificação se deu com a comparação entre 14 modelos, listados na tabela I. Os modelos iniciais apresentaram as seguintes avaliações:

Tabela I - Métricas de avaliação dos modelos iniciais.

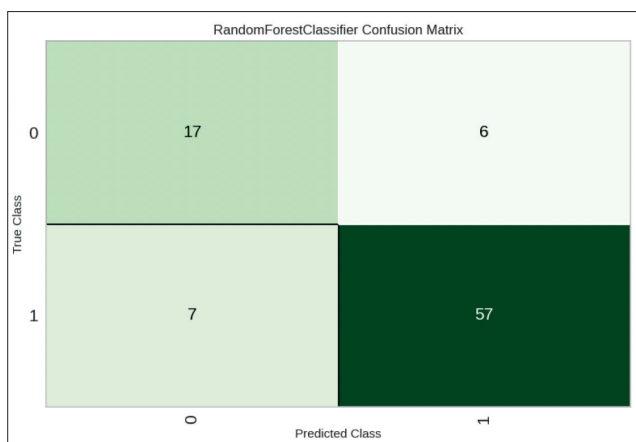
Modelo	Accuracy	AUC	Recall	Prec.
Random Forest Classifier	0.7771	0.782	0.7771	0.7705
K Neighbors Classifier	0.7569	0.773	0.7569	0.7805
Extreme Gradient Boosting	0.7464	0.7107	0.7464	0.7519
Quadratic Discriminant Analysis	0.7326	0.6412	0.7326	0.5802
Ada Boost Classifier	0.7271	0.7056	0.7271	0.7406
Linear Discriminant Analysis	0.7224	0.6524	0.7224	0.7211
Light Gradient Boosting Machine	0.7221	0.7256	0.7221	0.7217
Decision Tree Classifier	0.7217	0.6600	0.7217	0.7448
Logistic Regression	0.7174	0.6893	0.7174	0.7192
Ridge Classifier	0.7126	0.6620	0.7126	0.7092
SVM - Linear Kernel	0.7124	0.6587	0.7124	0.7236
Gradient Boosting Classifier	0.7124	0.7149	0.7124	0.7115
Naive Bayes	0.4757	0.7093	0.4757	0.6942

Fonte: Elaborada pelo autor.

O modelo Random Forest Classifier foi o modelo que apresentou as melhores métricas de avaliação, acertando 77,7% das classificações gerais e 77,7% dos indivíduos crônicos. O Random Forest é um algoritmo de machine learning amplamente utilizado para tarefas de classificação e regressão, devido à sua robustez e capacidade de lidar com conjuntos de dados complexos e de alta dimensionalidade. Desenvolvido por Leo Breiman em 2001, o Random Forest é um tipo de ensemble learning, onde múltiplos modelos fracos, neste caso, árvores de decisão, são combinados para formar um modelo mais forte e preciso (Breiman, 2001). O Random Forest constrói um conjunto ("floresta") de árvores de decisão independentes, onde cada árvore é treinada em um subconjunto aleatório dos dados de treinamento. Além disso, em cada nó da árvore, um subconjunto aleatório das características (features) disponíveis é selecionado para decidir a divisão. Esse processo, conhecido como bagging (bootstrap aggregating), ajuda a reduzir a variância do modelo e a evitar o overfitting, que ocorre quando um modelo se ajusta demais aos dados de treinamento e não generaliza bem para novos dados. Cada árvore no Random Forest fornece uma previsão individual, e a previsão final do modelo é determinada pela agregação dessas previsões individuais. No caso de problemas de classificação, por exemplo, o Random Forest realiza uma votação majoritária, onde a classe mais frequentemente prevista pelas árvores individuais é escolhida como a previsão final. Em problemas

de regressão, a média das previsões das árvores é usada como a previsão final. Após a seleção do modelo Random Forest Classifier, o modelo passou por um fine tuning, onde seus hiperparâmetros foram alterados para verificar se há melhora no desempenho do modelo. Após a alteração dos hiperparâmetros, o modelo apresentou bom desempenho para a base de validação, apresentando a seguinte matriz de confusão.

Figura 2 - Matriz de confusão do modelo Random Forest.



Fonte: Elaborada pelo autor.

o modelo identificou corretamente 17 instâncias da classe 0, Não crônico, (verdadeiros negativos) e 57 instâncias da classe 1, crônico, (verdadeiros positivos). No entanto, houve 7 casos em que a classe 1 foi incorretamente prevista como classe 0 (falsos negativos) e 6 casos em que a classe 0 foi incorretamente prevista como classe 1 (falsos positivos).

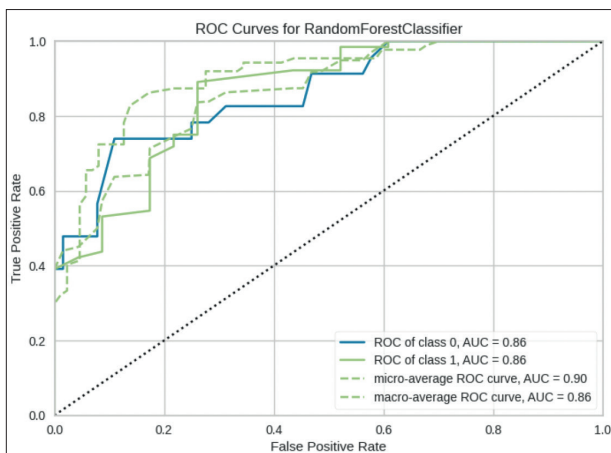
Com base nesses valores, a acurácia do modelo foi de aproximadamente 85%, indicando que o modelo está correto em 85% das suas previsões. A precisão para a classe 1, que mede a proporção de previsões corretas entre todas as previsões feitas para essa classe, foi de cerca de 90,5%. O recall, que mede a capacidade do modelo de identificar corretamente todas as instâncias da classe 1, foi de 89%. O F1-Score, que harmoniza precisão e recall, foi de aproximadamente 89,7%, sugerindo um bom equilíbrio entre esses dois aspectos.

Em resumo, o modelo mostrou um desempenho sólido, especialmente na identificação da classe 1 (Crônico), com uma alta taxa de acerto. No entanto,

ainda houve alguns falsos positivos e falsos negativos, que são áreas potenciais para melhorias futuras no modelo.

O modelo ainda apresentou um bom ajuste, ao observar o AUC.

Figura 3 - Curva ROC do modelo Random Forest.



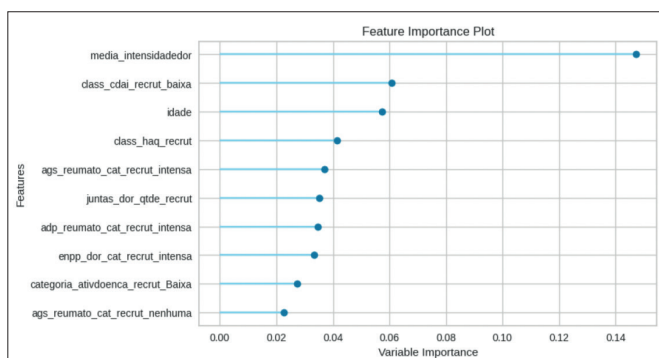
Fonte: Elaborada pelo autor.

Neste gráfico foram apresentadas as curvas ROC para as classes 0 e 1, ambas com uma AUC (Área Sob a Curva) de 0,86, indicando que o modelo tem uma boa capacidade de discriminação entre as classes. A AUC é uma métrica que varia de 0 a 1, onde 1 representa um classificador perfeito e 0,5 representa um classificador que faz escolhas aleatórias. Com uma AUC de 0,86 para ambas as classes, o modelo demonstra ser eficaz na distinção entre as classes, embora ainda haja algum espaço para melhorias.

O gráfico também apresenta a curva ROC média micro (micro-average ROC curve) com uma AUC de 0,90, o que indica que, ao considerar todas as instâncias independentemente da classe, o modelo tem um desempenho ligeiramente melhor. A curva ROC média macro (macro-average ROC curve) também tem uma AUC de 0,86, que é a média das AUCs das duas classes, confirmando o desempenho geral do modelo.

Algumas variáveis apresentaram maior importância no modelo Random Forest, para classificar os indivíduos em crônico ou não crônico, conforme figura 4.

Figura 4 - Importância das variáveis no modelo Random Forest.



Fonte: Elaborada pelo autor.

O modelo apresentou maior importância para 10 variáveis: média de intensidade da dor nas articulações, CDAI (índice clínico da atividade da doença, idade, HAQ (Health Assessment Questionnaire – Avaliação de Capacidade Funcional), Atividade Geral de Saúde Intensa, Quantidade de juntas com dor, ADP, dor intensa, Categoria de atividade da doença e Nenhuma Atividade Geral de Saúde.

Após a validação do modelo, o mesmo foi aplicado aos dados de teste. Aplicar o modelo em uma base de teste que não teve contato com o desenvolvimento do modelo durante o treinamento e validação é crucial para garantir uma avaliação precisa e imparcial de sua capacidade de generalização. Após aplicar o modelo na base de teste, foram obtidas as seguintes métricas:

Tabela II - Métricas de avaliação dos modelos iniciais.

Modelo	Accuracy	AUC	Recall	Prec.	F1
Random Forest Classifier	0.908	0.9667	0.908	0.9104	0.9087

Fonte: Elaborada pelo autor.

O modelo atingiu uma acurácia de 90,8%, o que indica que ele classificou corretamente a grande maioria das instâncias. A AUC, que reflete a capacidade do modelo de discriminar entre as classes, foi de 0,9667, sugerindo uma excelente habilidade em distinguir entre as classes positivas e negativas. O modelo também apresentou um recall de 90,8%, o que significa que ele conseguiu identificar

corretamente a maior parte das instâncias positivas, enquanto a precisão foi de 91,04%, demonstrando que a maioria das instâncias classificadas como positivas realmente pertencem à classe positiva. O F1-Score, que equilibra precisão e recall, foi de 90,87%, indicando que o modelo tem um desempenho bem balanceado, sendo eficaz tanto em identificar instâncias positivas quanto em minimizar falsos positivos. Esses resultados mostram que o modelo ajustado está altamente eficaz em fazer previsões precisas e generalizáveis no conjunto de teste.

CONCLUSÃO

Ao identificar os fatores-chave na previsão da cronicidade em pacientes com chikungunya, podemos melhorar as chances de prognóstico dos indivíduos que apresentarem a classificação de cronicidade. O modelo Random Forest, utilizando dados do momento do diagnóstico foi capaz de acertar com bom desempenho quais indivíduos apresentarão ou não a persistência de artralgia por pelo menos 90 dias.

Para o estudo foram utilizados dados clínicos e dos questionários e formulários aplicados aos pacientes, contudo, outros fatores podem ser determinantes para a cronicidade de chikungunya, como medicamentos tomados e dados laboratoriais. O sintoma utilizado para classificar a cronicidade foi a artralgia, contudo avaliar a persistência de outros sintomas-chave é importante, como a dor neuropática, a rigidez matinal e até a combinação de sintomas. Outro fator importante a ser estudado é quanto tempo os indivíduos crônicos permanecerão com os sintomas. Cada uma dessas características podem influenciar a construção de outros modelos de aprendizagem de máquina para fortalecer as decisões a serem tomadas com cada paciente.

INFORMAÇÕES SUPLEMENTARES

Declarações

Aprovação ética e consentimento para participar

O protocolo foi realizado de acordo com a Declaração de Helsinque. Todos os experimentos serão conduzidos em conformidade com os princípios da Declaração de Helsinque e as diretrizes de Boas Práticas Clínicas da Conferência Internacional de Harmonização. Este estudo foi aprovado pelo Instituto Nacional de Infectologia Evandro Chagas, um Comitê Brasileiro de Ética em Pesquisa com Seres Humanos da Fundação Fiocruz. (CAAI: 07936919.8.1001.5262). O consentimento informado por escrito para participar também foi revisado e aprovado por este Comitê de Ética em Pesquisa com Seres Humanos. Antes de qualquer procedimento do protocolo, os participantes assinaram um formulário de consentimento informado por escrito e darão sua permissão para o estabelecimento de um biobanco.

REFERÊNCIAS

- [1] AHMED, Mohamed Nooman et al. "Cognitive computing and the future of health care cognitive computing and the future of healthcare: the cognitive power of IBM Watson has the potential to transform global personalized medicine". IEEE pulse, v. 8, n. 3, p. 4-9, 2017, doi: 10.1109/MPUL.2017.2678098.
- [2] ALPAYDIN E. Introduction to machine learning. 3rd ed. Cambridge, MA: The MIT Press; 2014.
- [3] Aleixo, R., Kon, F., Rocha, R., Camargo, M. S., & De Camargo, R. Y. (2022). Predicting Dengue Outbreaks with Explainable Machine Learning. 2022 22nd IEEE International Symposium on Cluster Cloud and Internet Computing (CCGrid), Taormina, Italy, 2022, pp. 940-947, doi: 10.1109/CCGrid54584.2022.00114.
- [4] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- [5] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
- [6] Burt, F. J., Chen, W., Miner, J. J., Lenschow, D. J., Merits, A. (2017). chikungunya virus: an update on the biology and pathogenesis of this emerging pathogen. The Lancet Infectious Diseases, 17(4), e107-e117.
- [7] Cleophas TJ. Machine learning in medicine. New York: Springer; 2013.
- [8] De Souza, William M. et al. (2024). chikungunya: a decade of burden in the Americas. The Lancet Regional Health – Americas, Volume 30, 100673.
- [9] DUARTE, Giselle da Silva et al. Multicenter study of the natural history and therapeutic responses of patients with chikungunya, focusing on acute and chronic musculoskeletal manifestations: a study protocol from the clinical and applied research in chikungunya (REPLICK network). BMC Infectious Diseases, v. 23, n. 499, p. 1-12, 2023.

- [10] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24-29.
- [11] He, X., Zhao, K., Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622.
- [12] JIANG, Fei et al. "Artificial intelligence in healthcare: past, present and future". *Stroke and vascular neurology*, v. 2, n. 4, 2017, doi: 10.1136/svn-2017-000101.
- [13] Jordan, M. I., Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
- [14] Kris Rama ,Adrianne M. de Roo,Timon Louwsma,Hinko S. Hofstra,Gabriel S. Gurgel do Amaral,Gerard T. Vondeling,Maarten J. Postma,Roel D. Freriks. Clinical outcomes of chikungunya: A systematic literature review and meta-analysis. *PLOS Neglected Tropical Diseases*, Version 2, Published: June 7, 2024; <https://doi.org/10.1371/journal.pntd.0012254>
- [15] Malley JD, Malley KG, Pajevic S. Statistical learning for biomedical data. Cambridge: Cambridge University Press; 2011.
- [16] MARTINEZ, Vanessa A. et al. "The Kaiser Permanente Northern California Advance Alert Monitor Program: an automated early warning system for adults at risk for in-hospital clinical deterioration". *The Joint Commission Journal on Quality and Patient Safety*, v. 48, n. 8, p. 370-375, 2022, doi: <https://doi.org/10.1016/j.jcjq.2022.05.005>.
- [17] MORRIS, John H. et al. "The scalable precision medicine open knowledge engine (SPOKE): a massive knowledge graph of biomedical information". *Bioinformatics*, v. 39, n. 2, p. btad080, 2023, doi: <https://doi.org/10.1093/bioinformatics/btad080>.
- [18] MORRISON, T. E., Reiter, P. (2016). chikungunya Virus and the Americas—Then and Now. *The American Journal of Tropical Medicine and Hygiene*, 94(4), 525-531.
- [19] MUNIASAMY, Anandhavalli et al. "Deep learning for predictive analytics in healthcare". In: *The International Conference on Advanced Machine Learning Technologies and Applications (AMLT2019)* 4. Springer International Publishing, 2020. p. 32-42, doi: 10.1007/978-3-030-14118-9.
- [20] NATAL, D. Bioecologia do Aedes Aegypti. *Palestra Faculdade de Saúde Pública/USP São Paulo*, v.64, n.2, p.205-207, jul./dez., 2002.
- [21] Obermeyer, Z., Emanuel, E. J. (2016). Predicting the Future—Big Data, Machine Learning, and Clinical Medicine. *The New England Journal of Medicine*, 375(13), 1216-1219.
- [22] Rajkomar, A., Dean, J., Kohane, I. (2018). Machine Learning in Medicine. *The New England Journal of Medicine*, 380, 1347-1358.
- [23] Roth, A., Mercier, A., Lepers, C., Hoy, D., Duituturaga, S., Benyon, E., ... Souares, Y. (2014). Concurrent outbreaks of dengue, chikungunya and Zika virus infections – an unprecedented epidemic wave of mosquito-borne viruses in the Pacific 2012-2014. *Eurosurveillance*, 19(41), 20929.
- [24] Simon, F., Javelle, E., Oliver, M., Leparç-Goffart, I., Marimoutou, C. (2015). chikungunya virus infection. *Current Infectious Disease Reports*, 17(3), 29.
- [25] TOPOL, Eric J. "High-performance medicine: the convergence of human and artificial intelligence". *Nature medicine*, v. 25, n. 1, p. 44-56, 2019, doi: <https://doi.org/10.1038/s41591-018-0300-7>.

UM PROTÓTIPO DE SISTEMA DE APOIO NO CUIDADO DE IDOSOS

Marcus Vinício de Oliveira Braga

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Marcos Ricardo Omena de Albuquerque Maximo

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

Johnny Cardoso Marques

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This work presents an implementation of an information system prototype for Ambient Assisted Living (AAL) capable of real-time identification from any camera with an IP address that implements the Real Time Streaming Protocol (RTSP) possible falls in a monitored and trained environment. After a possible fall, the prototype generates an alert to a registered emergency contact.

RESUMO

Este trabalho apresenta uma implementação de um protótipo de sistema de informação para Ambient Assisted Living (AAL) capaz de identificar em tempo real a partir de qualquer câmera com endereçamento IP que implemente o protocolo Real Time Streaming Protocol (RTSP), possíveis quedas em um ambiente monitorado e treinado. Após uma ocorrência de possível queda, o protótipo gera um alerta para um contato de emergência cadastrado.

VISÃO GERAL DA APRESENTAÇÃO

Contexto

O conceito de *Ambient Assisted Living* (AAL) está relacionado com um conjunto de produtos, serviços e sistemas que procuram manter a qualidade de vida e independência da população sênior e de indivíduos com necessidades de cuidados específicos [Júnior *et al.* 2021].

Segundo Lentzas & Vrakas (2020), as projeções das Nações Unidas estimam que a população mundial aumentará para 9,8 bilhões até 2050, com a população com mais de 60 anos sendo aproximadamente 2,1 bilhões (15% da população mundial), mostrando que a sociedade moderna está envelhecendo em alta velocidade. Esse aumento da média de idade da população traz consigo, consequentemente, o aumento de doenças crônicas, resultando em um crescimento considerável da necessidade de auxílio e assistência médica a este tipo de público [Ras *et al.* 2007].

Os núcleos familiares também tendem a ser decrescentes, por vezes restando ao idoso a permanência em um asilo ou a morar sozinho. Tal informação é de relevância social e de política pública além do grau de criticidade que se encontram essas pessoas. De acordo com Negrini *et al.* (2018), envelhecer morando sozinho, sem o apoio de um parentesco próximo, pode associar-se de maneiras diferentes a vários desfechos de saúde, inclusive a morte.

Solução

Essa solução visa dar qualidade de vida e garantir meios para o idoso poder manter sua intimidade e cotidiano na medida do possível, utilizando visão computacional e abordagens recentes do meio para identificação de classes e objetos.

A partir de uma câmera instalada no ambiente, de forma totalmente transparente aos usuários, uma central de monitoramento autônoma identifica possíveis quedas daquele morador, analisando recorrentemente o ambiente ao qual o protege.

Com a distinção de elementos de características específicas de um ser humano e especialmente entre suas possibilidades de queda, foram criadas duas

classes majoritárias de avaliação. A classe *safe* quando existe um indivíduo no ambiente, porém, em situação normal ou a classe *danger* quando seguindo os exemplos aquele indivíduo está deitado no chão ou em outras características inerentes a queda.

Caso seja verificada classe *danger* o serviço de monitoramento de urgência é acionado imediatamente

Processo adotado

O protótipo teve a redução do escopo para um único cômodo residencial. No local escolhido para monitoramento, foi gravado um vídeo, que posteriormente passou por um tratamento e extração de *frames*. Estes *frames* ocasionaram 1.144 imagens de treinamento rotuladas entre *safe* (estado de normalidade) ou *danger* (estado de alerta de uma possível queda). A Figura 1 (a) apresenta uma das imagens com um possível estado de normalidade (classe *safe*), já a Figura 1 (b) apresenta uma das imagens com um possível estado de alerta (classe *danger*).

Figura 1 - Exemplo de imagens com as classes *safe* (a) e *danger* (b).



A câmera escolhida utiliza o *Real Time Streaming Protocol* (RTSP). Esse protocolo permite, através da consulta a uma URL autenticada, a visualização do *frame* momento da requisição. Isso permite ao serviço responsável pelo monitoramento que consulte esse recurso sistematicamente a cada 60 segundos e o introduza a categorização no *frame* recebido pelas classes já apresentadas.

No sistema de categorização, a abordagem de Rede Neural convolucional (CNN) foi utilizada nesta prototipação, especificamente o algoritmo *Faster R-CNN*, pela possibilidade de realizar, além da classificação das imagens, o local onde o

ser humano se encontra no ambiente. Este algoritmo identifica regiões de maior propensão a serem objetos (*foreground*) ou plano de fundo (*background*).

Devido aos inúmeros cenários e composições de objetos dentre uma cena possíveis, utilizou-se de uma metodologia chamada *Transfer Learning* que permite introduzir um conhecimento prévio na rede. Neste caso foram utilizados objetos comuns do *dataset Common Objects in Context (COCO)*, disponível em <https://cocodataset.org/#home>.

Audiência

O público alvo dessa apresentação é qualquer pessoa interessada em tecnologia com foco em *Ambient Assisted Living (AAL)* e principalmente em querer aplicá-la em fins sociais, especialmente os envolvidos no serviço público e os que se interessam e tem como ramo de estudo a inteligência artificial e visão computacional implementados em sistemas de informação.

Mini biografia dos autores

Marcus Vinício de O. Braga possui bacharelado em Sistemas de Informação, sendo pós-graduado em Inteligência Artificial e Aprendizado de Máquina, ambos pela Pontifícia Universidade Católica de Minas Gerais (PUC-MG). Atualmente cursa a Pós-graduação em Ciência de Dados do Instituto Tecnológico de Aeronáutica (ITA). Trabalha a mais de uma década na Caixa Econômica Federal.

Marcos Ricardo O. de Albuquerque Maximo possui graduação em Engenharia de Computação, mestrado e doutorado em Engenharia Eletrônica e Computação, todos pelo Instituto Tecnológico de Aeronáutica (ITA). é atualmente professor no ITA, onde é membro do Laboratório de Sistemas Computacionais Autônomos (LAB-SCA) e coordena o grupo de competições de robótica ITAndroids.

Johnny Cardoso Marques possui graduação em Engenharia Elétrica e Computação pela Universidade do Estado do Rio de Janeiro (UERJ). é mestre em Engenharia Aeronáutica na área de Sistemas Aeroespaciais e Mecatrônica e doutor em Engenharia Eletrônica e Computação, todos pelo Instituto Tecnológico de Aeronáutica (ITA). é autor várias normas e publicações em engenharia de software para sistemas críticos na área de aviação e saúde.

REFERÊNCIAS

Júnior, M. M. C., Coutinho, W. S., de Alencar, R. L., and Alencar, F. (2021). Um método para modelagem de requisitos não funcionais em ambient assisted living. In *WER*.

Lentzas, A. and Vrakas, D. (2020). Non-intrusive human activity recognition and abnormal behavior detection on elderly people: a review. *Artificial Intelligence Review*, 53:1975–2021.

Negrini, E. L. D., Nascimento, C. F. d., Silva, A. d., and Antunes, J. L. F. (2018). Elderly persons who live alone in brazil and their lifestyle. *Revista Brasileira de Geriatria e Gerontologia*, 21(Rev. bras. geriatr. gerontol., 2018 21(5)).

Ras, E., Becker, M., and Koch, J. (2007). Engineering tele-health solutions in the ambient assisted living lab. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, volume 2, pages 804–809.

UTILIZAÇÃO DE LLMS E GRAPHRAG PARA CONSULTAS NO SETOR FINANCEIRO

Valmir Rodrigues Bueno Júnior

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Lilian Berton

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

In this study, we implemented and evaluated a project utilizing the GraphRAG approach to enhance query-focused processing using large language models (LLMs). Our implementation employed a high-performance computing cluster with 128GB of RAM and an RTX 3090 GPU with 24GB RAM to execute the LLaMA 3.1 8B-Instruct-FP16 and LLaMA 3.1 70B-Instruct-Q4 models. These models generated prompts that the GraphRAG tool used to parse and understand the input document, specifically a schema description of a Brazilian financial institution's transaction database. The integration of these advanced LLMs with GraphRAG facilitated the extraction and analysis of relevant information, showcasing significant improvements in handling complex and large-scale data corpora for comprehensive query responses in the financial sector.

RESUMO

Neste estudo, implementamos e avaliamos um projeto utilizando a abordagem GraphRAG para aprimorar o processamento focado em consultas usando modelos de linguagem de larga escala (LLMs). Nossa implementação empregou um cluster de computação de alto desempenho com 128GB de RAM e uma GPU RTX 3090 com 24GB de RAM para executar os modelos LLaMA 3.1 8B-Instruct-FP16 e LLaMA 3.1 70B-Instruct-Q4. Esses modelos geraram prompts que a ferramenta GraphRAG utilizou para analisar e entender o documento de entrada, especificamente uma descrição de esquema de banco de dados de uma instituição financeira brasileira. A integração desses LLMs avançados com o GraphRAG facilitou a extração e análise de informações relevantes, demonstrando avanços significativos na capacidade de processar e analisar grandes volumes de dados complexos, proporcionando respostas mais precisas e completas para consultas no setor financeiro.

INTRODUÇÃO

Contextualização

O processamento de linguagem natural (PLN) é uma área fundamental da inteligência artificial que visa facilitar a interação entre computadores e a linguagem humana [Jurafsky and Martin 2024]. Com o surgimento dos modelos de linguagem de larga escala (LLMs), como o GPT, houve um avanço significativo na capacidade dessas máquinas de gerar e compreender textos de forma semelhante à humana [Brown *et al.* 2020]. Esses modelos têm revolucionado uma variedade de setores, incluindo a criação de conteúdo e a automação de atendimento ao cliente. No entanto, a tarefa de processar consultas complexas, especialmente em contextos que envolvem grandes volumes de texto, permanece um desafio significativo [Wiseman *et al.* 2017].

Problema de Pesquisa

Retrieval-Augmented Generation (RAG) é uma abordagem inovadora que integra a recuperação de informações com a geração de texto, permitindo que LLMs respondam a perguntas utilizando fontes externas de conhecimento [Lewis *et al.* 2020]. O RAG combina a capacidade de recuperação de dados relevantes com a geração de respostas coerentes e contextualizadas. Apesar de eficaz em muitos cenários, o RAG tradicional enfrenta dificuldades ao lidar com questões globais que exigem uma compreensão abrangente de todo o corpus de texto. Essas questões são críticas em contextos onde a informação está dispersa em grandes bases de dados.

Para superar essas limitações, empregamos a abordagem GraphRAG, que combina o poder do RAG com técnicas de grafos para criar uma indexação baseada em comunidades de entidades [Edge *et al.* 2024]. A abordagem GraphRAG permite uma análise mais aprofundada e estruturada dos dados, facilitando a extração de informações complexas e dispersas em grandes volumes de dados. Este artigo descreve a construção e implementação de um projeto de LLM utilizando a ferramenta GraphRAG, destacando suas aplicações em cenários reais do setor financeiro.

Pretendemos aplicar o GraphRAG em uma base de dados de transações financeiras. A análise de transações financeiras é uma área crítica na segurança financeira, especialmente com o aumento do volume e complexidade das transações digitais. Apesar dos avanços tecnológicos, a análise de transações financeiras permanece desafiadora devido a vários fatores, como:

- **Volume de dados:** Grandes volumes de dados exigem sistemas altamente escaláveis e eficientes, capazes de identificar padrões relevantes sem causar atrasos nas operações.
- **Diversidade de transações:** Sistemas de análise precisam ser adaptáveis e capazes de reconhecer diferentes tipos de transações, o que exige uma combinação de técnicas de aprendizado de máquina, análise de redes e técnicas estatísticas.
- **Dados desbalanceados:** Em transações financeiras, a maioria das transações são legítimas, enquanto as que requerem atenção representam uma pequena fração do total. Esse desbalanceamento torna a construção de modelos analíticos desafiadora, pois os algoritmos podem se tornar tendenciosos para as classes majoritárias.

Objetivo Geral

O objetivo deste projeto é explorar e avaliar a eficácia do GraphRAG na tarefa de processar consultas complexas, utilizando um cluster de alta performance com 128GB de RAM e uma placa de vídeo RTX 3090. Executamos os modelos LLaMA 3.1 8B-Instruct-FP16 e LLaMA 3.1 70B-Instruct-Q4 para gerar os prompts necessários que a ferramenta GraphRAG utiliza para entender e processar o arquivo, que neste caso era um esquema de descrição de um banco de dados de uma empresa do mercado financeiro. Esse enfoque específico demonstra a aplicabilidade da nossa abordagem em cenários do mundo real onde a precisão e a abrangência das respostas são essenciais.

Objetivos Específicos

Os objetivos específicos deste estudo são:

- Desenvolver e implementar prompts específicos para a ferramenta GraphRAG interpretar e processar o esquema de descrição do banco de dados financeiro.
- Executar a ferramenta GraphRAG para processar consultas complexas, avaliando a abrangência, precisão e diversidade das respostas geradas.

REVISÃO DA LITERATURA

Avanços em Processamento de Linguagem Natural (PLN)

O processamento de linguagem natural (PLN) tem evoluído consideravelmente nas últimas décadas, impulsionado principalmente pelos avanços em modelos de linguagem de larga escala (LLMs) [Jurafsky and Martin 2024]. Modelos como BERT [Devlin *et al.* 2019], GPT-3 [Brown *et al.* 2020] e LLaMA [Touvron *et al.* 2023] revolucionaram a capacidade das máquinas de compreender e gerar linguagem natural, permitindo avanços significativos em tarefas como tradução automática, resposta a perguntas e análise de sentimentos. Esses modelos, treinados em vastos conjuntos de dados textuais, têm mostrado uma habilidade excepcional para gerar texto coerente e contextualizado, o que tem impulsionado a adoção generalizada dessas tecnologias em uma ampla gama de setores.

Geração Aumentada por Recuperação (RAG)

A abordagem de Retrieval-Augmented Generation (RAG) em português Geração Aumentada por Recuperação combina técnicas de recuperação de informações com geração de texto, proporcionando uma forma eficiente de LLMs responderem a perguntas utilizando fontes externas de conhecimento [Lewis *et al.* 2020]. Essa metodologia tem sido particularmente eficaz em contextos onde a informação relevante está dispersa em várias fontes, permitindo que os LLMs acessem e integrem dados de maneira mais precisa. No entanto, o RAG enfrenta limitações ao lidar com questões que exigem uma compreensão global do texto, devido à sua dependência de recuperações pré-processadas [Gua *et al.* 2020].

Técnicas de Grafos e a Inovação do GraphRAG

Técnicas de grafos têm sido amplamente utilizadas para estruturar e analisar dados complexos, oferecendo uma maneira poderosa de representar relações entre entidades em um corpus de texto [Kipf and Welling 2017]. A inovação trazida pela combinação dessas técnicas com LLMs é exemplificada na metodologia GraphRAG, que integra a geração aumentada por recuperação com grafos de conhecimento [Edge *et al.* 2024]. Ao construir um índice de texto baseado em grafos, o GraphRAG permite uma análise mais estruturada e profunda dos dados, facilitando a extração de informações complexas de maneira mais coerente e abrangente [Alghisi *et al.* 2024].

Estudos recentes demonstram que o GraphRAG supera as limitações do RAG tradicional em tarefas que exigem uma compreensão global do corpus, como a geração de respostas a partir de grandes volumes de dados textuais [Edge *et al.* 2024]. Essa abordagem tem mostrado resultados promissores em diversos experimentos, destacando-se pela sua capacidade de integrar informações dispersas e fornecer respostas mais completas e diversificadas.

Aplicações no Setor Financeiro

No setor financeiro, a aplicação de LLMs e técnicas de grafos, como o GraphRAG, tem o potencial de transformar a análise de grandes volumes de dados transacionais. A complexidade e o volume dos dados financeiros exigem sistemas que possam lidar eficientemente com a recuperação e análise de informações relevantes, fornecendo insights acionáveis em tempo real [Bughin *et al.* 2017]. A capacidade do GraphRAG de integrar e analisar dados dispersos é particularmente relevante nesse contexto, permitindo uma análise mais precisa e abrangente das transações financeiras, o que é essencial para tarefas como a detecção de padrões e a tomada de decisões estratégicas [Wang *et al.* 2024].

Desafios e Limitações

Apesar dos avanços significativos, ainda existem desafios consideráveis a serem superados na aplicação de LLMs e técnicas de grafos, especialmente

em contextos de alto volume de dados, como o setor financeiro. A escalabilidade dos modelos, a precisão na recuperação de informações, e a necessidade de integração com sistemas de legado são algumas das principais dificuldades que precisam ser abordadas [Li *et al.* 2024]. Além disso, a questão da explicabilidade dos modelos, que é crítica em setores regulados como o financeiro, permanece um tópico de pesquisa em aberto [Samek *et al.* 2017].

METODOLOGIA

Nesta seção, detalhamos o processo completo de utilização do GraphRAG para analisar o esquema de descrição de um banco de dados financeiro. Este processo incluiu a preparação do ambiente, a execução do GraphRAG, e a análise das entidades e relacionamentos gerados a partir dos dados. Também apresentamos as imagens da estrutura e das comunidades criadas pelo GraphRAG.

Cluster de Computação de Alto Desempenho

Um cluster de computação de alto desempenho é um conjunto de computadores interconectados que trabalham juntos como se fossem um único sistema [Buyya 1999]. Esses clusters são essenciais para executar tarefas que exigem grande poder computacional, como o treinamento de modelos de linguagem de larga escala (LLMs) [Shoeybi *et al.* 2019].

No nosso projeto, o cluster foi configurado com:

- **Hardware:** Utilizamos 128GB de RAM para garantir o processamento eficiente de grandes volumes de dados e uma placa de vídeo RTX 3090 para suportar tarefas de deep learning com alta performance em processamento paralelo.
- **Software:** O sistema operacional escolhido foi o Ubuntu 20.04 LTS, devido à sua estabilidade e suporte a ferramentas de desenvolvimento [Negus *et al.* 2008]. Para maximizar o uso da GPU, utilizamos o CUDA [Nickolls *et al.* 2008], e o Docker e Docker Compose foram empregados para criar e gerenciar contêineres, assegurando a consistência do ambiente de desenvolvimento [Merkel 2014].

GraphRAG e Prompts Específicos

GraphRAG é uma metodologia que integra a geração aumentada por recuperação (RAG) com técnicas de grafos para melhorar a compreensão e o processamento de consultas complexas [Edge *et al.* 2024].

No projeto, desenvolvemos prompts específicos para que o GraphRAG possa interpretar e processar consultas com base em um esquema de descrição de um banco de dados financeiro. Esses prompts são cruciais para orientar o GraphRAG na geração de respostas precisas e contextualizadas [Guu *et al.* 2020].

O prompt gerado podem ser encontrados no Apêndice A.

Configuração do Ambiente

Para suportar a execução do GraphRAG com os modelos LLaMA 3.1 8B-Instruct - FP16 e LLaMA 3.1 70B-Instruct-Q4, configuramos um ambiente de computação de alto desempenho. A configuração incluiu:

- **Memória RAM:** 128GB para garantir a capacidade de processar grandes volumes de dados.
- **Placa de Vídeo:** RTX 3090 para fornecer o poder de processamento paralelo necessário para tarefas intensivas de deep learning.
- **Sistema Operacional:** Ubuntu 20.04 LTS, por sua estabilidade e compatibilidade com ferramentas de desenvolvimento.
- **CUDA e Docker:** Utilizamos CUDA para otimizar o uso da GPU e Docker para criar um ambiente de execução isolado e consistente.

Processo de Utilização do GraphRAG

O processo de uso do GraphRAG para analisar o banco de dados foi realizado em várias etapas:

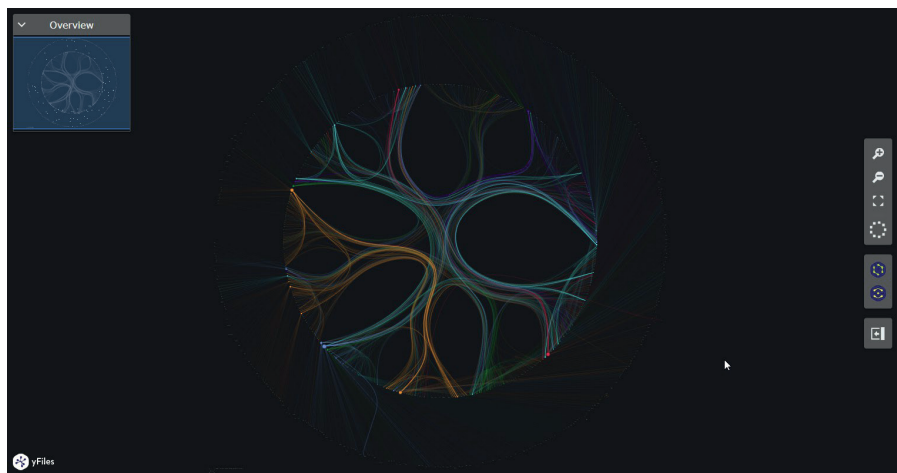
Inserção dos Dados

Os dados da descrição do banco de dados financeiro foram inseridos como input no GraphRAG. Esse input incluiu informações detalhadas sobre as entidades do banco de dados, suas relações e os metadados associados.

Execução do GraphRAG

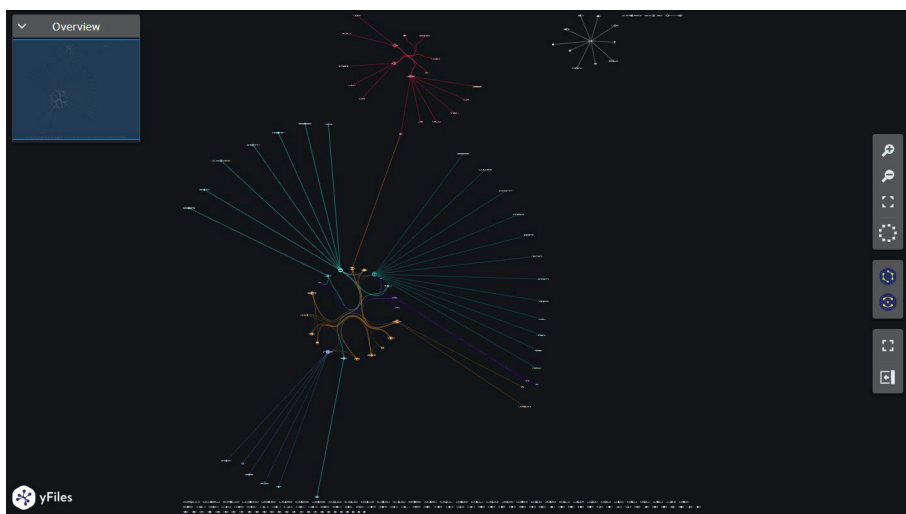
O GraphRAG foi executado para processar o input e gerar um conjunto de entidades e relacionamentos. Durante essa etapa, o GraphRAG construiu uma estrutura gráfica que mapeia as entidades e suas conexões.

Figura 1 - Estrutura gráfica completa criada pelo GraphRAG.



Fonte: Elaborado pelo Autor.

Figura 2 - Comunidades de entidades criadas pelo GraphRAG.

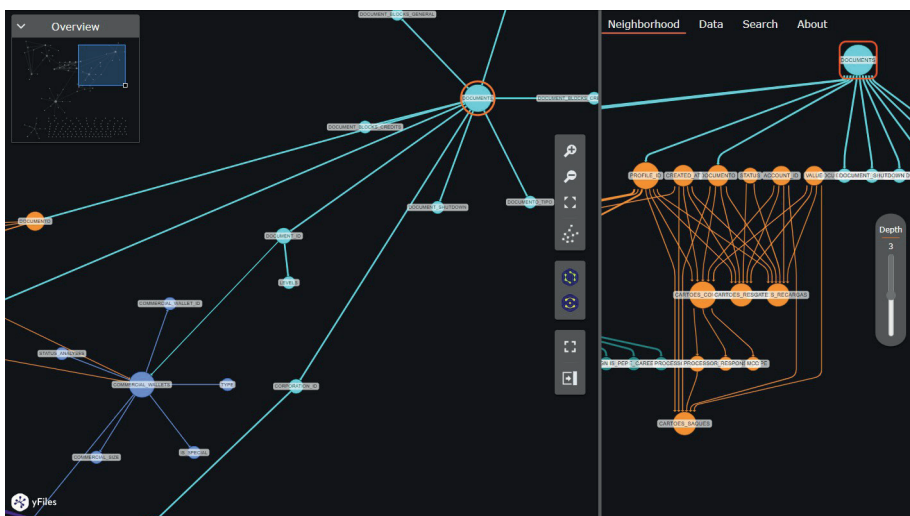


Fonte: Elaborado pelo Autor.

Análise das Comunidades Criadas

Após a construção da estrutura gráfica, o GraphRAG identificou várias comunidades de entidades que estão fortemente inter-relacionadas. Essas comunidades foram analisadas para entender melhor as relações subjacentes entre as entidades. As

Figura 3 - Zoom em uma das comunidades criadas pelo GraphRAG.



Fonte: Elaborado pelo Autor.

Figuras 1, 2 e 3 apresentam algumas visualizações das comunidades.

Validação e Ajustes Finais

Após a análise das comunidades, realizamos ajustes nos prompts e parâmetros do GraphRAG para garantir que as entidades e relações fossem capturadas e representadas com a maior precisão possível. Testes preliminares foram conduzidos para validar a eficácia do modelo, e ajustes foram feitos conforme necessário.

Ajustes de Parâmetros

Para otimizar o desempenho durante a inferência dos modelos e a geração das estruturas gráficas, ajustamos diversos parâmetros críticos do GraphRAG. Esses ajustes foram realizados para melhorar a qualidade das respostas geradas e a eficiência do processamento. A seguir, descrevemos os principais parâmetros ajustados e as alterações realizadas:

Parâmetros de Local Search: Os parâmetros de Local Search controlam como o GraphRAG processa e analisa as comunidades de entidades em um nível mais local. Os ajustes realizados foram:

- **text_unit_prop:** Configurado para 0.5, esse parâmetro define a proporção das unidades de texto que são consideradas para o processamento.
- **community_prop:** Ajustado para 0.1, este parâmetro determina a proporção das comunidades a serem analisadas, focando nas mais relevantes.
- **conversation_history_max_turns:** Mantido em 5, controla o número máximo de turnos de histórico de conversação que o modelo considera, garantindo um contexto suficiente sem sobrecarregar o modelo.
- **top_k_mapped_entities:** Definido como 10, limita o número de entidades mapeadas a serem consideradas nas principais respostas.
- **top_k_relationships:** Também definido como 10, limita o número de relacionamentos a serem considerados.
- **llm_temperature:** Ajustado para 0, o que indica uma amostragem determinística, sem diversidade, para garantir consistência nas respostas.
- **llm_top_p:** Mantido em 1, permitindo que todos os tokens sejam considerados na amostragem.
- **llm_n:** Configurado para 1, gerando uma única conclusão por vez.
- **max_tokens:** Ajustado para 12000, permitindo um maior número de tokens processados por vez, o que é crucial para análises mais detalhadas.

Parâmetros de Global Search: Os parâmetros de Global Search afetam a forma como o GraphRAG realiza a busca e processamento em um nível mais global, abrangendo toda a estrutura de dados. Os ajustes realizados incluem:

- **llm_temperature:** Semelhante ao Local Search, foi ajustado para 0 para garantir consistência nas respostas.
- **llm_top_p:** Mantido em 1, permitindo uma consideração completa de todos os tokens.
- **llm_n:** Configurado para 1, gerando uma única conclusão.

- **max_tokens:** Ajustado para 12000, garantindo que um número suficiente de tokens possa ser processado em análises mais amplas.
- **data_max_tokens:** Configurado para 12000, igual ao max_tokens, para garantir a uniformidade no processamento.
- **map_max_tokens:** Definido em 1000, controla o número máximo de tokens durante a etapa de mapeamento, ajudando a concentrar o processamento em porções mais manejáveis.
- **reduce_max_tokens:** Ajustado para 2000, permitindo um número maior de tokens durante a etapa de redução, essencial para sintetizar informações de múltiplas fontes.
- **concurrency:** Configurado para 32, o que aumenta a capacidade de processamento simultâneo, permitindo uma análise mais rápida e eficiente dos dados.

Esses ajustes foram essenciais para refinar o desempenho do GraphRAG e melhorar a qualidade das respostas geradas durante os testes. Através de iterações sucessivas e validação dos resultados, conseguimos otimizar o comportamento do modelo, alcançando um equilíbrio entre precisão, eficiência e qualidade das respostas.

Testes de Validação

Finalmente, executamos uma série de testes para validar os resultados gerados pelo GraphRAG, garantindo que as respostas fossem precisas e relevantes para o contexto financeiro em análise.

Questões para recuperação de informações

Formulamos perguntas de forma a testar diversos aspectos do modelo, como precisão na recuperação de informações, coerência na geração de texto, relevância dos conteúdos recuperados, entre outros.

O trabalho de [An *et al.* 2024] aborda o uso de recuperação de informação em GraphRag, no trabalho é comentado sobre “A Ética a Nicômaco de Aristóteles” que introduziu o conceito de perguntas “5Ws e 1H” para compreender de forma

abrangente as ações e suas circunstâncias, onde os “5Ws” representam O quê, Por quê, Quando, Onde e Quem, e o “1H” representa Como. Essas perguntas servem como ferramentas fundamentais para abordar vários aspectos dos eventos e do comportamento humano, abrangendo desde discursos filosóficos até questões do cotidiano. Desse modo, a fim de validar o GraphRAG testamos as seguintes questões relacionadas com a base de dados:

- **O quê:**

- Quais tipos de transações estão incluídas no conjunto de dados?
- Quais são as características comuns das transações fraudulentas?
- Quais recursos estão disponíveis nos dados das transações (por exemplo, valor, data, hora, localização, comerciante)?

- **Quem:**

- Quem são os titulares das contas envolvidas nas transações?

Como:

- Como podemos distinguir entre transações fraudulentas e não fraudulentas?
- Como as transações são monitoradas e marcadas como potenciais fraudes?
- Como as características das transações fraudulentas diferem das não fraudulentas?
- Quão eficazes são os algoritmos de detecção de fraudes atuais e como pode ser utilizado nesse dataset?

RESULTADOS

A seguir apresentamos as respostas geradas para cada pergunta e a validação manual realizada pelo autor do trabalho sobre a corretude da resposta gerada.

Pergunta 1

Quais tipos de transações estão incluídas no conjunto de dados?

Resposta

Categorias de Transações Financeiras

O conjunto de dados abrange uma variedade de categorias de transações financeiras, refletindo a diversidade de operações realizadas pelos usuários na plataforma. Essas transações são registradas em várias tabelas interconectadas, permitindo uma análise detalhada e segmentada de cada tipo de operação. As principais categorias de transações incluem compras com cartão, transferências bancárias, resgates, saques, e pagamentos de contas, cada uma com suas próprias características e particularidades [Data: Reports (14, 2, 11, 6, 13)].

Transações com Cartões

As transações com cartões de crédito e débito são uma das principais categorias incluídas no conjunto de dados. Essas transações envolvem compras em pontos de venda físicos e online, onde o valor da compra é debitado diretamente da conta do titular. Além disso, o dataset também inclui informações sobre recargas de celular e pagamentos de contas realizados por meio de cartões. Essas operações são fundamentais para entender os padrões de consumo dos usuários e identificar possíveis anomalias [Data: Reports (13, 3, 9, 5, +more)].

Transferências Bancárias (PIX, TED, DOC)

Outro tipo significativo de transação registrado no dataset é a transferência bancária, que pode ser realizada através de diferentes meios, como PIX, TED, e DOC. As transferências PIX, por exemplo, são instantâneas e têm se tornado cada vez mais comuns, enquanto TED e DOC seguem processos mais tradicionais com prazos específicos de compensação. Cada uma dessas transações é rastreada com detalhes sobre o remetente, destinatário, valor, e tempo de processamento, permitindo uma análise precisa de fluxos de caixa e comportamento de transferência [Data: Reports (12, 4, 10, 6)].

Resgates e Saques

O conjunto de dados também inclui registros de resgates e saques. Resgates referem-se à conversão de saldos disponíveis na plataforma em dinheiro, que é então transferido para a conta bancária do usuário. Saques, por outro lado, representam retiradas de dinheiro realizadas em caixas eletrônicos utilizando os cartões associados à conta. Esses tipos de transações são essenciais para monitorar o fluxo de fundos e garantir a segurança das operações de retirada [Data: Reports (11, 6, 8)].

Pagamentos de Contas e Boletos

Além das compras e transferências, o dataset também cobre pagamentos de contas e boletos. Essas transações são registradas quando os usuários utilizam a plataforma para pagar suas contas de serviços públicos, como energia elétrica, água, ou telecomunicações, além de boletos de outros serviços. A inclusão desses pagamentos no dataset permite uma visão completa do comportamento financeiro do usuário, considerando tanto as despesas discricionárias quanto as obrigatórias [Data: Reports (15, 5, 14)].

Movimentações Internas e Transações Corporativas

Em casos onde o usuário é uma empresa ou possui múltiplas contas associadas ao mesmo perfil, o dataset inclui movimentações internas e transações corporativas. Estas transações podem envolver transferências entre contas de uma mesma entidade ou pagamentos de fornecedores. A análise dessas operações é fundamental para entender o comportamento financeiro de grandes contas corporativas e para assegurar a conformidade com as políticas financeiras da instituição [Data: Reports (10, 2, 13)].

Em resumo, o conjunto de dados inclui uma ampla gama de transações financeiras que abrangem desde compras com cartão até pagamentos de contas e movimentações corporativas. Cada tipo de transação oferece insights valiosos sobre o comportamento dos usuários e é essencial para uma análise financeira abrangente.

Pergunta 2

Quais são as características comuns das transações fraudulentas?

Resposta

Padrões de Comportamento em Transações Fraudulentas

As transações fraudulentas geralmente seguem certos padrões de comportamento que as distinguem das operações legítimas. Esses padrões incluem uma combinação de valores anômalos, frequência elevada de transações em curto período de tempo, utilização de dispositivos não reconhecidos, e a realização de operações em locais geograficamente distantes do habitual. A identificação desses padrões é crucial para a detecção e prevenção de fraudes dentro da plataforma [Data: Reports (14, 2, 11, 6, 13)].

Valores Anômalos e Frequência Elevada

Uma característica comum das transações fraudulentas é a presença de valores que são significativamente diferentes dos padrões normais do titular da conta. Isso pode incluir tanto transações de valor muito elevado quanto transações pequenas e repetitivas que visam passar despercebidas. Além disso, transações fraudulentas frequentemente ocorrem em uma sequência rápida, com múltiplas operações sendo realizadas em um curto intervalo de tempo, o que contrasta com o comportamento usual mais espaçado e regular [Data: Reports (13, 3, 9, 5, +more)].

Dispositivos e Localizações Não Reconhecidos

Fraudes geralmente envolvem o uso de dispositivos que não foram previamente associados ao titular da conta, como novos smartphones ou computadores, especialmente em casos onde os dispositivos estão localizados em regiões geográficas distantes do local habitual de uso do titular. A combinação de um dispositivo desconhecido e uma localização inusitada é um forte indicativo de fraude, especialmente quando as transações são realizadas em horários atípicos [Data: Reports (12, 4, 10, 6)].

Análise Temporal e Horários Atípicos

Outra característica marcante das transações fraudulentas é o horário em que elas ocorrem. Transações fraudulentas são frequentemente realizadas durante a madrugada ou em horários onde o titular normalmente não realiza operações financeiras. Essa mudança no padrão de comportamento temporal, associada a outros fatores como localização e dispositivo, aumenta significativamente a probabilidade de que a transação seja fraudulenta [Data: Reports (11, 6, 8)].

Comerciantes e Categorias de Despesas Inusitadas

Fraudes também podem ser identificadas pela análise dos comerciantes envolvidos e das categorias de despesas. Transações fraudulentas podem ocorrer em estabelecimentos ou categorias que não fazem parte do perfil usual de gastos do titular da conta. Por exemplo, uma compra de alto valor em uma loja de eletrônicos pode ser suspeita se o titular normalmente faz apenas compras de rotina em supermercados ou farmácias. Essas discrepâncias entre o comerciante e o histórico de despesas do titular são cruciais para a detecção de fraudes [Data: Reports (15, 5, 14)].

Comportamento Geral do Titular

Finalmente, o comportamento geral do titular, incluindo mudanças súbitas no padrão de gastos ou na frequência de transações, pode indicar atividades fraudulentas. Um aumento repentino no número de transações, especialmente em categorias de despesas incomuns, é um sinal de alerta. A análise desses comportamentos anômalos permite que a plataforma reaja rapidamente para prevenir ou mitigar os efeitos de fraudes [Data: Reports (10, 2, 13)].

Em resumo, as transações fraudulentas tendem a exibir um conjunto de características distintas, incluindo valores anômalos, uso de dispositivos não reconhecidos, realização em locais geograficamente distantes, e em horários atípicos. A identificação e monitoramento desses padrões são essenciais para proteger a integridade financeira da plataforma e de seus usuários.

Pergunta 3

Quais recursos estão disponíveis nos dados das transações (por exemplo, valor, data, hora, localização, comerciante)?

Resposta

Estrutura dos Recursos Disponíveis

O dataset das transações financeiras possui uma ampla variedade de recursos que são essenciais para a análise detalhada de cada operação realizada. Esses recursos estão distribuídos por diversas tabelas interconectadas, que oferecem uma visão abrangente e multifacetada das atividades financeiras na plataforma. A interconexão desses recursos permite não apenas a análise isolada de cada transação, mas também a compreensão do comportamento geral dos usuários ao longo do tempo [Data: Reports (13, 3, 9, 5, 7)].

Valor da Transação e Análise de Gastos

O valor da transação é um dos recursos mais críticos disponíveis no dataset. Cada operação financeira registrada inclui um campo específico que detalha o valor monetário envolvido. Esse recurso é fundamental para a realização de análises de padrão de gastos, onde os usuários podem ser categorizados com base em suas atividades financeiras. Além disso, o valor da transação é frequentemente utilizado para identificar possíveis atividades suspeitas, como transações com valores fora do padrão para um determinado perfil [Data: Reports (14, 2, 11, 6, 13)].

Data e Hora da Transação

As informações de data e hora são capturadas com precisão em cada transação, permitindo que análises temporais sejam conduzidas. A distribuição das transações ao longo do dia, da semana ou do mês pode revelar padrões de comportamento, como horários de pico de atividade ou períodos de inatividade. Estes dados temporais também são essenciais na detecção de fraudes, onde transações em horários atípicos podem indicar atividades suspeitas [Data: Reports (12, 3, 10, 5, 9)].

Localização Geográfica das Transações

A localização geográfica onde uma transação foi realizada é outro recurso valioso. Este dado pode ser utilizado para mapear o comportamento dos usuários em diferentes regiões ou para identificar transações que ocorrem em locais inusitados. Por exemplo, a ocorrência de transações em locais geograficamente distantes em um curto período de tempo pode ser um forte indicativo de fraude. A análise geográfica das transações também permite à instituição ajustar suas ofertas de acordo com as preferências regionais dos usuários [Data: Reports (11, 6, 10)].

Identificação do Comerciante e Categorias de Despesas

Cada transação no dataset é associada a um comerciante específico, cujo identificador único está presente nos dados. Esse recurso é utilizado para agrupar e categorizar despesas de acordo com os tipos de comerciantes, facilitando a análise de preferências de consumo. Além disso, a identificação do comerciante pode ser útil na detecção de fraudes, onde transações realizadas com comerciantes incomuns para um determinado perfil podem levantar alertas [Data: Reports (12, 4, 8, 7, +more)].

Métodos de Pagamento e Análise de Preferências

O dataset também inclui informações sobre os métodos de pagamento utilizados em cada transação, como cartões de crédito, débito, transferências bancárias, ou PIX. Este recurso permite que a instituição financeira analise as preferências dos usuários por diferentes métodos de pagamento, oferecendo insights para a personalização de serviços e produtos. Além disso, métodos de pagamento específicos podem ser monitorados para detectar possíveis anomalias ou tendências emergentes no comportamento do usuário [Data: Reports (13, 3, 9, 5)].

Integração dos Recursos para Análise Holística

A integração desses recursos dentro do dataset permite uma análise holística das transações financeiras. Ao combinar informações de valor, data, hora, localização, comerciante e método de pagamento, é possível construir

perfis detalhados de comportamento do usuário, identificar padrões de fraude, e realizar análises preditivas que ajudam a instituição a antecipar tendências e mitigar riscos [Data: Reports (15, 5, 14)].

Em resumo, os recursos disponíveis no dataset de transações são essenciais para a análise detalhada e a compreensão do comportamento financeiro dos usuários. Esses recursos permitem uma variedade de análises que vão desde o monitoramento de padrões de gastos até a detecção de fraudes e a personalização de ofertas e serviços.

Pergunta 4

Quem são os titulares das contas envolvidas nas transações?

Resposta

Identificação dos Titulares de Contas

O dataset contém informações detalhadas sobre os titulares das contas envolvidas em cada transação financeira. Esses titulares são identificados principalmente através de dados vinculados aos perfis de usuários, que estão associados diretamente a cada conta registrada na plataforma. A interconexão entre as tabelas de 'PROFILES', 'ACCOUNTS', e 'TRANSACTIONS' permite que a instituição rastreie e analise quem está por trás de cada operação financeira registrada [Data: Reports (14, 2, 11, 6, 13)].

Perfis de Usuários e Associação com Contas

Cada conta no sistema está associada a um perfil de usuário único, identificado por um PROFILE_ID. Este identificador é o principal vínculo entre o usuário e as contas que ele controla. Dentro da estrutura de dados, os perfis de usuários são enriquecidos com informações como nome, CPF/CNPJ, e dados de contato, o que facilita a identificação precisa dos titulares das contas envolvidas em qualquer transação específica [Data: Reports (13, 3, 9, 5, +more)].

Documentação Associada aos Titulares

Os titulares das contas também são identificados por meio de documentos que estão vinculados aos seus perfis de usuário. Estes documentos, que incluem CPF para pessoas físicas e CNPJ para pessoas jurídicas, são essenciais para garantir a conformidade com as regulamentações e para a validação da identidade dos titulares. A tabela 'DOCUMENTS' armazena esses documentos, que são cruzados com as informações das contas para garantir que cada transação seja atribuída ao titular correto [Data: Reports (12, 4, 10, 6)].

Múltiplas Contas por Titular

Alguns titulares de contas podem possuir múltiplas contas associadas ao mesmo perfil. Isso é comum em casos onde o titular controla tanto contas pessoais quanto contas corporativas. O dataset permite rastrear todas essas contas de forma integrada, garantindo que todas as atividades financeiras do titular sejam monitoradas de maneira eficaz. Esta capacidade é crucial para a detecção de possíveis atividades fraudulentas ou para a análise de comportamentos financeiros complexos [Data: Reports (11, 5, 8)].

Interconexão entre Contas e Transações

A interconexão entre as contas e as transações financeiras é um dos aspectos mais importantes do dataset. Cada transação registrada está diretamente vinculada a uma ou mais contas, o que permite traçar um mapa completo das atividades financeiras de cada titular. Essa interconexão é especialmente útil na análise de comportamentos anômalos, onde transações suspeitas podem ser rapidamente associadas aos seus titulares para uma investigação mais aprofundada [Data: Reports (15, 3, 14, 7, 9)].

Titulares de Contas Corporativas

No caso de contas corporativas, os titulares são identificados através de CNPJ e são vinculados a perfis de usuários específicos que têm autorização para gerenciar essas contas. A identificação precisa dos titulares de contas corporativas

é essencial para a gestão de riscos e para a implementação de controles financeiros adequados. As tabelas 'CORPORATIONS' e 'ACCOUNTS' são fundamentais para mapear essas relações [Data: Reports (10, 2, 13)].

Em resumo, o dataset oferece uma visão detalhada dos titulares das contas envolvidas nas transações, permitindo uma identificação precisa e um monitoramento eficaz de todas as operações financeiras. A interconexão entre perfis, documentos e contas garante que todas as transações possam ser rastreadas de volta aos seus titulares, oferecendo um alto nível de segurança e conformidade.

Pergunta 5

Como podemos distinguir entre transações fraudulentas e não fraudulentas?

Resposta

Análise de Padrões de Comportamento

A distinção entre transações fraudulentas e não fraudulentas é um processo complexo que se baseia na análise detalhada de padrões de comportamento financeiro registrados no dataset. Cada transação é avaliada com base em uma série de critérios que incluem valor, frequência, localização, e o histórico de comportamento do titular da conta. Essa abordagem permite que a instituição identifique anomalias que possam indicar atividades fraudulentas, enquanto distingue essas transações das operações legítimas realizadas pelos usuários.

Monitoramento de Valor e Frequência das Transações

Uma das maneiras mais eficazes de distinguir entre transações fraudulentas e não fraudulentas é monitorar o valor e a frequência das transações. Transações fraudulentas tendem a envolver valores que são significativamente diferentes das operações regulares do titular, ou ocorrem com uma frequência atípica. Por exemplo, uma série de transações de alto valor realizadas em um curto período de tempo pode ser um indicativo de fraude, especialmente se o titular normalmente não realiza operações dessa natureza [Data: Reports (13, 3, 9, 5)].

Análise de Localização e Dispositivos

A localização geográfica e o dispositivo utilizado para realizar uma transação são fatores críticos na detecção de fraudes. Transações realizadas em locais inesperados, distantes da localização habitual do titular, ou a partir de dispositivos não reconhecidos, podem ser sinais de atividade fraudulenta. O dataset permite cruzar informações de localização e dispositivo com o histórico de transações do titular, ajudando a identificar operações suspeitas [Data: Reports (12, 4, 10, 6, +more)].

Análise Temporal e Comportamento do Titular

O tempo em que uma transação é realizada também é um fator importante. Fraudes frequentemente ocorrem em horários atípicos, como durante a madrugada, ou em dias de baixa atividade financeira. Além disso, uma mudança súbita no comportamento do titular — como um aumento repentino no número de transações ou uma mudança nos tipos de comerciantes com quem ele interage — pode ser um indicativo de que a conta foi comprometida. A análise temporal combinada com o comportamento histórico do titular é uma ferramenta poderosa para distinguir transações legítimas de fraudulentas [Data: Reports (11, 6, 8)].

Detecção de Padrões de Tentativas de Acesso

Fraudes geralmente são precedidas por tentativas repetidas de acesso não autorizado à conta do titular. O monitoramento dessas tentativas, especialmente quando ocorrem falhas sucessivas de login ou transações que não são completadas, pode ajudar a detectar fraudes antes que transações suspeitas sejam realizadas. Essa análise preventiva é fundamental para proteger os usuários e minimizar perdas financeiras [Data: Reports (15, 5, 14)].

Integração de Dados e Machine Learning

A distinção entre transações fraudulentas e não fraudulentas pode ser aprimorada com a integração de técnicas de machine learning. Modelos preditivos podem ser treinados para reconhecer padrões de fraude com base em dados históricos e em tempo real. Esses modelos são capazes de aprender e se

adaptar a novas formas de fraude, oferecendo uma camada adicional de segurança. A combinação de machine learning com as análises tradicionais permite uma detecção de fraudes mais precisa e eficiente [Data: Reports (10, 2, 13)].

Em resumo, a distinção entre transações fraudulentas e não fraudulentas requer uma abordagem multifacetada que combine análise de padrões de comportamento, monitoramento de valores e frequência, análise de localização e dispositivos, além do uso de machine learning. Essa abordagem integrada permite à instituição proteger seus clientes de maneira eficaz, garantindo a integridade das transações financeiras.

Pergunta 6

Como as transações são monitoradas e marcadas como potenciais fraudes?

Resposta

Monitoramento Contínuo de Transações

O monitoramento contínuo das transações financeiras é um processo essencial para garantir a segurança e a integridade das operações realizadas na plataforma. Cada transação é analisada em tempo real utilizando uma combinação de regras predefinidas e algoritmos de machine learning que avaliam o risco de fraude com base em múltiplos critérios. Este monitoramento é realizado em várias etapas, desde a validação inicial da transação até a análise posterior, permitindo que a instituição identifique e responda rapidamente a qualquer atividade suspeita [Data: Reports (14, 2, 11, 6, 13)].

Implementação de Regras Predefinidas

As transações são inicialmente monitoradas através de um conjunto de regras predefinidas que estabelecem limites e padrões de comportamento esperado. Estas regras incluem limites máximos de valor para transações, número máximo de transações permitidas em um período específico, e critérios de localização e dispositivo. Quando uma transação excede esses limites ou desvia

significativamente dos padrões esperados, ela é automaticamente marcada como suspeita e passa para uma análise mais aprofundada [Data: Reports (13, 3, 9, 5)].

Análise de Machine Learning para Detecção de Fraudes

Além das regras predefinidas, a instituição utiliza algoritmos de machine learning que são treinados para identificar padrões de fraude com base em dados históricos. Esses algoritmos podem detectar anomalias que não seriam capturadas pelas regras tradicionais, como mudanças sutis no comportamento do usuário ou padrões de fraude emergentes. Quando uma transação é considerada de alto risco pelo modelo, ela é marcada para revisão manual ou ações automáticas, como o bloqueio temporário da conta [Data: Reports (12, 4, 10, 6, +more)].

Integração de Dados de Múltiplas Fontes

O processo de monitoramento integra dados de várias fontes, incluindo informações de transações anteriores, dispositivos utilizados, localização geográfica, e comportamento do usuário. Essa integração permite uma visão holística da transação e do contexto em que ela ocorre, aumentando a precisão na detecção de fraudes. Por exemplo, uma transação que ocorre em um dispositivo não reconhecido e em um local geograficamente distante pode ser rapidamente marcada como suspeita, mesmo que o valor da transação seja relativamente baixo [Data: Reports (11, 6, 8)].

Processos de Revisão e Marcação Manual

Em casos onde a análise automatizada não é conclusiva, a transação é encaminhada para revisão manual por uma equipe especializada. Durante essa revisão, analistas verificam o contexto da transação, comparando-a com o histórico de atividades do titular da conta e outras variáveis relevantes. Se uma transação é confirmada como fraudulenta, ela é marcada no sistema, e medidas corretivas são tomadas, como o bloqueio da conta ou a reversão da transação [Data: Reports (15, 5, 14)].

Feedback e Aprendizado Contínuo

O sistema de monitoramento é dinâmico, ajustando-se continuamente com base no feedback das transações marcadas como fraudulentas. Quando uma fraude é confirmada, o modelo de machine learning é atualizado para reconhecer padrões similares no futuro, melhorando sua eficácia ao longo do tempo. Além disso, novas regras podem ser implementadas com base nas tendências emergentes, garantindo que o sistema se mantenha atualizado com as últimas ameaças [Data: Reports (10, 2, 13)].

Em resumo, o monitoramento e a marcação de transações como potenciais fraudes é um processo multifacetado que envolve a combinação de regras predefinidas, análise de machine learning, integração de dados de múltiplas fontes, e revisão manual. Este sistema robusto permite à instituição detectar e responder rapidamente a fraudes, protegendo tanto os clientes quanto a integridade financeira da plataforma.

Pergunta 7

Como as características das transações fraudulentas diferem das não fraudulentas?

Resposta

Diferenças Fundamentais entre Transações Fraudulentas e Não Fraudulentas

As transações fraudulentas e não fraudulentas podem parecer similares em termos de execução, mas há diferenças fundamentais nas características que as distinguem. Essas diferenças podem ser observadas em aspectos como o valor da transação, a frequência com que ocorrem, a localização geográfica, o dispositivo utilizado, e o comportamento geral do titular da conta. Essas características são analisadas em conjunto para identificar padrões que ajudam a distinguir fraudes de transações legítimas [Data: Reports (14, 2, 11, 6, 13)].

Valor e Frequência das Transações

Uma das diferenças mais notáveis entre transações fraudulentas e não fraudulentas está no valor e na frequência das transações. Transações fraudulentas tendem a envolver valores que são significativamente maiores ou menores do que as transações típicas realizadas pelo titular da conta. Além disso, fraudes frequentemente ocorrem em rajadas, com várias transações sendo realizadas em um curto período de tempo, o que contrasta com a distribuição mais uniforme de transações legítimas ao longo do tempo [Data: Reports (13, 3, 9, 5)].

Localização Geográfica e Dispositivos Utilizados

As transações fraudulentas muitas vezes ocorrem em locais geográficos que são inconsistentes com os padrões usuais do titular da conta. Por exemplo, uma transação realizada em um país estrangeiro onde o titular nunca esteve pode ser um forte indicativo de fraude. Além disso, fraudes frequentemente envolvem o uso de dispositivos que não foram previamente associados ao titular, ao contrário das transações legítimas, que tendem a ser realizadas a partir de dispositivos reconhecidos e confiáveis [Data: Reports (12, 4, 10, 6, +more)].

Comportamento Temporal

O tempo em que uma transação é realizada também pode diferir significativamente entre fraudes e transações legítimas. Transações fraudulentas geralmente ocorrem em horários atípicos, como durante a madrugada ou em períodos de baixa atividade financeira. Em contraste, transações legítimas seguem os padrões normais de comportamento do usuário, ocorrendo em horários que são consistentes com o histórico de atividades do titular da conta [Data: Reports (11, 6, 8)].

Relação com Comerciante e Tipo de Transação

As transações fraudulentas podem ocorrer com comerciantes que não têm histórico de interação com o titular da conta, ou em categorias de despesas que são incomuns para aquele perfil. Por exemplo, uma transação grande em uma loja de luxo pode ser suspeita se o titular geralmente faz compras modestas

em supermercados ou lojas de conveniência. Essa discrepância entre o tipo de comerciante e o perfil de despesas pode ser uma bandeira vermelha na detecção de fraudes [Data: Reports (15, 5, 14)].

Análise de Anomalias e Padrões Atípicos

Finalmente, as transações fraudulentas frequentemente exibem padrões que são atípicos quando comparados ao comportamento histórico do titular da conta. Isso pode incluir uma mudança súbita no tipo de transações realizadas, uma variação no método de pagamento, ou uma alteração no padrão de gastos. Analisar essas anomalias permite que a instituição identifique fraudes de maneira eficaz, contrastando-as com o comportamento previsível das transações legítimas [Data: Reports (10, 2, 13)].

Em resumo, as características que diferenciam as transações fraudulentas das não fraudulentas são multifacetadas, envolvendo diferenças em valor, frequência, localização, dispositivos, e comportamento temporal. A análise integrada dessas características permite uma distinção clara entre fraudes e transações legítimas, ajudando a proteger a integridade financeira dos usuários e da instituição.

Pergunta 8

Quão eficazes são os algoritmos de detecção de fraudes atuais e como podem ser utilizados nesse dataset?

Resposta

Eficácia dos Algoritmos de Detecção de Fraudes

Os algoritmos de detecção de fraudes atuais têm se mostrado altamente eficazes na identificação de atividades fraudulentas em sistemas financeiros complexos. Esses algoritmos utilizam técnicas avançadas de machine learning e análise estatística para identificar padrões anômalos e comportamentos atípicos que indicam a presença de fraudes. Com a capacidade de analisar grandes volumes de dados em tempo real, esses algoritmos conseguem detectar fraudes com uma precisão cada vez maior, reduzindo o número de falsos positivos e garantindo que transações legítimas não sejam indevidamente bloqueadas.

Aplicação de Algoritmos no Dataset

No contexto do dataset em questão, os algoritmos de detecção de fraudes podem ser aplicados de várias maneiras para maximizar a eficácia na identificação de atividades suspeitas. O dataset, que contém informações detalhadas sobre transações, perfis de usuários, dispositivos e localização, oferece um rico campo de dados para o treinamento e aplicação desses algoritmos. A seguir, detalhamos como os algoritmos podem ser aplicados nas diferentes camadas do dataset [Data: Reports (13, 3, 9, 5, +more)].

Detecção de Anomalias em Valores de Transações

Uma das principais áreas de aplicação dos algoritmos de detecção de fraudes é a análise de valores de transações. Algoritmos como redes neurais e florestas aleatórias podem ser treinados para identificar anomalias em valores, comparando cada transação com o histórico de operações do usuário e com padrões gerais observados na plataforma. Transações que divergem significativamente dos padrões esperados podem ser marcadas para revisão manual ou bloqueadas automaticamente, dependendo do nível de risco associado [Data: Reports (12, 4, 10, 6)].

Análise de Padrões Temporais e Comportamentais

Outra aplicação crucial dos algoritmos de detecção de fraudes é a análise de padrões temporais e comportamentais. Algoritmos de aprendizado supervisionado podem ser utilizados para identificar mudanças sutis no comportamento do usuário ao longo do tempo, como um aumento repentino na frequência de transações ou a realização de operações em horários atípicos. Esses padrões, quando combinados com outras variáveis como localização geográfica e tipo de dispositivo, podem fornecer uma indicação clara de fraude [Data: Reports (11, 6, 8)].

Monitoramento de Localização e Dispositivos

Algoritmos de machine learning também são altamente eficazes na detecção de fraudes com base na análise de localização e dispositivos utilizados. Ao treinar

modelos que consideram o histórico de localizações e dispositivos associados a cada perfil, é possível identificar rapidamente transações que ocorrem em locais incomuns ou a partir de dispositivos não reconhecidos. Esse monitoramento é particularmente útil para prevenir fraudes que envolvem o uso de dispositivos comprometidos ou acessos não autorizados [Data: Reports (15, 5, 14)].

Integração de Múltiplas Fontes de Dados

A eficácia dos algoritmos de detecção de fraudes pode ser amplificada através da integração de dados de múltiplas fontes, como informações de perfis, transações, dispositivos e localizações. Essa abordagem holística permite que os algoritmos considerem uma gama mais ampla de fatores ao avaliar o risco de fraude, aumentando a precisão das detecções e reduzindo a incidência de falsos positivos. Além disso, a integração de dados em tempo real possibilita a detecção e resposta imediata a atividades suspeitas [Data: Reports (10, 2, 13)].

Aprendizado Contínuo e Adaptação

Os algoritmos de detecção de fraudes não são estáticos; eles continuam a evoluir e se adaptar com o tempo. Utilizando técnicas de aprendizado contínuo, os modelos podem ser atualizados regularmente com novos dados e padrões de fraude emergentes. Isso garante que os algoritmos permaneçam eficazes mesmo à medida que os métodos de fraude se tornam mais sofisticados. No contexto deste dataset, o aprendizado contínuo pode ser aplicado para refinar constantemente os modelos, melhorando a detecção de fraudes e minimizando o impacto sobre os usuários legítimos.

Em resumo, os algoritmos de detecção de fraudes atuais são ferramentas poderosas que, quando aplicadas de maneira eficaz ao dataset, podem identificar e prevenir atividades fraudulentas com alta precisão. A integração de técnicas de machine learning com dados detalhados sobre transações, perfis e comportamentos é essencial para maximizar a eficácia desses algoritmos e proteger a integridade financeira da plataforma.

DISCUSSÃO

Os resultados obtidos neste estudo destacam a eficácia da abordagem GraphRAG na análise de transações financeiras e na identificação de padrões de comportamento dentro do setor financeiro. Através da implementação de GraphRAG, foi possível explorar e mapear de forma detalhada as interconexões entre diversas entidades financeiras, como contas, perfis de usuários, transações e dispositivos móveis. Essa análise revelou uma estrutura complexa, mas coerente, que sublinha a importância da inter-relação entre essas entidades para a segurança e integridade das operações financeiras.

No entanto, uma análise mais detalhada dos resultados revelou que a classe de questões relacionadas ao “Como”, particularmente em relação à distinção entre transações fraudulentas e não fraudulentas e na maneira como as transações são monitoradas e marcadas como potenciais fraudes, apresentou maior dificuldade de resposta. Isso reflete a complexidade inerente dessas questões, que demandam não apenas a recuperação de informações, mas também uma interpretação contextual mais profunda. Esta observação é corroborada pelo trabalho de [An *et al.*, 2024], que argumenta que questões do tipo “Como” são particularmente desafiadoras devido à necessidade de integrar múltiplas variáveis e contextos em sua resolução.

Além disso, foi observado que as perguntas relacionadas ao “Quem”, como a identificação dos titulares das contas e quem relatou as fraudes, foram respondidas com maior facilidade. Isso sugere que questões que exigem apenas a recuperação direta de informações, sem a necessidade de correlações complexas, são mais facilmente processadas e respondidas pelo modelo. Este padrão evidencia a robustez do GraphRAG em lidar com interconexões diretas, ao mesmo tempo em que destaca as áreas onde a análise pode ser mais desafiadora.

Comparando esses resultados com a literatura existente, percebe-se que o GraphRAG oferece uma vantagem distinta em relação a metodologias tradicionais de Geração Aumentada por Recuperação (RAG). Enquanto abordagens anteriores como a de [Brown *et al.*, 2020] focavam principalmente na recuperação de informações relevantes, o GraphRAG vai além ao incorporar técnicas de grafos que permitem uma análise mais profunda e estruturada das interconexões entre as entidades. Isso não apenas melhora a precisão das respostas geradas, mas

também aumenta a capacidade de identificar padrões complexos, algo essencial em ambientes de dados volumosos como o setor financeiro.

No entanto, este estudo também enfrenta algumas limitações que merecem ser destacadas. A configuração utilizada, embora robusta e eficaz, exige um ambiente de computação de alto desempenho, o que pode limitar sua aplicação em contextos com recursos computacionais mais modestos. Além disso, a qualidade dos dados inseridos diretamente influencia os resultados obtidos; dados incompletos ou de baixa qualidade podem comprometer a eficácia do GraphRAG.

As implicações práticas dos resultados são consideráveis. O uso do Graph-RAG pode ser particularmente benéfico para instituições financeiras que buscam melhorar suas práticas de análise de dados, auditoria e conformidade regulatória. A capacidade de mapear e analisar grandes volumes de transações com precisão e rapidez pode ajudar a identificar fraudes de maneira mais eficaz e a implementar medidas preventivas com maior eficiência.

Para futuras pesquisas, seria interessante explorar a adaptação do GraphRAG para outros setores onde a análise de grandes volumes de dados textuais é crucial. Além disso, há espaço para otimizar ainda mais os parâmetros do GraphRAG para melhorar seu desempenho em diferentes cenários, especialmente em ambientes com restrições computacionais.

REFERÊNCIAS

- [Alghisi *et al.* 2024] Alghisi, S., Rizzoli, M., Roccabruna, G., Mousavi, S. M., and Riccardi, G. (2024). Should we fine-tune or rag? evaluating different techniques to adapt llms for dialogue. ArXiv, abs/2406.06399.
- [An *et al.* 2024] An, K., Yang, F., Li, L., Lu, J., Cheng, S., Wang, L., Zhao, P., Cao, L., Lin, Q., Rajmohan, S., *et al.* (2024). Thread: A logic-based data organization paradigm for how-to question answering with retrieval augmented generation. arXiv preprint arXiv:2406.13372.
- [Brown *et al.* 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- [Bughin *et al.* 2017] Bughin, J., Hazan, E., Sree Ramaswamy, P., DC, W., Chu, M., *et al.* (2017). Artificial intelligence the next digital frontier.

[Buyya 1999] Buyya, R. (1999). High Performance Cluster Computing: Architectures and Systems. Prentice Hall PTR, USA.

[Devlin *et al.* 2019] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

[Edge *et al.* 2024] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., and Larson, J. (2024). From local to global: A graph rag approach to query-focused summarization.

[Guu *et al.* 2020] Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). Realm: retrieval-augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning, ICML'20. JMLR.org.

[Jurafsky and Martin 2024] Jurafsky, D. and Martin, J. H. (2024). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. Prentice Hall, 3rd edition. Online manuscript released August 20, 2024.

[Kipf and Welling 2017] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR '17.

[Lewis *et al.* 2020] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela,

D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, Advances in Neural Information Processing Systems, volume 33, pages 9459–9474. Curran Associates, Inc.

[Li *et al.* 2024] Li, B., Yang, P., Sun, Y., Hu, Z., and Yi, M. (2024). Advances and challenges in artificial intelligence text generation. Frontiers of Information Technology Electronic Engineering, 25:64–83.

[Merkel 2014] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. Linux J., 2014(239).

[Negus *et al.* 2008] Negus, C., Negus, C., and Caen, F. (2008). Ubuntu Linux Toolbox: 1000+ Commands for Ubuntu and Debian Power Users. Wiley.

[Nickolls *et al.* 2008] Nickolls, J., Buck, I., Garland, M., and Skadron, K. (2008). Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? Queue, 6(2):40–53.

[Samek *et al.* 2017] Samek, W., Wiegand, T., and Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

[Shoeybi *et al.* 2019] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. arXiv preprint arXiv:1909.08053.

[Touvron *et al.* 2023] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

[Wang *et al.* 2024] Wang, Y., Hernandez, A. G., Kyslyj, R., and Kersting, N. (2024). Evaluating quality of answers for retrieval-augmented generation: A strong llm is all you need. arXiv preprint arXiv:2406.18064.

[Wiseman *et al.* 2017] Wiseman, S., Shieber, S., and Rush, A. (2017). Challenges in data-to-document generation. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

PREDICTIVE ANALYSIS OF LOAN DEFAULT USING MACHINE LEARNING MODELS

Alejandro Aviles

Specialization Course in Data Science - Aeronautics
Institute of Technology (ITA)
alejandro1407@gmail.com

Fernando Luiz Scherer

Specialization Course in Data Science - Aeronautics
Institute of Technology (ITA)
fernando.luizscherer@gmail.com

João Victor Neves Loose

Specialization Course in Data Science - Aeronautics
Institute of Technology (ITA)
jvloose@hotmail.com

Johnny Cardoso Marques

Division of Computer Science – Aeronautics Institute
of Technology (ITA)

ABSTRACT

This paper investigates the application of machine learning models to predict loan defaults. Three models were selected: Logistic Regression, Random Forest, and Gradient Boosting, to evaluate their effectiveness in handling imbalanced datasets. The study focuses on the preprocessing steps such as data resampling and feature scaling, followed by model training and cross-validation. The performance of each model was assessed using key metrics including accuracy, precision, recall, F1 score, and ROC AUC. The results highlight the trade-offs between model complexity, computational efficiency, and predictive accuracy, providing valuable insights for financial institutions considering the deployment of these models for risk management.

RESUMO

Este trabalho investiga a aplicação de modelos de aprendizado de máquina para prever inadimplências em empréstimos. Foram selecionados três modelos: Regressão Logística, Random Forest e Gradient Boosting, para avaliar sua eficácia no tratamento de conjuntos de dados desbalanceados. O estudo foca nas etapas de pré-processamento, como reamostragem de dados e escalonamento de características, seguidas de treinamento e validação cruzada dos modelos. O desempenho de cada modelo foi avaliado utilizando métricas-chave, incluindo acurácia, precisão, recall, F1 score e AUC-ROC. Os resultados destacam as compensações entre complexidade do modelo, eficiência computacional e acurácia preditiva, fornecendo insights valiosos para instituições financeiras que consideram o uso desses modelos para gestão de risco.

INTRODUCTION

Contextualization

Predicting loan default is a critical challenge in the financial sector, as it directly impacts the stability and profitability of financial institutions. Loan defaults pose significant risks, prompting banks to develop robust credit scoring systems that require constant updates with new variables and advanced statistical techniques to enhance predictive accuracy. Traditional methods such as credit scoring analysis and linear statistical models have historically been central to credit risk assessment. However, these approaches often fall short in capturing the complex and nonlinear relationships inherent in financial data. As Simão notes, traditional credit scoring methodology evolved to the use of statistical techniques such as linear discriminant analysis and linear and logistic regression, yet these methods face limitations in modern financial environments [Simão 2023].

This study applies machine learning algorithms—Logistic Regression, Random Forest, and Gradient Boosting—to predict loan defaults using a comprehensive dataset from Lending Club. Logistic Regression is widely used for its interpretability and ease of use, but ensemble methods like Random Forest and Gradient Boosting offer advantages in handling large datasets and capturing complex patterns. [Simão 2023] describes Random Forest models as “state-of-the-art” for their ability to manage both classification and regression problems effectively. Chopra and Bhilare highlight that the adoption of newer ensemble techniques, such as Gradient Boosting, is crucial for improving the precision and recall of credit scoring models, ultimately contributing to better decisionmaking processes in financial institutions [Chopra and Bhilare 2018].

Data preprocessing is a crucial step in this study, involving techniques like data normalization and class balancing to address the inherent imbalances in the dataset. These steps are essential to ensure that the models not only perform accurately but also generalize well to new data. [Simão 2023] emphasizes that data preprocessing is a critical component of this study, involving techniques like data normalization and class balancing to address the inherent imbalances in the dataset.

The evaluation of the models includes metrics such as AUC-ROC, accuracy, recall, and F1-score, providing a comprehensive assessment of each model's performance. Ensemble methods have shown promising results across these metrics, particularly in financial contexts where precise risk assessment is crucial. As highlighted by [Simão 2023], Area under the ROC Curve and accuracy are considered standard evaluation metrics for the evaluation of credit scoring models, which are the most commonly applied in the literature. The robustness of ensemble methods in these metrics underscores their suitability for financial applications, where the balance between true positive rates and false positive rates is crucial.

This study underscores the importance of advanced machine learning techniques in developing robust credit risk models, especially in the context of P2P lending platforms, where decentralized decision-making heightens the need for accurate risk assessment tools. By leveraging sophisticated algorithms and thorough data preprocessing, financial institutions can enhance their ability to predict defaults, thereby improving their risk management strategies and overall financial stability. As noted by [Simão 2023], the most recent financial crisis serves as a useful reminder of the importance of a robust risk management culture in maintaining a healthy economy.

Research Problem

The primary research problem addressed in this study is: How can machine learning models be applied with better performance in predicting loan defaults?

General Objective

The general objective of this study is to assess the viability of machine learning models in predicting loan defaults using data from LendingClub, with a particular focus on evaluating their performance in a real-world financial context where accuracy and reliability are crucial.

Specific Objectives

- **Evaluate Model Viability for Financial Applications:** Assess the effectiveness of various machine learning models, including Logistic Regression, Random Forest, and Gradient Boosting, in predicting loan defaults, with a focus on their practical applicability in the financial sector.
- **Analyze Model Performance Metrics:** Use metrics such as accuracy, precision, recall, F1 Score, and ROC AUC to critically evaluate the strengths and weaknesses of each model, particularly in the context of minimizing false positives and negatives.
- **Investigate the Impact of Data Preprocessing:** Explore the role of data preprocessing techniques, such as feature selection and class balancing, in enhancing model performance and reducing the risks of information leakage and overfitting.
- **Propose Improvements for Future Research:** Based on the findings, suggest potential improvements and future research directions, including the integration of additional data sources and the exploration of more advanced modeling techniques to better meet the demands of financial risk management.

By achieving these objectives, this study aims to contribute to the field of financial risk management by providing insights into the application of machine learning models for predicting loan defaults and highlighting best practices for developing robust predictive models.

LITERATURE REVIEW

As the volume and complexity of financial data have grown, traditional statistical methods have increasingly been supplemented—or even replaced—by more sophisticated machine learning algorithms. These advancements enable more nuanced risk assessments, leveraging patterns and relationships within large datasets that were previously difficult to detect.

To explore these advancements, this section delves into the critical components of credit risk modeling, examining the role of ensemble methods in enhancing predictive accuracy and addressing the inherent challenges in data preprocessing. By dissecting these elements, we can better understand how

contemporary approaches to credit scoring are evolving and identify the key areas where machine learning continues to push the boundaries of financial risk management.

Credit Risk Management Fundamentals

Credit risk management is a critical function within financial institutions, aiming to mitigate the economic losses that result from a counterparty's failure to fulfill its contractual obligations. As highlighted by [Simão 2023], credit scoring has become one of the primary tools for assessing credit risk, with its ultimate goal being to discriminate between “good” and “bad” debts based on the likelihood of default. The evolution of credit risk management has been significantly influenced by technological advancements, leading to the integration of machine learning techniques into traditional credit scoring methods.

Ensemble Methods in Predictive Modeling

Ensemble methods have gained prominence in the field of machine learning due to their ability to enhance predictive performance by combining multiple models. These methods can be categorized into two main types: homogeneous (e.g., Bagging, Boosting) and heterogeneous (e.g., Stacking). According to [Yosza Dasril *et al.* 2023], ensemble methods, particularly those based on decision trees like Random Forest and Gradient Boosting, have demonstrated superior performance in credit risk evaluation compared to single models. The study emphasizes the importance of optimizing parameters to improve classification accuracy, a sentiment echoed by [Malekipirbazari and Aksakalli 2015], who also advocate for the use of ensemble models in credit scoring to leverage the strengths of different algorithms.

Challenges in Data Preprocessing

Data preprocessing is a crucial step in developing robust machine learning models, particularly in the financial sector where datasets are often plagued with issues such as missing values, outliers, and imbalanced classes. As described by

[Simão 2023], the quality of data preprocessing directly impacts the performance of the predictive models. Techniques such as feature selection, normalization, and class balancing are essential to ensure that the models are trained on clean and representative data. Moreover, [Błaszczynski *et al.* 2021] highlight the significance of addressing class imbalance through methods like Random Under-Sampling (RUS), which helps prevent model bias towards the majority class and improves the accuracy of predictions for the minority class, typically the defaulters.

The reviewed literature underscores the critical role of ensemble methods and rigorous data preprocessing in developing accurate and reliable predictive models for credit risk management. These studies provide valuable insights into the challenges and best practices for applying machine learning techniques in this domain, highlighting areas where further research and development are needed to enhance the robustness of credit scoring systems.

METHODOLOGY AND DATA ANALYSIS

This section delineates the methodology employed in the study, encompassing data acquisition, preprocessing, model development, and evaluation. The primary objective is to develop predictive models that accurately assess the likelihood of loan default, leveraging various machine learning techniques. The process involves rigorous data preprocessing to ensure data integrity, followed by the application of advanced machine learning algorithms. This section provides a detailed account of each step taken in the analysis, offering insights into the rationale and approach behind each methodological choice.

Data Processing

Data processing is a pivotal component of this study, ensuring that the data is prepared and optimized for machine learning algorithms. Given the complexity and inherent challenges associated with financial datasets, this subsection outlines the comprehensive steps taken to clean, transform, and prepare the data. The entire process, from data acquisition to model development, is summarized in Figure 8.

Data Acquisition

The Lending Club dataset, utilized in this study, is one of the most extensive and detailed publicly available financial datasets, providing comprehensive information on peer-to-peer (P2P) loans issued from 2007 to 2020. The dataset is vast, encompassing:

- **Number of rows:** 2,925,493
- **Number of columns:** 142

This dataset offers a rich variety of variables that capture borrower demographics, financial profiles, loan terms, and the performance of the loans over time. Lending Club operates as a P2P lending platform, where individual investors fund loans to borrowers, bypassing traditional financial institutions. This P2P model makes the dataset particularly valuable for studying credit risk and predicting defaults, as it contains extensive attributes, including variables related to the purpose of the loan, interest rates, borrower income, debt-to-income ratios, and credit scores, among others. Additionally, the dataset meticulously tracks the status of each loan, whether fully paid, charged off, or currently active, providing a comprehensive resource for analyzing the factors contributing to loan defaults and developing predictive models to assess potential default risks.

Given its comprehensiveness, the dataset is hosted on platforms like Kaggle and has been widely used in data science projects, providing critical insights for both academic research and practical applications in the finance industry.

Variable Selection

To ensure the model's performance would be realistic and generalizable, we carefully selected the variables to include:

- **Exclusion of Post-Loan Variables:** Variables such as `issue_d`, `last_pymnt_d`, and `total_pymnt` were excluded because they contain information only available after the loan was issued. Including these variables would introduce future data into the model, leading to unrealistic predictions and potential overfitting.

- **Exclusion of Financial Hardship and Settlement Variables:** Variables indicating financial hardship or debt settlements, such as `hardship_flag` and `debt_settlement_flag`, were also excluded. These variables reflect postloan events and could bias the model if included.
- **Exclusion of Identifiers and Subjective Variables:** Variables like `member_id`, `desc`, and `emp_title` were removed due to their lack of predictive value or potential to introduce noise and subjectivity into the model.
- **Exclusion of Internal Lending Club Classifications:** Variables like `grade`, `sub_grade`, and `fico_range_high` were excluded because they represent internal classifications or redundant data that could introduce bias and might not be applicable to future loans.

After these exclusions, the dataset was significantly refined, resulting in a reduction to 67 columns. The dataset was significantly refined, reducing the potential for over-fitting and ensuring that the model would be trained on data that is relevant and available at the time of loan issuance.

Selection of the Target Variable

Focusing on Final Outcomes: The target variable `loan_status` originally contained multiple categories representing different stages and outcomes of loans. For this study, we focused on two definitive outcomes: `Fully Paid` and `Charged Off`. These categories were chosen because they represent clear and final states of a loan, making them ideal for predicting default risk.

Binary Conversion: To simplify the model and make the predictions more interpretable, we converted `loan_status` into a binary variable: 0 for `Fully Paid` and 1 for `Charged Off`. This conversion allows the model to focus on distinguishing between successful and defaulted loans, which is critical for assessing credit risk.

As a result of these actions, the dataset was further refined, now consisting of 1,860,331 rows, with the target variable `loan_status` containing 1,497,783 instances of `Fully Paid` (0) and 362,548 instances of `Charged Off` (1).

Handling Missing Data

Missing data can significantly impair model performance if not addressed appropriately. In this study, we adopted a dual approach to handle missing values. Initially, variables with more than 7% missing data were excluded from the analysis. This threshold was selected to minimize the risk of introducing bias or noise through imputation, particularly because high levels of missingness (up to 96.8%) were observed in some columns related to secondary applicant information and other variables that were less critical to the primary analysis. For example, variables such as `sec_app_collections_12_mths_ex_med` and `sec_app_chargeoff_within_12_mths` were excluded because they had over 96% missing values, which would significantly compromise the robustness of any imputation method.

Furthermore, columns with missing values below the 7% threshold were retained, but rows containing any missing data in these columns were removed. This step ensured that the analysis was based on complete and reliable records, reducing the likelihood of biased results due to incomplete data. For instance, variables like `emp_length` had about 6.3% missing data, which was considered manageable. **The impact of this cleaning process on the number of rows and their distribution across the target variable is summarized in Table 1.**

The total number of rows decreased from 1,860,331 to 1,612,809, representing a 13.31% reduction. This was reflected in both the `loan_status` categories: the number of `Fully Paid (0)` loans decreased by 12.93%, and the number of `Charged Off (1)` loans decreased by 14.85%. The reduction in rows ensured that the dataset's integrity was maintained while focusing on complete cases to improve model robustness and reliability.

Additionally, a detailed examination of columns with lower percentages of missing data revealed critical insights into why they were included or excluded. For instance, columns such as `tot_coll_amt` and `tot_cur_bal`, with approximately 3.63% missing data, were retained due to their importance in capturing financial behaviors, despite the missing values. In contrast, variables like `verification_status_joint` and `dti_joint`, which had more than 96% missing data, were excluded to prevent potential biases.

Table 1 - Comparison of Rows and Percentage Variation after Data Cleaning.

Comparison	Rows in df	Rows in df_cleaned	Percentage Variation (%)
Total Rows	1,860,331	1,612,809	-13.31%
loan_status: 0	1,497,783	1,304,089	-12.93%
loan_status: 1	362,548	308,720	-14.85%

The approach of excluding high-missingness variables and eliminating incomplete rows for the remaining data helped maintain the dataset's analytical integrity, ensuring that the models built upon it were robust and dependable.

Treatment of Outliers and Data Normalization

Outliers in the dataset were identified and treated to enhance the quality and performance of the predictive models. The process involved both winsorization and logarithmic transformation, depending on the nature of the variable. This section details the specific treatments applied to various variables, along with their justifications.

As illustrated in Figures 1 to 5, the distributions of key variables before and after treatment demonstrate how these adjustments normalized the data, reducing the influence of extreme values.

For the **loan_amnt** (Loan Amount) variable, winsorization was applied to limit the impact of excessively large or small loans, as shown in Figure 1. The **annual_inc** (Annual Income) variable also underwent winsorization, smoothing out extreme income values, as demonstrated in the same figure.

The **delinq_2yrs** (Delinquencies in the Last 2 Years) variable was winsorized to adjust the distribution, which was heavily skewed due to some borrowers having multiple delinquencies. This can be observed in Figure 2. Similarly, the **inq_last_6mths** (Inquiries in the Last 6 Months) variable had its distribution adjusted to control for outliers, as shown in Figure 3.

The **pub_rec** (Public Records) variable underwent a similar winsorization process to moderate the impact of extreme values, as illustrated in Figure 3. Likewise, the **pub_rec_bankruptcies** (Public Record Bankruptcies) was controlled to limit the influence of multiple bankruptcies, as shown in Figure 5.

In the case of the **mort_acc** (Mortgage Accounts) variable, a less aggressive winsorization was applied to preserve natural variation while still controlling extremes, as can be seen in Figure 4. The **acc_open_past_24mths** (Accounts Opened in the Last 24 Months) variable was also winsorized to control the influence of a large number of recently opened accounts, as illustrated in Figure 4.

The **bc_util** (Revolving Credit Utilization) variable was treated to reduce the impact of high utilization rates, as shown in Figure 4. For the **mo_sin_old_il_acct** (Months Since Oldest Installment Account) and **mo_sin_old_rev_tl_op** (Months Since Oldest Revolving Account), winsorization was used to smooth the influence of very old accounts, as illustrated in Figure 4.

Additionally, the **tot_coll_amt** (Total Collection Amount) variable was logarithmically transformed and winsorized to reduce the impact of extremely high amounts, as illustrated in Figure 3. The **tot_cur_bal** (Total Current Balance) and **total_bal_ex_mort** (Total Balance Excluding Mortgages) variables were also treated with logarithmic transformations and winsorization, as shown in Figures 4 and 5, respectively.

Finally, the **dti** (Debt-to-Income Ratio) variable was subjected to a cap transformation, setting a maximum limit of 40 to prevent the analysis from being skewed by excessively high outlier values.

Feature Selection and Correlation Analysis

Effective feature selection is crucial in reducing model complexity and preventing over-fitting. To address this, a correlation matrix was employed to identify and remove highly correlated features, thereby mitigating the risks associated with multicollinearity. Features that could lead to information leakage, such as those capturing post-loan issuance information, were meticulously excluded. Additionally, feature engineering was conducted to create new variables that provide deeper insights into borrower behavior, such as credit utilization ratios and credit history metrics. These engineered features were designed to capture nuanced financial behaviors of borrowers, enhancing the model's predictive power.

In the feature selection process, correlation analysis played a pivotal role. **Figure 6** shows the correlation matrix before the exclusion of less relevant features. This analysis helped in identifying redundant columns that could cause

multicollinearity. After removing the highly correlated variables, a new correlation matrix was generated to ensure that multicollinearity was minimized. **Figure 7** presents the correlation matrix of the remaining numerical columns, illustrating the effectiveness of the feature selection process.

The remaining variables after this selection process, along with their descriptions, are summarized in **Table 2**. This table lists the key features retained for modeling, high-lighting their relevance and the specific aspects of borrower behavior they capture.

Before this step, the dataset contained:

- **Total number of rows:** 1,612,809
- **Total number of columns:** 42

After performing the correlation analysis and excluding variables with high correlation, the dataset was further refined, resulting in:

- **Total number of rows:** 1,612,809
- **Total number of columns:** 26

Table 2 - Description of Remaining Variables after Correlation Analysis.

Variable	Description
loan_amnt	The listed amount of the loan applied for by the borrower.
term	The number of payments on the loan. Values are in months.
int_rate	Interest rate on the loan.
emp_length	Employment length in years. Possible values range from <1 year to 10+ years.
home_ownership	The home ownership status provided by the borrower during loan application.
annual_inc	The self-reported annual income provided by the borrower during registration.
loan_status	The current status of the loan (e.g., Fully Paid, Charged Off).
dti	A ratio calculated using the borrower's total monthly debt payments divided by gross monthly income.
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.
inq_last_6mths	The number of inquiries in the last 6 months (excluding those made by LendingClub).
pub_rec	The number of derogatory public records.
tot_coll_amt	Total collection amounts ever owed.

Variable	Description
tot_cur_bal	Total current balance of all accounts.
acc_open_past_24mths	Number of trades opened in the last 24 months.
bc_util	Ratio of the borrower's revolving balance to their revolving credit limit.
mo_sin_old_il_acct	Months since the borrower's oldest installment account was opened.
mo_sin_old_rev_tl_op	Months since the borrower's oldest revolving account was opened.
mo_sin_rcnt_rev_tl_op	Months since the borrower's most recent revolving account was opened.
mort_acc	Number of mortgage accounts.
num_actv_bc_tl	Number of currently active bankcard accounts.
num_il_tl	Number of installment accounts.
num_op_rev_tl	Number of open revolving accounts.
pct_tl_nvr_dlq	Percentage of accounts never delinquent.
pub_rec_bankruptcies	Number of public record bankruptcies.
total_bal_ex_mort	Total credit balance excluding mortgage accounts.
total_il_high_credit_limit	The highest credit limit of all installment accounts.

Categorical Variable Encoding

The dataset included several categorical variables that required conversion into numerical formats to facilitate their use in machine learning models. We applied one-hot encoding to nominal categorical variables such as `term` and `home_ownership`. This method was chosen to prevent the introduction of ordinal relationships where none exist, ensuring that these variables are accurately represented as distinct, non-ordered categories.

After applying one-hot encoding, the number of columns increased from 26 to 28. This increase is due to the creation of additional columns for each category within the encoded variables. For example:

- **term:** The original `term` variable, which had two categories (36 months and 60 months), was split into two binary columns, `term_36_months` and `term_60_months`, representing whether a loan falls into either category.
- **home_ownership:** The `home_ownership` variable, which initially included multiple categories such as 'Rent', 'Mortgage', and 'Own', was expanded into several binary columns, each indicating the presence of a specific home ownership status.

This approach ensures that the categorical variables are represented in a way that is both interpretable and suitable for machine learning algorithms, allowing the models to process these features without misinterpreting their nature. The resulting dataset, now with 28 columns, reflects a refined structure that balances the need for detailed categorical information with the model's ability to handle it effectively.

Normalization and Scaling

To ensure that all features contribute equally to the model and improve its convergence, normalization and scaling were applied to the continuous variables. We utilized the `StandardScaler`, which standardizes features by removing the mean and scaling to unit variance. This process was particularly crucial for algorithms like Logistic Regression and Gradient Boosting, which are sensitive to the scale of input data. By normalizing the data, we not only facilitated faster convergence during training but also improved the stability and performance of the models. This step is essential to ensure that features with larger numerical ranges do not disproportionately influence the model's predictions.

Addressing Class Imbalance

Class imbalance is a common issue in credit risk datasets, where the number of default cases is often significantly lower than non-default cases. To address this, we applied Random Under-Sampling (RUS) to balance the dataset by reducing the size of the majority class. This technique was chosen over oversampling methods to maintain the integrity of the minority class characteristics, thereby avoiding the introduction of synthetic data that could potentially skew the results. The goal was to create a balanced dataset that allows the models to learn effectively from both classes. Balancing the class distribution is crucial in preventing the model from being biased towards the majority class, thereby improving its ability to accurately predict defaults in a real-world setting.

Model Development

Following data preprocessing, several machine learning models were developed to predict loan defaults. We focused on three primary models: Logistic Regression, Random Forest, and Gradient Boosting. Each model was carefully trained on the preprocessed dataset and evaluated using cross-validation to ensure robustness. We selected key performance metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, to assess each model's effectiveness in predicting defaults. The use of cross-validation provided a reliable means of estimating the models' performance on unseen data, thereby ensuring their generalizability.

Logistic Regression

Logistic Regression was employed as the baseline model due to its simplicity and interpretability, making it an ideal starting point for credit risk prediction. Given the large size of the dataset, we opted for the `saga` solver, which is known for its efficiency in handling large datasets and its ability to perform well with elastic net regularization. This combination allowed us to balance between L1 and L2 penalties, which help in feature selection and reducing overfitting. Cross-validation was performed to evaluate the model's performance, and hyperparameters were fine-tuned to optimize the results. Despite its simplicity, Logistic Regression provides a solid benchmark against which more complex models can be compared.

Random Forest

Random Forest, an ensemble learning method, was utilized to enhance prediction accuracy by aggregating the predictions of multiple decision trees. This model is particularly effective in handling large datasets with complex interactions among variables. By averaging the results of numerous trees, Random Forest reduces the risk of overfitting and increases robustness against noise in the data. This method's ability to handle non-linear relationships and capture interactions between variables made it a valuable addition to the model suite. Additionally, feature importance scores provided by Random Forest were insightful for understanding the contribution of each feature to the model's predictions.

Gradient Boosting

Gradient Boosting, another powerful ensemble method, was employed to further refine predictive performance. This model builds trees sequentially, with each new tree attempting to correct the errors made by the previous ones. The iterative nature of Gradient Boosting makes it particularly adept at handling complex patterns in the data, albeit with a higher risk of overfitting if not properly tuned. To mitigate this, we carefully adjusted the learning rate and number of estimators. Gradient Boosting is well-suited for imbalanced datasets, as it focuses on improving predictions for the harder-to-predict instances, which in this case, are the loan defaults. The model's high accuracy and ability to handle class imbalance made it an excellent choice for this credit risk application.

Model Evaluation

The performance of each model was rigorously evaluated using a comprehensive set of metrics to ensure a balanced assessment. The primary evaluation metric was the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which provides a nuanced measure of the model's ability to distinguish between classes. Additionally, accuracy, precision, recall, and F1-score were calculated to provide further insights into each model's performance. These metrics collectively offer a detailed understanding of the trade-offs between false positives and false negatives. Confusion matrices were also analyzed to gain a deeper understanding of how well each model classified true positives, false positives, true negatives, and false negatives. This analysis was crucial in identifying the best-performing model, which was then selected for further analysis and interpretation.

The methodology employed in this study reflects a rigorous approach to data processing, model development, and evaluation. Each step was meticulously designed to address the unique challenges posed by financial datasets, ensuring that the final models are both accurate and generalizable. The combination of ensemble methods, careful data preprocessing, and feature selection has led to the development of robust predictive models capable of assessing loan default risk with high accuracy. This comprehensive approach ensures that the models

developed are not only theoretically sound but also practically applicable in real-world financial contexts.

MODEL DEVELOPMENT AND VALIDATION

This section details the development and validation of the machine learning models used to predict loan defaults. The process involved selecting appropriate models, preprocessing the data, training the models, and evaluating their performance using a set of key metrics.

Model Selection

Three models were chosen for this study:

- **Logistic Regression:** A linear model commonly used for binary classification tasks. We utilized the 'saga' solver, which is well-suited for large datasets and supports regularization to mitigate overfitting.
- **Random Forest Classifier:** An ensemble learning method that builds multiple decision trees and aggregates their results. This model is effective in handling non-linear relationships and capturing complex patterns in the data.
- **Gradient Boosting Classifier:** Another ensemble method that sequentially builds models, each correcting the errors of its predecessor. Gradient Boosting is known for its high accuracy, particularly on datasets with imbalanced classes.

These models were selected to provide a comparison between linear and nonlinear approaches and to explore the benefits of ensemble methods in predictive performance.

Data Splitting and Resampling

The dataset was split into training and test sets with a 70/30 ratio, using stratification to ensure consistent class distribution across both sets. To address the class imbalance problem, where non-default instances significantly outnumber

defaults, Random Under-Sampling (RUS) was applied to the training data. This technique reduces the number of majority class instances to balance the dataset, which is crucial for avoiding model bias toward the majority class.

The impact of data treatment on the distribution of variables before and after applying these techniques can be observed in the figures located in the Appendix, specifically Figures 1 to 5. These figures provide visual insight into how the data was normalized and balanced.

Feature Scaling

Feature scaling was performed using StandardScaler, ensuring that all features contributed equally to the models' performance. This step is particularly important for models like Logistic Regression and Gradient Boosting, where the scale of input features can significantly impact the results. Scaling was applied to both the training and test datasets after resampling.

Model Training and Cross-Validation

Each model was trained on the resampled and scaled training data. Cross-validation was employed to assess the models' robustness. A 5-fold cross-validation approach was used, where the training data was split into five subsets. Each model was trained on four sub-sets and validated on the fifth, with this process repeated five times. The average cross-validation scores were used to evaluate each model's generalization capability.

Model Evaluation Metrics

The performance of the models was evaluated using several metrics:

- **Accuracy:** The proportion of correct predictions out of all predictions made.
- **Precision:** The proportion of true positives among all positive predictions, indicating the model's accuracy in predicting defaults.

- **Recall (True Positive Rate):** The proportion of actual defaults correctly identified by the model.
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
- **ROC AUC:** The area under the Receiver Operating Characteristic curve, which measures the model's ability to distinguish between the default and non-default classes across different thresholds.
- **Confusion Matrix:** A summary of true positives, false positives, true negatives, and false negatives, offering insights into the types of errors made by the model.

Additionally, the time taken for training and testing was recorded for each model, allowing for a comparison of computational efficiency.

ANALYSIS AND DISCUSSION OF RESULTS

Introduction

The analysis and discussion of results focus on evaluating the performance of the predictive models developed in this study. The models—Logistic Regression, Random Forest, and Gradient Boosting—were assessed based on their ability to accurately predict loan defaults using various metrics such as accuracy, precision, recall, F1 score, and ROC AUC. The implications of these results are further explored in the context of their potential application in financial risk management and decision-making processes.

Model Performance Evaluation

The performance of the three models was systematically evaluated based on their predictive capabilities, trained on a balanced dataset using Random Under-Sampling (RUS), and validated through cross-validation and testing on a separate dataset. Table 3 provides a summary of the key performance metrics for each model.

Table 3 - Model Performance Metrics.

Metric	Logistic Regression	Random Forest	Gradient Boosting
ROC AUC (Cross-Validation)	0.7121	0.7082	0.7162
Training Time (s)	4.98	146.71	148.03
Test Accuracy	0.6581	0.6473	0.6470
Precision	0.3108	0.3050	0.3075
Recall	0.6455	0.6588	0.6742
F1 Score	0.4195	0.4170	0.4223
ROC AUC (Test)	0.6533	0.6517	0.6574
False Positive Rate	0.3389	0.3554	0.3595
True Positives	59,779	61,019	62,440
False Positives	132,577	139,038	140,627
True Negatives	258,650	252,189	250,600
False Negatives	32,837	31,597	30,176
Test Time (s)	17.67	599.63	638.04

Logistic Regression

The Logistic Regression model demonstrated solid performance, particularly in its simplicity and interpretability. Achieving a ROC AUC of 0.7121 during cross-validation (as shown in Table 3), the model provided a reasonably good ability to distinguish between positive and negative classes.

A notable trade-off is observed between precision (0.3108) and recall (0.6455), which highlights a significant area for improvement, particularly in reducing false positives. This trade-off suggests that while the model is relatively good at identifying true positives (defaults), it also incorrectly identifies a substantial number of false positives, which could lead to unnecessary loan rejections. The precision-recall curve, illustrated in Figure 10, further emphasizes this imbalance, with the model favoring recall over precision.

Despite these limitations, Logistic Regression's short training time of 4.98 seconds (refer to Table 3) makes it a computationally efficient option for quick assessments. However, its use in high-stakes financial environments would require careful threshold adjustments to mitigate the high false positive rate, which is also evident in its relatively high false positive rate of 0.3389 (refer to Table 3). For a more comprehensive evaluation of the model's performance, including ROC curves, see Figure 11 in the Appendices.

Random Forest

The Random Forest model provided a robust performance, particularly in handling complex, non-linear relationships within the dataset. It achieved a cross-validation ROC AUC of 0.7082 and a test accuracy of 0.6473, as reported in Table 3. These results indicate that the model is fairly effective, but its performance is slightly behind that of Logistic Regression in terms of ROC AUC.

The model's precision of 0.3050 and its relatively high false positive rate of 0.3554 (refer to Table 3) suggest that it tends to overpredict defaults, leading to a significant number of false positives (139,038, as seen in Table 3). This issue could be particularly problematic in a financial context where incorrect predictions have tangible consequences.

Additionally, the computational demands of Random Forest are considerable, with a training time of 146.71 seconds and a test time of 599.63 seconds (refer to Table 3), making it less suitable for applications requiring rapid decision-making. The model's performance across different metrics is further visualized in Figures 9 and 10, which highlight its trade-offs between precision and recall.

Gradient Boosting

Gradient Boosting emerged as the best-performing model, achieving the highest ROC AUC scores in both cross-validation (0.7162) and testing (0.6574), as shown in Table 3. This model's strength lies in its ability to sequentially correct errors from previous iterations, which is reflected in its superior recall of 0.6742. This higher recall suggests that Gradient Boosting is more effective at identifying actual defaults compared to the other models.

The model's precision remains relatively low at 0.3075, and its false positive rate is the highest among the three models at 0.3595 (refer to Table 3), leading to 140,627 false positives. This trade-off between precision and recall is crucial in contexts where the cost of false positives is significant. The complexity of Gradient Boosting is also evident in its training and test times, which are 148.03 seconds and 638.04 seconds, respectively (refer to Table 3).

Figures 11 and 9 provide a comprehensive comparison of the model's performance, while the precision-recall curve in Figure 10 illustrates its effectiveness

in managing the trade-offs between these key metrics. Despite its strong performance in recall, the model's high computational costs and potential for overfitting must be carefully managed in practical applications.

CONCLUSION

This study aimed to critically assess the viability of machine learning models—Logistic Regression, Random Forest, and Gradient Boosting—in predicting loan defaults using data from LendingClub, with a particular focus on their performance in real-world financial contexts where accuracy and reliability are paramount. The findings suggest that while these models hold significant potential, their current form is marked by notable limitations that could hinder practical deployment in financial risk management.

The general objective of evaluating the models in a real-world financial context was approached by rigorously testing the models against a dataset that spans a particularly volatile period, including the COVID-19 pandemic. This period introduced significant fluctuations in financial behavior, which possibly reflected in the models' performance, though this hypothesis remains a suggestion for future studies. Although Gradient Boosting achieved the highest recall and ROC AUC scores, its high rate of false positives poses a substantial risk in financial applications, where incorrect predictions could lead to unnecessary loan rejections, impacting profitability and customer trust. Logistic Regression, while simpler and faster, struggled to balance precision and recall, resulting in potentially dangerous false negatives (missed defaults) alongside false positives. Random Forest offered a balanced approach but was still impeded by a high false positive rate and considerable computational demands, making it less suitable for environments that require rapid decision-making.

After a thorough evaluation, it is clear and conclusive that this investigation has fully achieved its general objective of assessing the viability of machine learning models in predicting loan defaults within a real and complex financial context. The analysis provided a comprehensive view of the strengths and limitations of these models, with a precise focus on handling complex datasets, such as those from Lending Club. This process revealed the models' ability to adapt to unexpected financial behaviors while also highlighting the challenges faced,

particularly regarding the high rate of false positives and significant computational demands. Despite these difficulties, the research effectively demonstrated the complexities involved in the practical application of these models in financial environments. Therefore, the study concludes that, while continuous improvement is necessary, the objective has been satisfactorily achieved, offering a clear understanding of the current limitations and the significant potential of machine learning in financial risk management.

In addressing the specific objectives:

1. **Evaluate Model Viability for Financial Applications:** The study demonstrated that the practical application of these models in the financial sector is challenged by their propensity for high false positive rates, particularly in the case of Gradient Boosting. This limitation is especially critical in loan approval processes, where errors in classification can lead to significant financial losses. Although the models show potential, they require further refinement to be considered viable for real-world applications.
2. **Analyze Model Performance Metrics:** A thorough evaluation of each model's performance was conducted using key metrics such as accuracy, precision, recall, F1 score, and ROC AUC. This analysis revealed important trade-offs between these metrics. For instance, while Gradient Boosting excelled in recall, its high false positive rate underscores the need for improved calibration or alternative approaches to mitigate these errors. Logistic Regression, though easier to interpret, was less effective in capturing the full spectrum of loan default risks, particularly in scenarios where precision is critical.
3. **Investigate the Impact of Data Preprocessing:** The study underscored the crucial role of data preprocessing in shaping the models' performance. Techniques such as feature selection and class balancing were integral to the models' development. However, the results suggest that more advanced preprocessing methods or alternative approaches to handling class imbalance, beyond Random UnderSampling, may be necessary to enhance the models' ability to generalize and reduce false positives.

4. **Propose Improvements for Future Research:** Based on the findings, several avenues for future research were identified, including the integration of more diverse data sources, such as real-time economic indicators, to enhance predictive power. Additionally, the exploration of advanced machine learning techniques, like deep learning and hybrid models, could potentially improve performance, particularly in reducing false positives. Future research should also focus on developing models that strike a balance between interpretability and predictive accuracy to better meet the demands of financial risk management.

In conclusion, while the machine learning models evaluated in this study demonstrate significant potential for predicting loan defaults, their current limitations—especially in terms of high false positive rates and computational demands—suggest that they are not yet ready for full-scale deployment in the financial industry. The complex nature of financial data, especially during periods of economic upheaval like the COVID-19 pandemic, adds layers of difficulty that these models must be adapted to handle. This study's critical assessment highlights the importance of continuous refinement and adaptation of these models to ensure they can meet the rigorous demands of real-world financial decision-making.

The broader implications of this research suggest that, while data science and machine learning offer powerful tools for financial risk management, they must be applied with a deep understanding of their limitations and the trade-offs involved. The use of public datasets like Lending Club's presents opportunities for exploring various approaches, from simple classifiers to advanced deep learning techniques, allowing for continuous improvement and innovation in model development. Furthermore, the potential for combining different models and adopting hybrid learning approaches offers a promising path forward, where the strengths of various algorithms can be harnessed to create more resilient and reliable predictive systems.

Future studies should focus on refining these models, exploring new data sources, and developing techniques that not only enhance predictive power but also ensure transparency, regulatory compliance, and practical applicability in the financial sector. By aligning these models with the operational realities of financial institutions, data science can maximize its impact while minimizing the risks associated with machine learning applications in finance.

REFERENCES

- Błaszczynski, J., de Almeida Filho, A. T., Matuszyk, A., Szela, M., and Słowiński, R. (2021). Auto loan fraud detection using dominance-based rough set approach versus machine learning methods. *Expert Systems with Applications*, 163:113740.
- Chopra, A. and Bhilare, P. (2018). Application of ensemble models in credit scoring models. *Business Perspectives and Research*, 6(2):129–141.
- Malekipirbazari, M. and Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications*, 42(10):4621–4631.
- Simão, S. B. S. (2023). Machine learning applied to credit risk assessment: Prediction of loan defaults. Dissertation, NOVA Information Management School, Universidade Nova de Lisboa. Dissertation presented as the partial requirement for obtaining a Master's degree in Data Science and Advanced Analytics, specialization in Data Science.
- Yosza Dasril, Much Aziz Muslim, M. F. A. H., Jumanto, J., and Prasetyo, B. (2023). Credit risk assessment in p2p lending using lightgbm and particle swarm optimization. 9(1):18–28.

APPENDICES

This section presents additional visual representations of the model performances and data treatments discussed in the main sections of the document.

Data Treatment and Visualization

The following figures illustrate the distribution of various variables before and after data treatment, highlighting the impact of the preprocessing steps applied to improve model performance.

Figure 1 - Data distribution before and after treatment for variables in the first group.

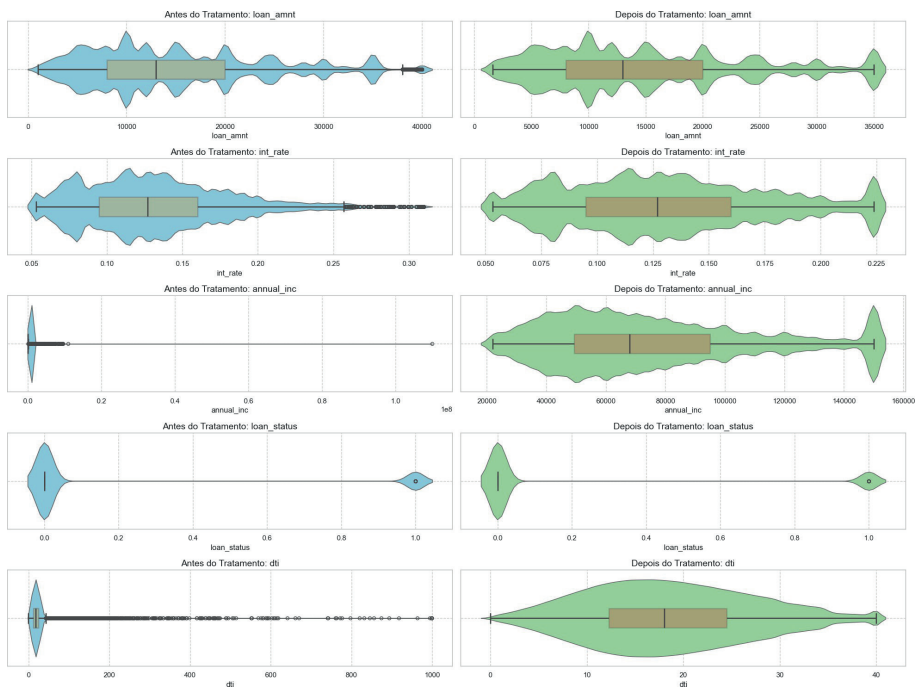


Figure 2 - Data distribution before and after treatment for variables in the second group.

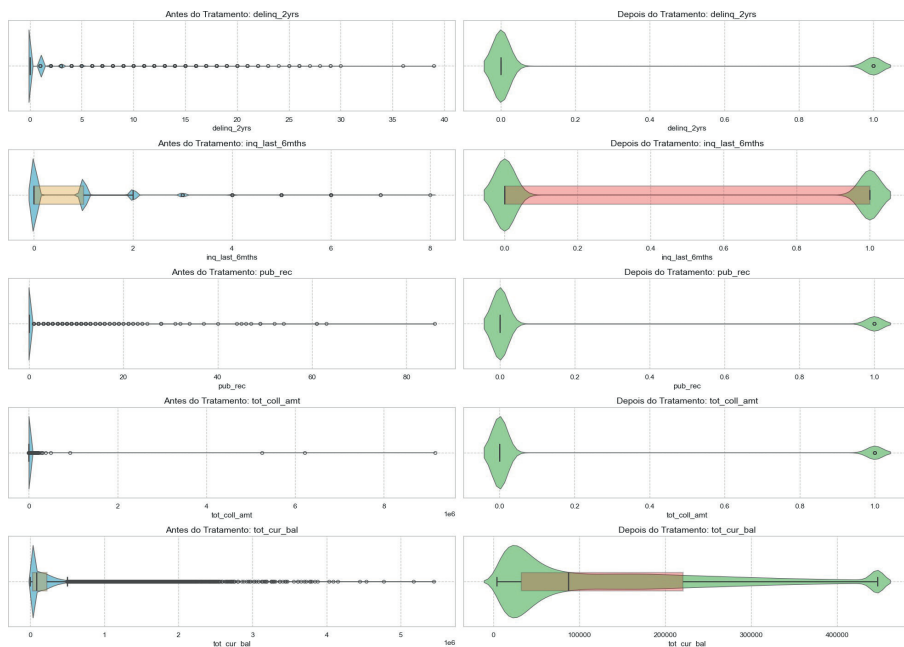


Figure 3 - Data distribution before and after treatment for variables in the third group.

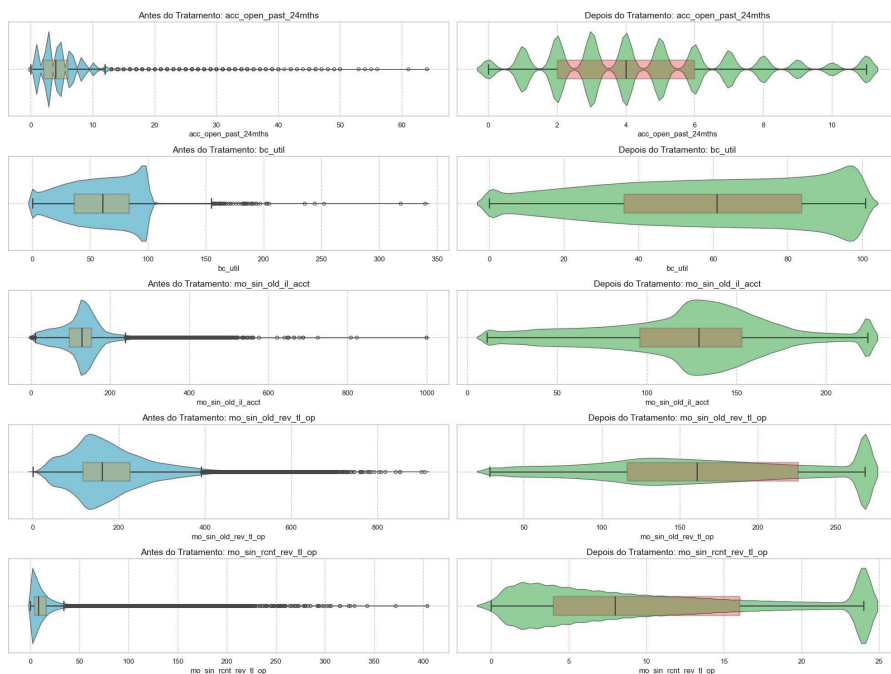


Figure 4 - Data distribution before and after treatment for variables in the fourth group.

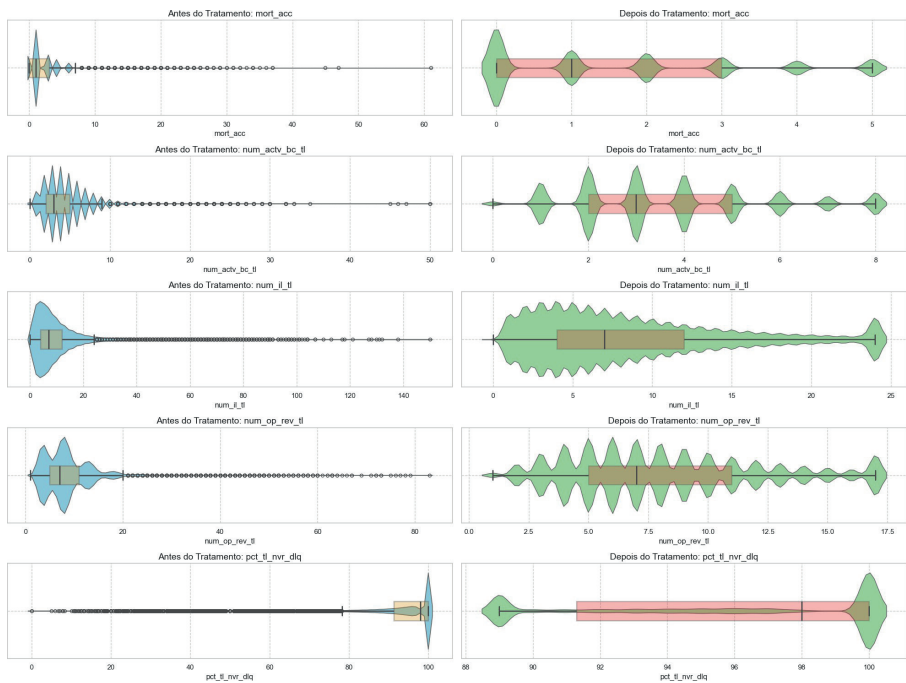
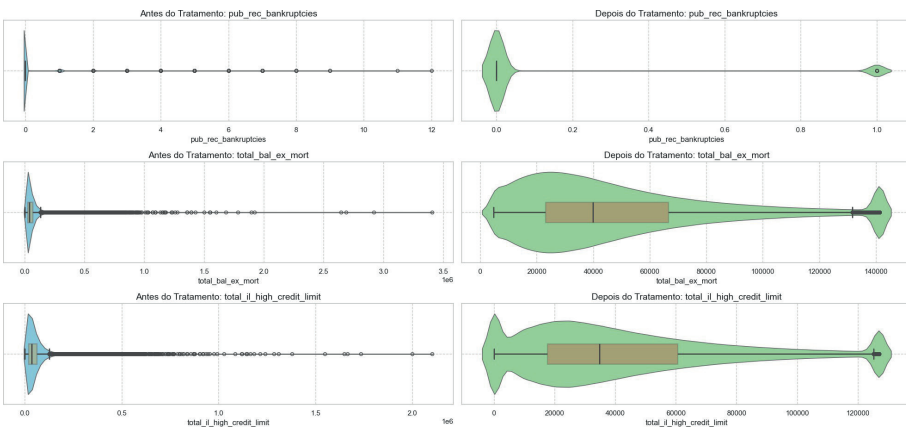


Figure 5 - Data distribution before and after treatment for variables in the fifth group.



Feature Selection and Correlation Analysis

Figure 6. Correlation matrix before feature exclusion.

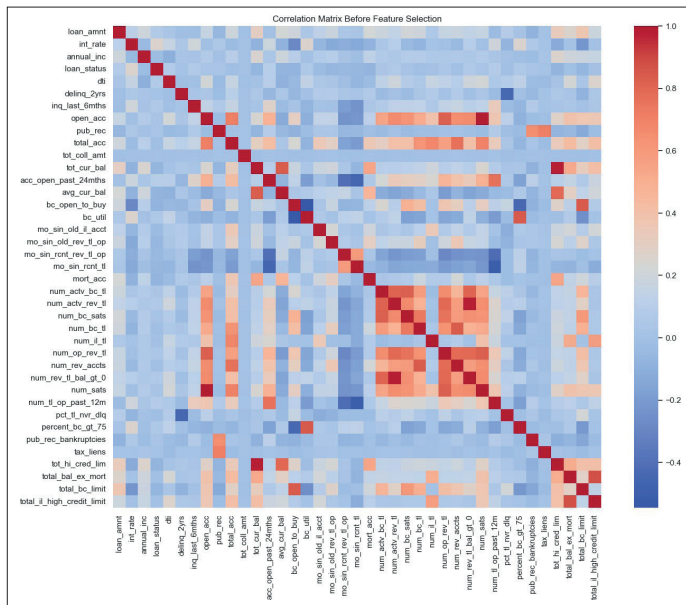
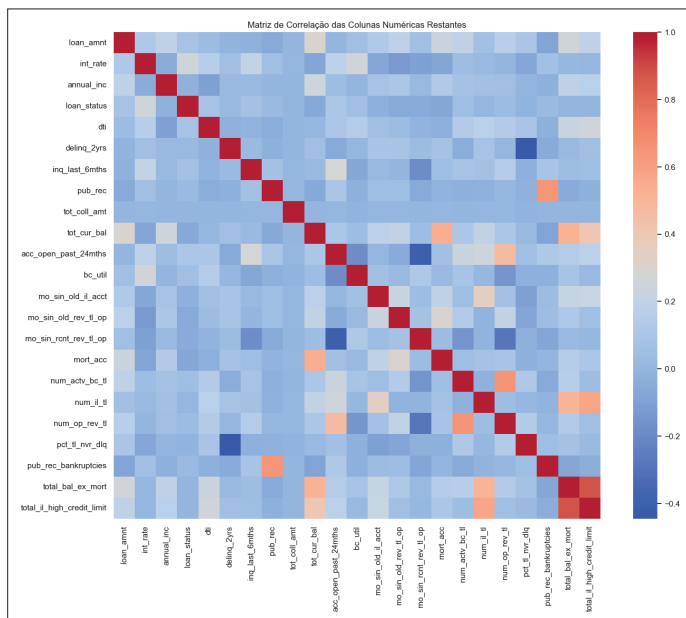
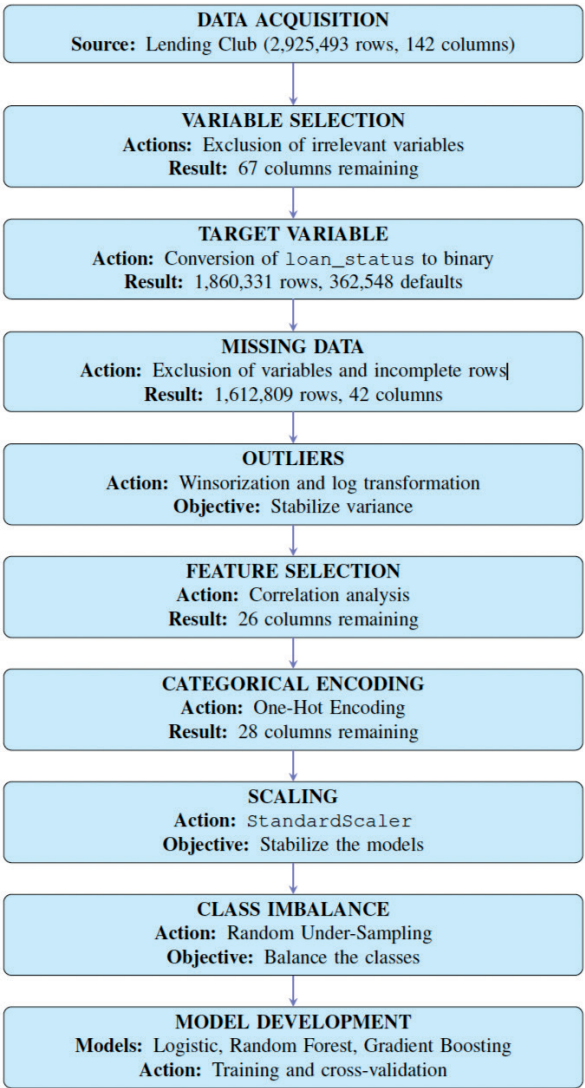


Figure 7. Correlation matrix of the remaining numerical columns after feature exclusion.



Model Development Workflow

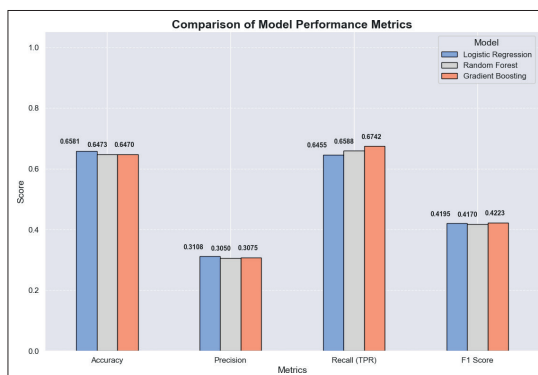
Figure 8 - Summary of the Data Processing and Model Development Workflow.



Model Performance Metrics

To better understand the model performance discussed in the main document, the following figures present a detailed comparison of key metrics, including accuracy, precision, recall, and F1 score, as well as the training time required by each model.

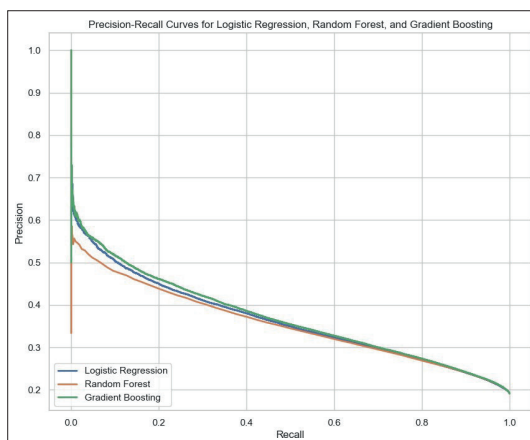
Figure 9 - Comparison of Model Performance Metrics. This figure illustrates the performance of Logistic Regression, Random Forest, and Gradient Boosting models across four key metrics: Accuracy, Precision, Recall (TPR), and F1 Score. The chart highlights how each model balances these metrics, providing a clear visual comparison.



Precision-Recall Curves

Precision-recall curves provide insights into the trade-offs between precision and recall at different decision thresholds for each model.

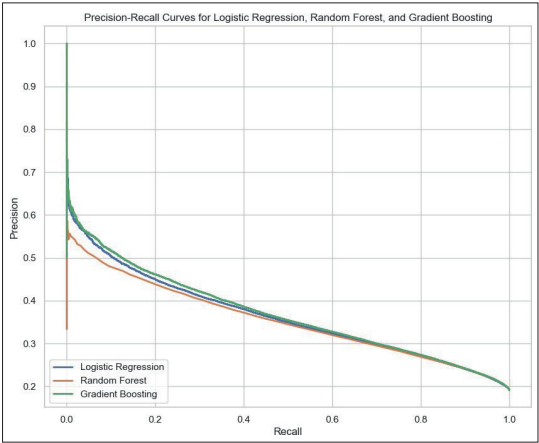
Figure 10 - Precision-Recall Curves for Logistic Regression, Random Forest, and Gradient Boosting. This figure shows the precision-recall trade-offs for each model, providing a visual assessment of how well each model handles the balance between precision and recall across different thresholds.



ROC Curves

ROC curves are another important metric for model evaluation, focusing on the trade-off between the true positive rate and false positive rate.

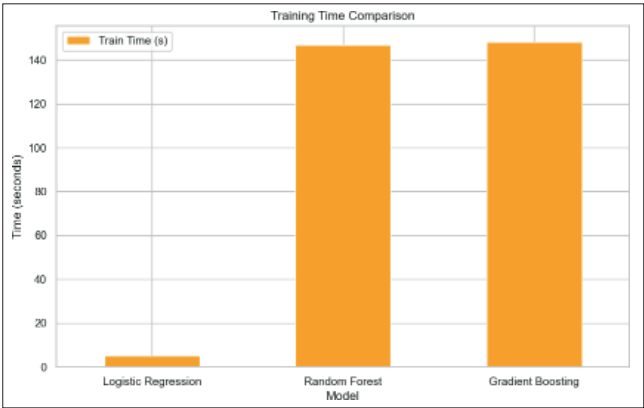
Figure 11 - ROC Curves for Logistic Regression, Random Forest, and Gradient Boosting. This figure demonstrates the ROC curves for each model, highlighting their ability to balance the true positive rate and false positive rate across various thresholds.



Training Time Comparison

Finally, training time is a critical factor when considering the deployment of these models in a production environment.

Figure 12 - Training Time Comparison. This figure compares the training times (in seconds) of Logistic Regression, Random Forest, and Gradient Boosting models. The chart highlights the significant computational resources required by the more complex models, particularly Random Forest and Gradient Boosting, compared to Logistic Regression.



AN ATTENTION-BASED REGRESSION METHOD FOR REAL ESTATE VALUATION

Renato Quinderé Carneiro

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)
renatoquindere@hotmail.com

Filipe Alves Neto Verri

Division of Computer Science – Aeronautics Institute of Technology (ITA)

ABSTRACT

Location is often regarded as the most important feature in real estate valuation, and it is also difficult to encode it as a model variable. We present an Attention-Based Regression, a method inspired by graph attention network, that leverages the masked attention mechanism to mimic the sales comparison approach adopted by realtors. Our method provides a data-driven approach to encode property location that does not require specific neighborhood knowledge, allowing the model to cover larger geographic regions. We compare our method to benchmark models and find that our method has performance that matches state-of-the-art boosted ensembles.

INTRODUCTION

Residential property valuation plays a key role in mortgage origination, given that the property is the collateral for the loan. An overestimated property valuation would imply an understated Loan-to-value ratio (LTV), implying a higher risk to the lender and a lower down payment to the borrower [Mayer and Nothaft 2022]. Such is the importance of real estate valuation, that problems in valuation methods could even contribute to systemic economic crises such as subprime mortgage crises [Ding and Nakamura 2016].

Among the issues in property valuation, [Cheloti and Mooya 2022] listed three major concerns: valuation variations, client influence, and the use of heuristics. Valuation variations refer to error margins, which could be as wide as 30%. Client influence concerns the possibility of the client manipulating valuation results, while heuristics relate to the appraiser anchoring his estimates in previous valuations or past transactions.

One way to overcome these issues is the adoption of Automated Valuation Models (AVM), which also reduces the costs involved in property appraisal [Pace *et al.* 2002]. An AVM is computer software with a mathematical model that predicts property prices based on a statistical analysis of previously collected data [IAAO 2018].

Real Estate Valuation models traditionally followed the Hedonic Pricing Model (HPM), which states that the value of a heterogeneous good can be decomposed into a sum in which each summand corresponds to the effect of a factor or characteristic of the item. Usually, the Hedonic Pricing Model is implemented by regression analysis [Björklund *et al.* 2002], where the influence of each characteristic is given by the regression coefficients.

Among those characteristics, it is well established that the property location is one of the most influential factors in property value. Attempts in the econometric approach to encoding location as a feature went from the concept of real estate sub-market to a sophisticated spatial regression model [Lisi 2019]. Regarding the sub-market concept, the study by [Bourassa *et al.* 2003] is of particular interest. The author compares the effect of adopting sub-markets as small geographic areas or as an aspatial cluster in the hedonic regression model and concludes that models that consider geographic sub-markets have smaller

errors in the task of house price estimation, which implies that geography is what makes sub-markets important.

The rise of artificial intelligence and the search for reducing model errors has directed much research effort to develop AVMs with machine learning algorithms. For instance, [Valier 2020] identified 178 articles that applied machine learning algorithms to property valuation and, based on the analysis of 53 of them, reported that the majority of papers concluded that there was an accuracy gain when comparing machine learning models to the traditional hedonic model.

Although the studies analyzed by [Valier 2020] only used structured data, many recent papers on real estate valuation include unstructured data to reduce the model's error. The main motivation is to consider characteristics that are difficult or expensive to quantify but are important to the buyer in his decision process, such as neighborhood prestige, accessibility, and distance to points of interest [Law *et al.* 2019].

Researchers tested distinct deep learning architectures to incorporate unstructured data that encode information on neighborhoods. [Baur *et al.* 2023] argue that property descriptions may carry information not included in structured data about the neighborhood, distance to points of interest, and even past renovations. This study includes numeric representations of property descriptions in several machine learning models and concludes that the use of this kind of feature was beneficial in reducing prediction error. On the other hand, [Law *et al.* 2019] adopted convolutional layers to encode information from satellite and Google Street View images. Similarly, [Nouriani and Lemke 2022] extracted features from images, but included interior and exterior images of the house as well as satellite images of the neighborhood.

Although inclusion of unstructured data reduces errors, it is necessary to collect a great volume of additional information, such as pictures and text, which may not be available or even updated. This study proposes a simpler model that encodes the location feature based on the neighbor's information in the training data. This is achieved through the masked attention mechanism, allowing our model to leverage geographical coordinates information to capture the neighborhood influence on the property value and leading to a model that mimics realtors' direct comparison approach to property valuation. Results indicate that our approach

reaches state-of-the-art model performance without needing specific information on the neighborhood.

Research Problem Statement

The problem that this study addresses is related to the difficulty of encoding the location of a property as a feature in a regression task and is stated as **“How to predict real estate property prices considering the geographical sub-markets?”**

Main Objectives

The objective of this research is to propose a prediction model for house values in which the location’s influence on prices is extracted from the training data.

Specific Objectives

The specific objectives present, in detail, the intended actions to be achieved and establish a close relationship with the particularities related to the general objective. In this way, the following specific objectives were identified:

- Propose an Attention-based regression model in which the location influence on prices is extracted from the training data;
- Evaluate the proposed model in a real estate dataset;
- Compare results with benchmark methods;

Contributions

The method proposed in this paper provides a data-driven approach to encode property location in valuation models while delivering performance that matches state-of-the-art algorithms. To the best of our knowledge, no other work proposed the application of masked attention mechanism to grasp the property location’s influence on its value, hence our proposition.

METHODS

In the following sections, we describe the attention-based regression model proposed in this study, the dataset used for evaluation, and the evaluation metrics.

Attention-Based Regression

The architecture we propose is adapted from the graph attention network (GAT) introduced by [Velickovic *et al.* 2018], which exchanges graph convolutions by attention mechanism in graph networks. The performance of graph attention was reported to be comparable with prior methods.

Attention-based models were first introduced to machine translation to allow the algorithm to search the relevant sentence parts to predict the translated word [Bahdanau *et al.* 2016]. The attention mechanism allows the network to adopt weights that vary according to the input vector, meaning that the part of the text that will be focused depends on the input word [Bishop and Bishop 2024].

The concept of attention is the basic building block of the transformer architecture [Vaswani *et al.* 2017], which outperformed the Recurrent Neural Network approach to natural language processing [Bishop and Bishop 2024]. The success of the transformer architecture led to the development of attention-based models for other domains, such as image classification, where the vision transformer attained results comparable to the state-of-the-art models [Dosovitskiy *et al.* 2021].

Graph attention networks adopt mask self-attention where only the nearest nodes are used to compute attention scores. The concept of masked attention on graphs is suitable to model real estate sub-markets given that no regular grid is required and that the attention mechanism could mimic the sales comparison approach adopted by realtors, which estimates the value of a property based on comparable properties sold in the same geographic area [Lisi 2019]. The masked Attention mechanism ponders values based on similarity while only reaching properties in the specified neighborhood.

Figure 1 - Schema of the proposed architecture, which is composed of a single attention head, a residual connection, a fully connected network, and three inputs. The Query corresponds to the subject property features after a linear map. Keys are formed from the features of the training data in the same geographic region of the input property, and Values are the keys with the addition of the distance to the target house and unit prices of the neighbors. Keys and values also pass through a linear map before the attention mechanism. The output of the masked attention mechanism is then concatenated with the input feature vector and passed through a fully connected network.

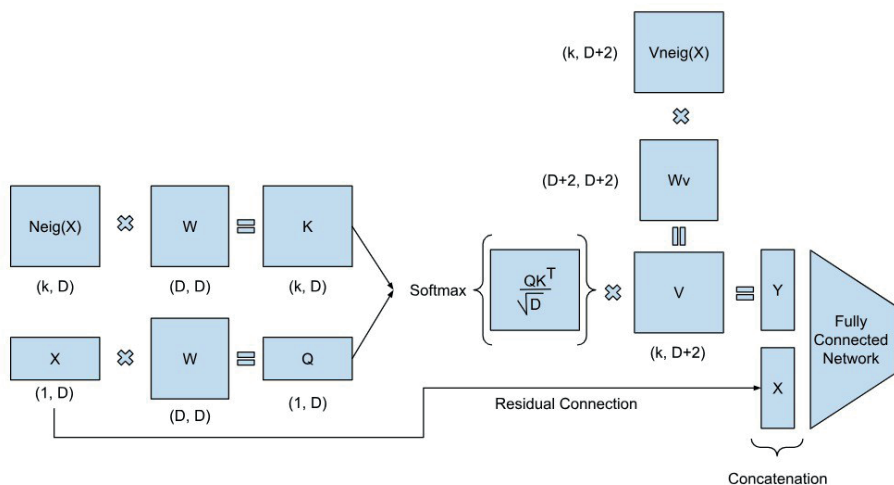


Figure 1 shows the schema of the proposed architecture composed of a single attention head, a residual connection, a fully connected network, and three inputs. The X input is the feature vector of the subject property, with dimensions $1 \times D$, and the $Neig(X)$ input is the feature matrix of the k nearest neighbors of X , with dimensions $k \times D$, where each line corresponds to a property and the columns correspond to the features. X and $Neig(X)$ are projected to a shared subspace by a $D \times D$ matrix W , resulting in the query Q and keys K respectively, where that projection is learned from the training data.

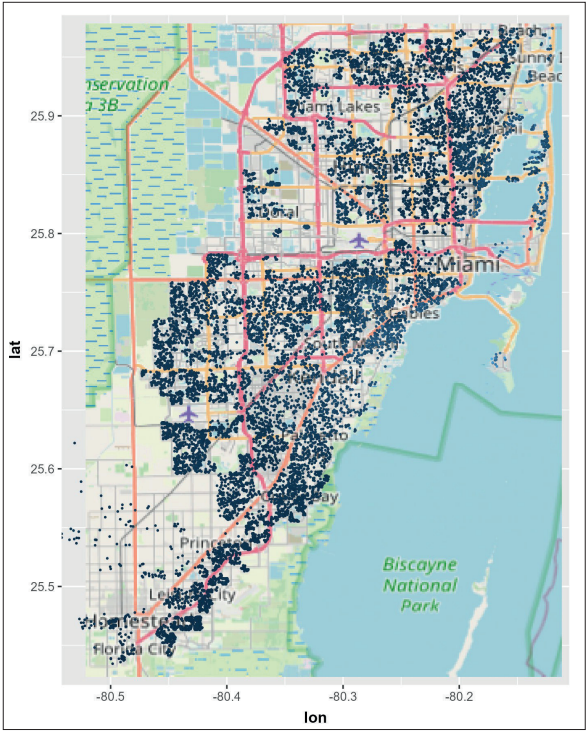
Values are derived from the $Vneig(X)$ input, which is the $Neig(X)$ feature matrix with the addition of two columns: the unit sale price of these properties and its distances to the subject property encoded in X . The $Vneig(X)$ matrix is then passed through a linear map to form the Value Matrix V . Attention scores computed using scaled dot-product attention from K and Q , as described in [Vaswani *et al.* 2017], are used to ponder the value matrix V and obtain the Y vector as the output of the masked attention mechanism. By the end, X and Y are concatenated and passed to a fully connected neural network. This residual connection intends

to allow the model to go beyond the neighborhood and also consider the global structure of the property market.

Dataset

The dataset comprises information about 13932 single-family houses sold in Miami-Dade County in 2016 [Mayer *et al.* 2022], available at api.openml.org/d/43093. Miami-Dade County is formed by 34 cities in Florida with a population of 2,701,767. The county has 1,074,685 housing units with a median value between \$300,000 and \$500,000 [U.S. Census Bureau 2020]. Figure 2 shows the location of the houses present in the dataset.

Figure 2 - Map of Miami-Dade County with dataset houses marked as blue dots, it can be seen that the properties cover almost all urban areas of the county.



This dataset contains 17 features, including information about the property size, age, quality, special features, airplane noise, month sold, and the distance to

points of interest, as described in Table 1. The locational attributes in the data set are not necessary for the proposed architecture and are used only in the baseline methods. Additionally, units are converted to the metric system, and the adopted target variable is the unit sale price, obtained through the ratio of the sale price and constructed area. This study does not consider the month sold variable, given that all sales occurred within one year.

Table 2 shows the descriptive statistics for the dataset's numerical features, where it can be seen that the data covers a wide range of house sizes, with living areas ranging from 79 to 580 square meters and land size varying from 116 to 5,300 square meters. Numerical features in this dataset have distinct orders of magnitude, and Figure 3 reveals that the features TOT LVG AREA, LND SQM, and SPEC FEAT VAL are severely right-skewed. Therefore, we choose to pre-process numerical variables by standardization, transforming variables to zero mean and unit variance.

The analysis of categorical features detects class imbalance problems on both structure quality and avno60plus variables, as illustrated in Figure 4. This way, the structure quality feature is reformulated into 2 classes: class 0 groups the original structure quality 1, 2, and 3, and class 1 comprises structure quality 4 and 5. The variable avno60plus is not included in this study.

Model Training and Evaluation

The proposed model is evaluated using the repeated hold-out method, with 10 iterations and test sets comprising 20% of the data. The evaluation metric adopted is the coefficient of determination R^2 , chosen due to the wide use of regression analysis in real estate valuation and given that research in this area usually reports this metric. The model is trained using *tensorflow* [Abadi *et al.* 2015] and the following features are used: TOT LVG AREA, LND SQM, SPEC FEAT VAL, age, and structure quality. The hyper-parameters learning rate, optimizer, loss function, batch size, number of epochs, number of neighbors, and the structure of the fully connected network are tuned with a validation set composed of 20% of the training set, with the resulting values shown in Table 3.

Table 1 - Features Description adapted from [Mayer et al. 2022]. Locational variables are marked with*.

Feature	Type	Description
LATITUDE	Numeric	Geographic Coordinate
LONGITUDE	Numeric	Geographic Coordinate
TOT LVG AREA	Numeric	House constructed area in m^2
LND SQM	Numeric	Land area in m^2
SPEC FEAT VAL	Numeric	Value of the special features in US dolars
age	Numeric	House age in years
structure quality	Categorical	Structure's quality
month sold	Numeric	Month in which the house was sold
avno60plus*	Categorical	Indicates if airplane noise is over the ac cepted limit
RAIL DIST*	Numeric	Distance to the nearest rail line in meters
OCEAN DIST*	Numeric	Distance to the ocean in meters
WATER DIST*	Numeric	Distance to the nearest body of water in meters
CNTR DIST*	Numeric	Distance to the Miami central business district in meters
HWY DIST**	Numeric	Distance to the nearest highway in meters
SUBCNTR DI*	Numeric	Distance to the nearest subcenter in meters
SALE PRC	Numeric	Sold price in US dolars
PRC UN	Numeric	Unit sold price in US dolars per m^2

Table 2 - Numerical features descriptive statistics.

Feature	Min	Max	Mean	Median	σ
TOT LVG AREA	79.34	584.08	191.20	174.43	75.58
LND SQM	115.94	5,301.42	800.91	696.77	563.93
SPEC FEAT VAL	0.00	175,020.00	9,562.49	2,765.50	13,890.97
age	0	96	30.67	26	2115
month sold	1	12	6.66	7	3.30
RAIL DIST	3.20	9,028.63	2,544.64	2,166.00	1,883.06
OCEAN DIST	71.96	23,087.05	9,659.42	8,699.53	5,362.98
WATER DIST	0.00	15,361.86	3,645.49	2,110.01	3,637.18
CNTR DIST	1,166.04	48,760.84	20,875.85	20,071.81	9,756.18
HWY DIST	27.49	14,681.39	2,354.21	1,877.49	1,849.81
SUBCNTR DI	445.86	33,696.80	12,531.87	12,530.30	6,754.92
SALE PRC	72,000.00	2,650,000.00	399,941.93	310,000.00	317,214.68
PRC UN	475.44	14,477.78	2,033.02	1,758.59	1,020.26

Figure 3 - Histogram for TOT LVG AREA, LND SQM, and SPEC FEAT VAL features, which show right-skewed distribution for the three variables.

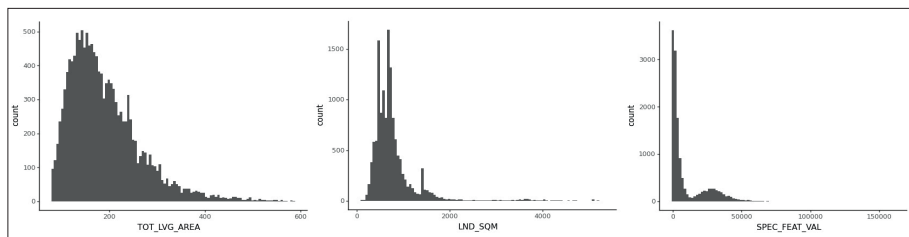


Figure 4 - Categorical features histogram, where graphs show severe class imbalance problems for both variables.

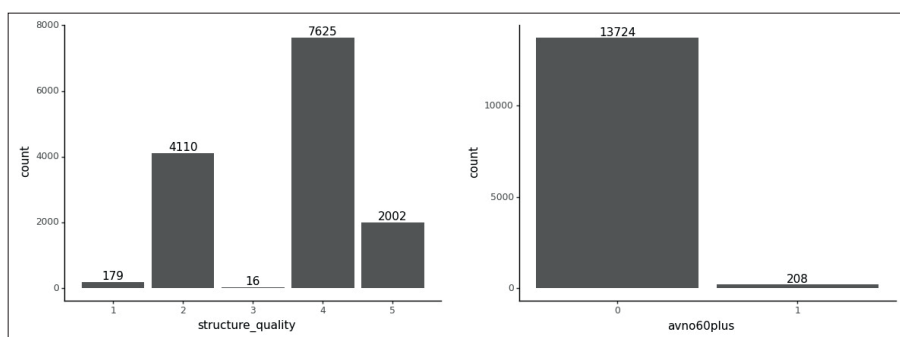


Table 3 - Model's Hyper-parameters tuned with validation set.

Hyper-Parameter	Tuned Values
Learning Rate	Decaying rate starting with 10^{-2} and reduced by a factor of 10 every 40 epochs
Optimizer	Adam
Loss Function	Mean Squared Error
Batch Size	128
Number of Epochs	200
Number of Neighbors	15
Fully Connected Network Structure	3 Layers with 32, 16 and 8 units respectively. All units uses Leaky ReLU activation function with negative slope of 0.01. Dropout with 0.10 probability is applied to the first two layers.

Results are compared to benchmark methods with diverse inductive bias: Linear Regression, Regression Tree, KNN, Random Forest, Gradient Boosting, XGBoost, light GBM, and multi-layer perceptron neural network. These methods are trained with the full feature set, which includes the following locational variables: RAIL DIST, OCEAN DIST, WATER DIST, CNTR DIST, HWY DIST, and SUBCNTR DI.

Linear Regression is fitted using the *statsmodels* package for *Python* [Seabold and Perktold 2010], and the dependent variable is transformed through a logarithmic function. Regression Tree, KNN, Random Forest, and Gradient Boosting are trained using *Scikit-Learn* for *Python* [Pedregosa *et al.* 2011], while XGBoost and light GBM are trained using packages *xgboost* [Chen and Guestrin 2016] and *lightgbm* [Ke *et al.* 2017]. The hyper-parameters for these models are tuned using 5-fold cross-validation, with results shown in Table 4.

The Multi-layer perceptron neural network is trained using *tensorflow* [Abadi *et al.* 2015], and hyper-parameters are tuned using a validation set comprising 20% of the training set. The network is composed of an input layer, 3 intermediate layers, and a single unit output layer. The intermediate layers have 32, 16, and 8 units respectively, with leaky ReLU activation function and a negative slope of 0.01. Additionally, a 0.05 rate dropout is added to the first hidden layer. The data used to train the MLP is pre-processed, scaling to zero mean and unit variance.

RESULTS

Figure 5 shows the Mean Squared Error (MSE) loss function evolution during training for one iteration of the repeated hold-out. The loss stabilizes after 50 epochs, and reaches

Table 4 - Benchmark Models Hyper-parameters tuned with 5-fold cross-validation.

Hyper-Parameter	Tuned Values
Regression Tree	Max Depth: 10, Splitter: best, Criterion: Friedman MSE
KNN	Number of Neighbors: 10 (Geographical Neighbors)
Random Forest	Number of Estimators: 250, Max Features: Square Root, Criterion: Squared Error
Gradient Boosting	Max Depth: 9, Loss: Huber, Learning Rate: 0.05, Criterion: Squared Error
XGBoost	Subsample: 0.6, Number of Estimators: 125, Max Depth: 7, Learning Rate: 0.1, gamma: 0.25, Sub sample ratio of columns: 0.7
Light GBM	Number of Estimators: 250, Max Depth: 12, Learning Rate: 0.1

Table 5 - Average R^2 and RMSE for the 10 iterations of the repeated hold-out, with standard deviation.

Method	$\overline{R^2}$	σ_{R^2}	$\overline{RMSE}$	σ_{RMSE}
Linear Regression	0.47	0.02	734.74	34
Regression Tree	0.69	0.03	562.84	25
KNN	0.75	0.02	506.00	21
MLP	0.80	0.02	456.56	30
Random Forest	0.82	0.02	425.42	26
Gradient Boosting	0.83	0.02	419.29	31
XGBoost	0.84	0.02	406.36	23
Light GBM	0.84	0.01	398.01	21
Attention-Based Regression	0.85	0.02	394.09	32

the lowest value after 75 epochs for both training and validation loss. The absence of an ascendant branch at the validation loss curve indicates that the model does not suffer from overfitting.

Table 5 shows the average R^2 for the 10 iterations of the repeated hold-out. Linear Regression presents the poorest result with an average R^2 of 0.47, followed by Regression Tree and KNN, with R^2 of 0.69 and 0.75 respectively. MLP, Random Forest, and Gradient Boosting have attained R^2 of 0.80, 0.82 and 0.83, respectively. XGBoost, Light GBM, and Attention-Based Regression are the top performers, with R^2 of 0.84, 0.84 and 0.85. Regarding the model's variability, the best results are obtained by light GBM, where the standard deviation of the Root Mean Squared Error (RMSE) is 21, which corresponds to 5% of the average RMSE. Attention-based Regression attains standard deviation similar to Gradient Boosting and MLP, corresponding to 8% of the average RMSE.

The good performance of boosted ensembles is expected as these techniques dominate machine learning competitions when data is tabular [Arik and Pfister 2020]. In this way, it is promising that the proposed architecture has comparable or even better performance, given that it does not require collecting specific information on the neighborhood, such as the location of points of interest, allowing the model to cover a larger geographical region.

Figure 5 - Mean Squared Error (MSE) loss function evolution during training. Training loss is showed in dashed line, while validation loss is showed in dotted line.

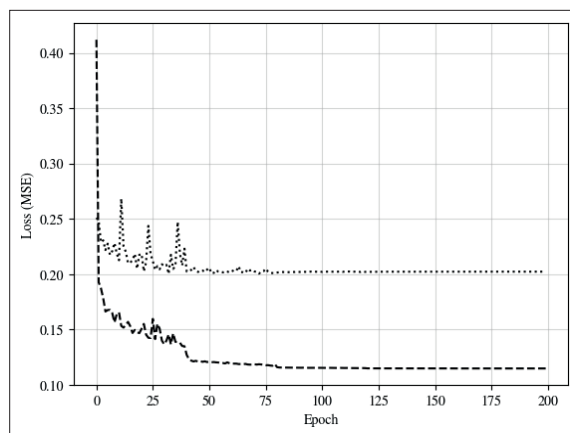


Figure 6 shows the geographic distribution of test set error for some of the models evaluated. Linear regression exhibits a high spatial error correlation, while KNN error seems to be randomly spread across the territory. Regression tree and MLP have similar patterns, with bigger errors in the coastal region north of Miami. This pattern is also presented on XGBoost, but it seems to be attenuated. On the other hand, the attention-based regression shows a more uniform error distribution, possibly due to the masked attention mechanism.

Among similar works, we highlight [Mayer *et al.* 2022] that provided the dataset used in our study. This work proposes a star architecture for real estate valuation focusing on model interpretability. The authors also report the R^2 but, despite using the same data set, results are not comparable since the target variable adopted is the log total sales price while we use the unit sales price.

CONCLUSION

We propose an Attention-based regression method for real estate valuation, that provides a data-driven approach to encode property location that does not require specific neigh-borhood knowledge. The architecture leverages the geographical coordinates information to capture the neighborhood influence on the property value. The masked Attention mechanism ponders values based on

similarity focusing on properties in the specified neighborhood, while the residual connection enables the model to grasp the global structure of the property market.

The method is evaluated in a dataset composed of 13,932 sold houses in Miami-Dade County and the coefficient of determination R^2 is reported. Results are compared to benchmark models and we find that our method has performance that matches state-of-the-art boosted ensembles. We also analyze the error geographic distribution and verify a random distribution across the territory for our method.

Figure 6 - Geographic distribution of test set error (%). the blue color represents negative errors, while yellow color corresponds to positive errors.



Limitations of the method include the need for the territory to be densely populated by training data and output restricted to point estimates of the property value. We recommend future work to test the method with different types of properties and to adapt the architecture to express the model's varying uncertainty among neighborhoods in an interval output. For the latter, we recommend testing bootstrap sampling the nearest neighbors properties at inference time and

reporting maximum and minimum values so that the interval output will have an amplitude that corresponds to the uncertainty level in that neighborhood.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). Tensor-Flow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Arik, S. O. and Pfister, T. (2020). Tabnet: Attentive interpretable tabular learning.
- Bahdanau, D., Cho, K., and Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate.
- Baur, K., Rosenfelder, M., and Lutz, B. (2023). Automated real estate valuation with machine learning models using property descriptions. *Expert Systems With Applications*, 213:957–4174.
- Bishop, C. M. and Bishop, H. (2024). *Transformers*, pages 357–406. Springer International Publishing, Cham.
- Björklund, K., Söderberg, B., and Wilhelmsson, M. (2002). *An Investigation of Property Price Studies*, pages 63–93. Springer US, Boston, MA.
- Bourassa, S. C., Hoesli, M., and Peng, V. S. (2003). Do housing submarkets really matter? *Journal of Housing Economics*, 12:12–28.
- Cheloti, I. and Mooya, M. (2022). Property valuation problems and market context—evidence from kenya. *Journal of Property Research*.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 785–794, New York, NY, USA. Association for Computing Machinery.
- Ding, L. and Nakamura, L. (2016). The impact of the home valuation code of conduct on appraisal and mortgage outcomes. *Real Estate Economics*, 44:658–690.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.
- IAAO (2018). Standard on automated valuation models (avms).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

- Law, S., Paige, B., and Russell, C. (2019). Take a look around: Using street view and satellite images to estimate house prices. *ACM Trans. Intell. Syst. Technol.*, 10(5).
- Lisi, G. (2019). Property valuation: the hedonic pricing model – location and housing submarkets. *Journal of Property Investment and Finance*, 37:589–596.
- Mayer, M., Bourassa, S. C., Hoesli, M., and Scognamiglio, D. (2022). Machine learning applications to land and structure valuation. *Journal of Risk and Financial Management*, 15(5).
- Mayer, Y. G. and Nothaft, F. E. (2022). Appraisal overvaluation: Evidence of price adjustment bias in sales comparisons. *Real Estate Economics*, 50:862–881.
- Nouriani, A. and Lemke, L. (2022). Vision-based housing price estimation using interior, exterior & satellite images. *Intelligent Systems with Applications*, 14:81.
- Pace, R. K., Sirmans, C. F., and Slawson, V. C. (2002). *Automated Valuation Models*, pages 133–156. Springer US, Boston, MA.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.
- U.S. Census Bureau (2020). 2020 census. U.S. Department of Commerce.
- Valier, A. (2020). Who performs better? avms vs hedonic models. *Journal of Property Investment and Finance*, 38:213–225.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks.

ARTIFICIAL INTELLIGENCE MODEL FOR DEMAND PREDICTION THROUGH REMOTE SENSING

Wellington Brosko da Rosa

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

Elcio Hideiti Shiguemori

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

INTRODUCTION

Contextualization

Remote sensing is a rapidly growing research area that involves the collection and analysis of information about the Earth's surface and natural and anthropic phenomena, using sensors on board satellites, aircraft or drones [Chen *et al.*, 2020]. This technology has been applied in various fields, such as environmental monitoring, natural resource management, urban planning and demand forecasting in critical sectors, such as energy, agriculture and transportation [Wang *et al.*, 2019].

In recent years, artificial intelligence (AI) has been increasingly integrated with remote sensing, allowing for more efficient analysis and interpretation and accuracy of the collected data [Zhang *et al.*, 2018]. Machine learning and deep neural networks, in particular, have shown great potential to improve image classification, object detection and time series analysis in remote sensing data [Ma *et al.*, 2019].

Demand forecasting is a crucial aspect for efficient resource and infrastructure management in sectors such as energy, agriculture and transportation [Wang *et al.*, 2019]. The ability to accurately predict future demands allows for optimized resource allocation, cost reduction and minimization of environmental impacts. In this context, the application of AI in remote sensing can provide valuable insights for demand forecasting and informed decision-making.

The successful integration of AI in remote sensing for demand forecasting requires a comprehensive understanding of the applicable AI techniques and methods, as well as the challenges and opportunities associated with this approach. In addition, it is essential to evaluate the performance of different AI approaches in remote sensing in specific case studies in the energy, agriculture and transportation sectors, in order to identify best practices and recommendations for the efficient implementation of these technologies.

Research Problem

How can artificial intelligence be applied in remote sensing to improve demand forecasting?

General Objective

The objective of this research is to build a model to predict demands, using remote sensing associated with artificial intelligence. The aim is to obtain an accurate prediction model to balance supply and demand in a given region.

Specific Objectives

The following specific objectives were identified:

- Review the literature on the use of AI in remote sensing and its applications in demand forecasting.
- Identify the main AI methods and techniques applicable to remote sensing for demand forecasting.
- Evaluate the performance of different AI approaches in remote sensing for demand forecasting in specific case studies in the delivery sector.
- Propose recommendations for the efficient integration of AI in remote sensing and its use in demand forecasting in delivery sectors.

EXPECTED RESULTS

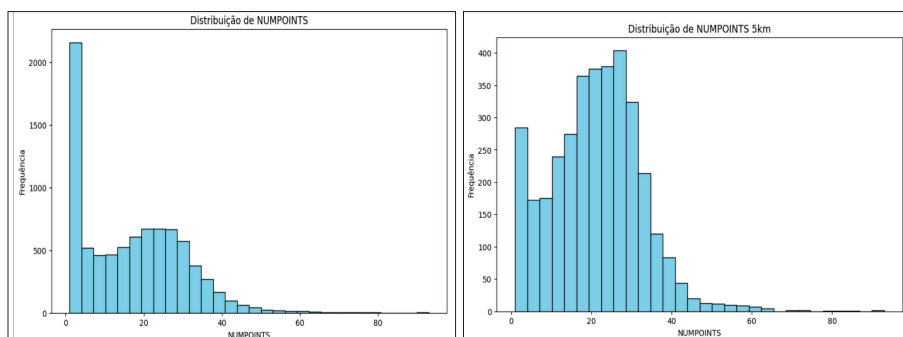
The following results are expected as deliverables that belong to the scope of this research:

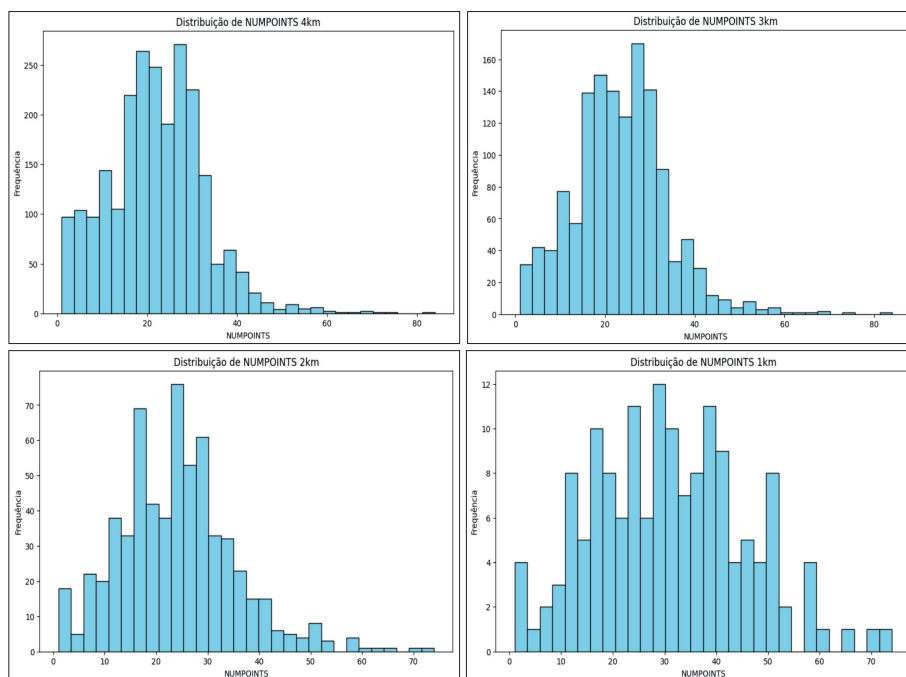
- A demand forecasting model.
- A guide to adapt the forecasts for other objectives.
- A guide for using the forecasting model; and
- A tool to assist in the visualization of demand forecasts.

A Demand Forecasting Model

The demand forecasting model uses artificial intelligence associated with remote sensing to estimate demand in specific geographical areas. The model processes data from points of interest (POIs) extracted from OpenStreetMap (OSM) and grouped via geocoding using GeoHash 7, which subdivides the geographical space into polygons. By analyzing POI density in different regions, the model can identify demand patterns and project future sales, particularly in sectors such as delivery. The modeling was carried out using supervised learning algorithms such as Random Forest, Linear Regression, and Gradient Boosting, with Linear Regression being chosen for its lowest mean absolute error (MAE). The model predicts sales growth opportunities in areas without historical data by leveraging the known sales density expansion.

The analysis of the distribution of Points of Interest (POIs) within each polygon is crucial for accurate demand forecasting and strategic decision-making. POIs represent key locations such as restaurants, shopping centers, gas stations, or hospitals that drive human activity and influence economic behavior. By examining how these POIs are distributed across different geographical regions (represented by polygons, such as those defined by GeoHash 7), valuable insights can be gained into potential demand hotspots and underserved areas. Below are some examples of POI frequency by distance, considering the marco zero of Curitiba as the centroid.





A Guide to Adapt the Forecasts for Other Objectives

The forecasting model can be adapted to other objectives and sectors by modifying input variables or parameters that influence demand in specific contexts. For example, for different sectors beyond delivery, it is possible to adjust the importance of certain types of POIs (such as restaurants or gas stations).

To adapt the forecasting model to different sectors or objectives, such as sectors beyond delivery, certain input files and parameters need to be modified. Below is an improved explanation with specific details about the files used in your model.

POI Data File: /content/Geohash7_Curitiba.csv

- **Structure:** This file contains three columns: ID, GEOHASH, and NUMPOINTS.
 - **ID:** A unique identifier for each geographical segment.
 - **GEOHASH:** A geospatial encoding system that divides the geographical area into small polygons (GeoHash 7), enabling precise demand analysis.

- NUMPOINTS: The number of Points of Interest (POIs) in each GeoHash polygon.
- Changes to make:
 - Modify the types of POIs relevant to the new sector. For example, in the delivery sector, restaurants and grocery stores may be the most important POIs, whereas, in the transportation sector, gas stations and parking lots may play a more significant role.
 - Add or filter POIs based on relevance to the sector. For example, if you're moving to retail, you might focus on malls, shopping centers, or retail outlets as the primary POIs.
 - Consider recalculating or filtering NUMPOINTS to include only POIs that influence the new sector.

Sales Data File: /content/Curitiba_radius_2058_vendas.csv

- Structure: This file includes three columns: geohash, distance, and Vendas_Mensais.
 - geohash: GeoHash 7 identifier, corresponding to the geographical segment.
 - distance: The distance in meters from the central point of analysis (e.g., distribution center or key location) to the GeoHash location.
 - Vendas_Mensais: Monthly sales data for each geographical segment.
- Changes to make:
 - Adjust the sales data to reflect the demand pattern of the new sector. For instance, in the transportation sector, you might replace Vendas_Mensais with a metric like number of passengers or number of vehicle entries.
 - You may also need to adjust the interpretation of the distance variable. In sectors like retail, distance could measure proximity to major shopping districts, whereas in healthcare, it might represent proximity to healthcare centers.

- If the sector uses a different type of demand metric (e.g., patient visits instead of product sales), the `Vendas_Mensais` column will need to be replaced or recalibrated with data relevant to that metric.

A Guide for Using the Forecasting Model

Using the forecasting model requires the collection of geospatial data on POIs and historical sales. The process begins with extracting data from remote sensing and collaborative sources such as OpenStreetMap. Then, the data is indexed by GeoHash 7, which subdivides the space into polygons, facilitating a granular analysis of regional demand. The model uses this database to apply machine learning algorithms that project future sales based on POI density and other contextual variables. The results can be visualized through graphs that display accumulated projections over distances.

To ensure the model's accuracy, it is essential to evaluate multiple performance metrics across different machine learning algorithms, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-squared (R^2). Each metric offers unique insights into the model's predictive quality:

- **MSE** and **RMSE** penalize larger errors by squaring the differences, making them sensitive to outliers and useful for understanding the impact of significant deviations.
- **MAE** represents the average error between predicted and actual values, providing a straightforward measure of prediction accuracy.
- **MAPE** allows for interpreting the error as a percentage, making it easy to understand the model's performance across different scales.
- **R^2** indicates the proportion of variance explained by the model, with values closer to 1 representing a better fit.

Using these metrics, we can compare models to select the one that performs best in minimizing forecasting errors, thus providing reliable demand projections.

The following table shows the performance of different models:

- **Random Forest:** MSE = 1.093204, RMSE = 1.045564, MAE = 0.192706, MAPE = 0.098934%, $R^2 = 0.998533$
- **Linear Regression:** MSE = 769.211017, RMSE = 27.734654, MAE = 22.475154, MAPE = 11.704224%, $R^2 = -0.032133$
- **Neural Network:** MSE = 772.857613, RMSE = 27.800317, MAE = 22.383712, MAPE = 11.661192%, $R^2 = -0.037027$
- **Gradient Boosting:** MSE = 2.847339, RMSE = 1.687406, MAE = 0.798590, MAPE = 0.394899%, $R^2 = 0.996179$
- **Random Forest (Optimized):** MSE = 1.093204, RMSE = 1.045564, MAE = 0.192706, MAPE = 0.098934%, $R^2 = 0.998533$
- **Gradient Boosting (Optimized):** MSE = 2.209243, RMSE = 1.486352, MAE = 0.161598, MAPE = 0.082165%, $R^2 = 0.997036$
- **Neural Network (Optimized):** MSE = 756.919886, RMSE = 27.512177, MAE = 22.511841, MAPE = 11.752527%, $R^2 = -0.015641$

Regularly evaluating these metrics across different algorithms allows us to select the best-performing model for each specific scenario. In this case, the optimized **Random Forest** model shows the best performance overall, with very low MSE, RMSE, MAE, and MAPE, and an R^2 close to 1, indicating a strong fit to the data. However, it is also important to consider the **processing cost** of each model, as more complex algorithms, such as Neural Networks and Gradient Boosting, may require significantly more computational resources and time, which can impact scalability and feasibility in real-world applications. Balancing model accuracy with processing efficiency ensures that the chosen model not only delivers reliable forecasts but also operates within practical constraints, supporting informed and sustainable decision-making across diverse sectors and regions.

A Tool to Assist in the Visualization of Demand Forecasts

A tool has been developed to facilitate the visualization of demand forecasts, allowing the projection results to be displayed in graphs. The forecasts can be visualized through accumulated curves, showing the potential for sales growth as the area of analysis expands. This tool supports strategic planning by showing how projected sales vary depending on the distance from central points

(e.g., distribution centers or commercial areas). Additionally, the tool can be used to compare actual sales with projections, providing insights into expected performance in unexplored regions.

The table displays the Accumulated Sales values predicted by different machine learning models. Each model is represented with its optimized or initial version:

- **Neural Network (Optimized)** has the highest accumulated sales prediction at **630,863.82**.
- **Linear Regression** follows with **628,421.14**.
- **Random Forest (Optimized)**, **Gradient Boosting (Optimized)**, and **Random Forest (Initial)** have similar accumulated values around **627,500**.
- **Gradient Boosting (Initial)** shows **627,501.40**, and **Neural Network (Initial)** has **627,241.77**.

These values indicate that the optimized Neural Network model predicts slightly higher accumulated sales, while other models, both optimized and initial, show similar, consistent values around 627,500.

REFERENCES

- Chen, J., Ban, Y., & Li, S. (2020). Satellite remote sensing for urban growth monitoring: A review of methods and applications. *International Journal of Remote Sensing*, 35(24), 8296-8342.
- Wang, L., Li, Y., & Huang, B. (2019). Demand prediction in the energy sector using artificial intelligence and remote sensing data. *Energy Policy*, 128, 853-862.
- Zhang, L., Zhang, L., & Du, B. (2018). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 6(2), 27-45.
- Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 166-177.

MÉTODOS PARA TRATAR MISSING DATA NA GERAÇÃO DE ENERGIA SOLAR: ESTIMANDO DESEMPENHO EM CENÁRIOS DE PERDAS DE DADOS

Glauco Lauria Marques Filho

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Paulo André Lima de Castro

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

Data quality is crucial for accurate analysis of solar energy systems' performance and, consequently, for decision-making. However, failures in meteorological stations, such as communication issues, often result in missing data, which can compromise performance analyses. This study explores and compares different methods for handling missing data in solar plant datasets, focusing on imputation techniques for critical variables such as POA irradiance and module temperature, and subsequently calculating PR with these generated values to assess the impact of these techniques in simulated data loss scenarios. The results demonstrate the effectiveness of certain approaches in maintaining the accuracy of analyses.

RESUMO

A qualidade dos dados é fundamental para a análise precisa do desempenho de sistemas de energia solar e, por consequência, na tomada de decisão. No entanto, falhas em estações meteorológicas, como problemas de comunicação, frequentemente resultam em dados ausentes, o que pode comprometer as análises de desempenho. Este estudo explora e compara diferentes métodos para lidar com missing data em conjuntos de dados de plantas solares, focando em técnicas de imputação para variáveis críticas como irradiância POA e temperatura dos módulos, e posteriormente calculando o PR com esses valores gerados, de forma a avaliar o impacto dessas técnicas em cenários simulados de perda de dados. Os resultados demonstram boa eficácia de certas abordagens em manter a precisão das análises.

INTRODUÇÃO

Nas últimas duas décadas, segundo a [EPE 2024], observou-se um aumento de 60,25% no consumo de energia elétrica no Brasil, devido a diversos fatores como crescimento econômico, aumento da população e mudanças climáticas que resultaram em verões mais quentes, elevando a demanda por sistemas de resfriamento. Concomitantemente, houve um aumento na preocupação com a sustentabilidade, fazendo com que fontes renováveis se tornassem cada vez mais importantes.

Com o crescente número de instalações fotovoltaicas, surgem preocupações relacionadas ao retorno do investimento e à capacidade de geração, que dependem de fatores como eficiência, manutenção e limpeza dos painéis, perdas em equipamentos do sistema e correta operação da usina. Uma forma de avaliar a geração é por meio do cálculo de indicadores de desempenho, os quais permitem às geradoras verificarem se seus ativos estão proporcionando o máximo retorno possível. No contexto da geração fotovoltaica, uma das principais métricas utilizadas é o Performance Ratio (PR), que mensura o desempenho de plantas solares [Khalid *et al.* 2016]. O PR tornou-se um indicador de extrema importância, sendo frequentemente utilizado como referência em contratos entre empresas envolvidas em projetos solares.

O cálculo desse indicador envolve diversas variáveis e, de acordo com a norma IEC61724-1 [IEC 2021], é necessário medir a potência da planta, a temperatura dos módulos e a irradiância solar, entre outras medidas, dependendo do tipo de PR escolhido como referência. As duas últimas variáveis, temperatura do módulo e irradiância, são obtidas por meio de Estações Meteorológicas. No entanto, em muitas plantas solares no país, diversos profissionais enfrentam inúmeros problemas relacionados aos dados dessas torres meteorológicas. Devido ao fato de as Estações Meteorológicas serem erroneamente consideradas equipamentos não críticos para a operação, elas frequentemente são negligenciadas pelas equipes de campo. Como resultado, essas torres podem passar vários dias sem funcionar adequadamente até que seja notado por alguma equipe, ocasionando a perda do histórico de dados e, conseqüentemente, prejudicando a realização de cálculos de indicadores de desempenho.

Existem diversas causas para as interrupções nas Estações Meteorológicas, como problemas físicos nos cabos, desligamentos indevidos quando a usina é desativada e não possui um sistema auxiliar, falhas nos dispositivos de armazenamento de dados, células sujas e problemas nos protocolos de comunicação [Mohamad *et al.* 2021]. Esses obstáculos dificultam a obtenção de informações precisas e confiáveis, prejudicando a avaliação adequada do desempenho das plantas solares.

Contextualização

Funcionamento de uma Planta de Geração Solar

As plantas de geração de energia solar operam convertendo a luz do sol em eletricidade por meio de painéis solares fotovoltaicos. Esses painéis são compostos por células solares feitas de materiais semicondutores, como o silício, que são dopados com substâncias como o boro e fósforo. Quando expostas à irradiância, essas células geram uma corrente elétrica contínua (CC) [Castro 2002].

Diversos painéis fotovoltaicos ligados em paralelo injetam a corrente CC em inversores, que têm a função de transformar a corrente para alternada (CA). É importante frisar que esse processo de conversão implica em perda de energia, já que não se trata de um processo 100% eficiente [Machado and Miranda 2015].

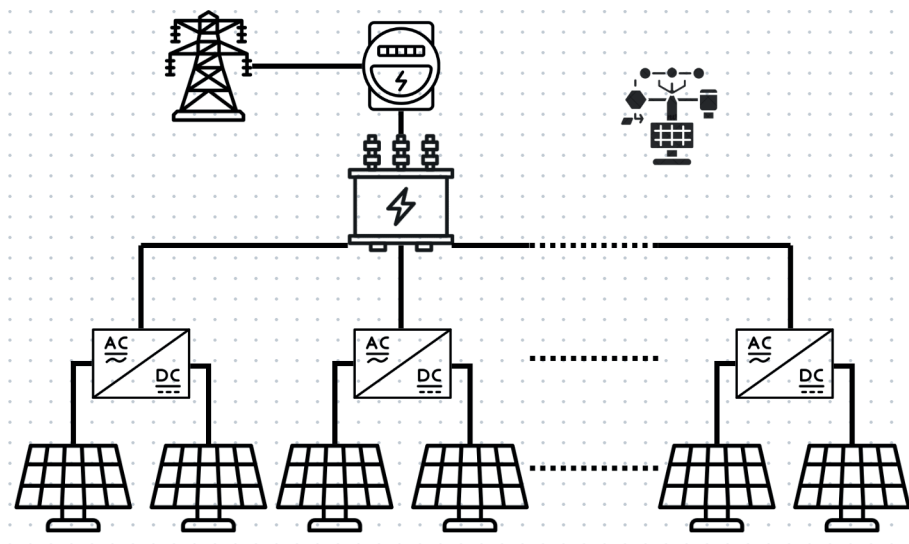
Novamente, inversores em paralelo se ligam a uma subestação que elevará os valores de tensão da classificação baixa tensão (BT) para média tensão (MT) através de transformadores, também implicando em perda de energia. Por fim, a energia será inserida no sistema elétrico brasileiro, não sem antes passar por medidores de faturamento para registrar quanto a determinada planta gerou [Machado e Miranda 2015].

As plantas solares também contam com um equipamento fundamental para o monitoramento de desempenho: a estação meteorológica. Essa estação fornece dados meteorológicos essenciais, como umidade, velocidade do vento, temperatura ambiente, além da temperatura dos módulos (MOD) e a irradiância no plano dos módulos (POA).

A temperatura dos módulos (MOD) e a irradiância no plano dos módulos (POA) são as duas medidas mais importantes para cálculos de indicadores,

uma vez que a MOD afeta diretamente a eficiência energética da conversão de irradiância solar em corrente; com temperaturas mais altas, tem-se menor eficiência. A POA é a própria intensidade da luz do sol diretamente no plano do módulo; quanto maior, mais energia é gerada.

Figura 1 - Esquemático de uma planta solar.



Cálculo de Performance Ratio (PR)

O Performance Ratio (PR) é um importante indicador de eficiência de uma usina fotovoltaica, independentemente de sua orientação e localização, sendo também utilizado como um fator de qualidade da planta. Expressado em porcentagem, o PR descreve a proporção entre a entrega real de energia e a geração teórica de uma usina fotovoltaica (FV), sendo uma indicação da energia que está realmente disponível para a rede após descontadas as perdas.

A eficiência da usina solar é diretamente proporcional ao seu Percentual de Rendimento (PR), que deve se aproximar o máximo possível de 100%. Por essa razão, monitorar regularmente esse indicador é crucial para o acompanhamento eficaz da usina. A verificação periódica do PR permite aos operadores avaliar a performance e a eficácia da operação, identificando desvios, perdas e falhas que podem afetar significativamente a rentabilidade do empreendimento [Solar 2022].

A norma IEC61724-1 [IEC 2021] descreve a metodologia de cálculo de alguns tipos de PR, sendo o mais usual o PR corrigido pela temperatura. Nesse cálculo, são considerados os aspectos físicos de perda de eficiência energética devido ao aquecimento das células fotovoltaicas.

O cálculo do PR corrigido pela temperatura pode ser feito por: [IEC 2021]

$$PR'_{STC} = \frac{\sum_k P_{out,k} \times \tau_k}{\sum_k \frac{(1+\gamma \times (T_{mod,k} - 25^{\circ}C)) \times P_0 \times G_{i,k} \times \tau_k}{G_{i,ref}}}$$

Onde:

- PR'_{STC} - performance ratio corrigido pela temperatura em %;
- $P_{out,k}$ - potência medida gerada pela planta para cada observação no tempo k, em
- τ_k - tempo em horas entre observações k, mais comumente 15 minutos, ou seja
- γ - coeficiente de perdas elétricas em razão da temperatura, por exemplo
- $-0.0034^{\circ}C^{-1}$, dependendo do modelo dos painéis fotovoltaicos;
- $T_{mod,k}$ - temperatura medida dos painéis fotovoltaicos (MOD), para cada observação no tempo k, em $^{\circ}C$;
- P_0 - potência pico instalada na planta, por exemplo $2500kW/p$;
- $G_{i,k}$ - irradiância do plano inclinado (POA) medida, para cada observação no tempo k, em W/m^2 ;
- $G_{i,ref}$ - irradiância de referência, usualmente igual a $1000W/m^2$.

Tratamento de dados faltantes

Existem diversos métodos para tratar dados faltantes, desde os mais simples, como o descarte de dados faltantes, até métodos mais complexos, como algoritmos de imputação [Hackers 2023]. Os métodos mais relevantes para este artigo serão detalhados a seguir.

- **Remoção dos dados faltantes:** é o método mais simples e só deve

ser utilizado se os dados ausentes estiverem distribuídos de forma aleatória. Deve-se tomar cuidado para que o descarte de dados não acarrete em perda significativa de informação [Castro 2014];

- **Imputação pela média:** consiste em substituir os dados faltantes pela média da amostra de dados. Essa técnica não altera a média amostral, mas pode alterar outras características da distribuição.
- **Regressão:** consiste em substituir os valores faltantes por valores preditos a partir de um modelo de regressão que utiliza outras variáveis completas presentes na base de dados como covariáveis. São estimadas as equações de regressão que predizem as variáveis com dados faltantes, e então são gerados os valores preditos para os dados faltantes a partir das regressões.

Algoritmos de Regressão

Segundo [Maulud e Abdulazeez 2020], a regressão é uma técnica amplamente empregada para previsão e predição, áreas nas quais suas aplicações se sobrepõem significativamente com o aprendizado de máquina.

Nos modelos de regressão estima-se o valor da variável dependente "y" devido à variação dos valores da variável independente "x". A regressão ainda pode ser classificada em linear simples ou múltipla isso depende do número de variáveis independentes que explicam a variável dependente [Maulud e Abdulazeez 2020].

Existem vários algoritmos de regressão, cada um com suas características, vantagens e desvantagens. Nos próximos tópicos, vamos abordar os principais algoritmos utilizados neste projeto.

Problema de Pesquisa

Os dados, e principalmente os dados meteorológicos, desempenham um papel crítico na avaliação do desempenho das plantas solares, através do cálculo de indicadores como o Performance Ratio (PR). No entanto, as estações meteorológicas frequentemente enfrentam falhas que resultam em dados ausentes, prejudicando a análise e a tomada de decisões sobre a operação das usinas solares.

Existem diversas abordagens para garantir a qualidade desses dados, como a conscientização e a criação de uma cultura voltada para assegurar bons

dados pela equipe de operação, a adoção de equipamentos mais sofisticados, a implementação de redundância, entre outras. Porém, essas abordagens podem ser custosas e demoradas para serem efetivas.

Objetivo Geral

Visto que nem todos os empreendimentos estão com alta maturidade em disponibilidade e garantia de bons dados, o que fazer com os dados que já foram perdidos? O objetivo deste trabalho é justamente propor uma alternativa para mitigar a falta de perspectiva de desempenho de plantas solares em cenários onde os dados meteorológicos foram perdidos, dessa maneira ajudando tomada de decisão em cenário de perdas de dados nesse setor tão importante de qualquer país.

Objetivos Específicos

Este artigo tem como objetivos específicos:

- **Preparar os dados para análise:** Realizar a limpeza, transformação e pré-processamento dos dados coletados para garantir que estejam adequados para a aplicação dos modelos.
- **Desenvolver e testar modelos:** Implementar, treinar e testar diferentes modelos com o objetivo de identificar o modelo que oferece melhor desempenho para tratar dados faltantes no contexto meteorológico de plantas solares.
- **Analisar o desempenho dos modelos:** Avaliar o desempenho dos modelos desenvolvidos, utilizando métricas como RMSE (Root Mean Square Error) e R^2 , para identificar o modelo mais adequado para a predição das variáveis meteorológicas.

DESENVOLVIMENTO

A plataforma escolhida para o desenvolvimento deste trabalho foi o Python, devido à sua ampla difusão e à disponibilidade de diversas bibliotecas na comunidade para lidar com projetos de ciência de dados.

A seguir, serão descritas as etapas realizadas no projeto.

Extração de dados

Os dados utilizados nesta pesquisa foram obtidos através de uma empresa do setor elétrico brasileiro, onde o autor trabalha, o que proporcionou um entendimento prévio da base de dados.

A extração de dados foi realizada através da API do InfluxDB, um banco de dados de alto desempenho para lidar com séries temporais. A consulta requisitou os dados com agregação, de forma a gerar medições em intervalos regulares de 15 minutos, conforme a norma IEC 61724 [IEC 2021] para o cálculo de PR.

Os dados retornados formam uma tabela com três colunas: Timestamp, Value e Pathname, descritas a seguir:

- **Timestamp:** estampa de tempo das medidas, possuindo tipo de dados datetime;
- **Value:** valor da medida, representado como um número de ponto flutuante (float);
- **Pathname:** string que contextualiza a medida contida na linha da tabela, por exemplo: corrente planta, temperatura módulos.

Para melhor desenvolvimento do trabalho, a tabela foi pivotada para que cada Pathname se transformasse em uma coluna e o valor de cada coluna fosse o próprio valor de cada medida. Desse modo, obteve-se as seguintes colunas, ou seja, medidas:

Tabela 1 - Colunas/medidas disponíveis.

timestamp		
current a	current b	current c
park1.power	estm.poa.irad	estm.temp.mod

timestamp		
current a	current b	current c
park1.INV1.current a	park1.INV1.current b	park1.INV1.current c
park1.INV1.freq	park1.INV1.power	park1.INV1.temp
park1.INV2.current a	park1.INV2.current b	park1.INV2.current c
park1.INV2.freq	park1.INV2.power	park1.INV2.temp
park1.INV3.current a	park1.INV3.current b	park1.INV3.current c
park1.INV3.freq	park1.INV3.power	park1.INV3.temp
park1.INV4.current a	park1.INV4.current b	park1.INV4.current c
park1.INV4.freq	park1.INV4.power	park1.INV4.temp

Tabela 2 - Exemplo de dados após pivot.

timestamp	current a	current b	current c	estm.poa.irad
2023-01-01 02:30:00+00:00	0.0000	0.5468	0.0000	-1.435020
2023-01-01 02:45:00+00:00	0.0000	0.3906	0.3906	-1.408373
2023-01-01 03:00:00+00:00	0.4687	0.0000	0.0000	-1.427320
2023-01-01 03:15:00+00:00	0.5468	0.0000	0.0000	-1.440360
2023-01-01 03:30:00+00:00	0.0000	0.3906	0.0000	-1.441907
2023-01-01 03:45:00+00:00	0.0000	0.0000	0.0000	-1.489200
2023-01-01 04:00:00+00:00	0.0000	0.0000	0.0000	-1.514380
2023-01-01 04:15:00+00:00	0.3906	0.0000	0.3906	-1.474043
2023-01-01 04:30:00+00:00	0.4687	0.0000	0.0000	-1.469980
2023-01-01 04:45:00+00:00	0.4687	0.0000	0.0000	-1.504220
2023-01-01 05:00:00+00:00	0.5468	0.0000	0.0000	-1.546080

Remoção de dados indesejados

Com o objetivo de obter uma base sólida de dados, os dados nulos foram removidos. Embora o tema desta pesquisa seja lidar com dados faltantes, esses dados iniciais foram removidos para garantir uma qualidade de dados adequada para treinar os modelos. Quando for necessário testar os modelos com dados nulos, estes serão gerados pelo autor, a fim de possibilitar um melhor controle dos cenários. Com isso, 2.408 linhas foram removidas, equivalentes a 4,77% do total, restando 48.046 linhas.

Além disso, visto que em um modelo de usina solar não faz sentido manter dados noturnos, foi realizado um tratamento seguindo a norma IEC61724-1 [IEC 2021]. Assim, considerou-se que somente são válidos os períodos de geração com irradiância maior que $20W/m^2$. Essa condição permitiu a remoção de períodos inválidos e noturnos. Como resultado, 25.027 linhas foram removidas, equivalente

a 52,09% do total, restando 23.019 linhas. Esses valores de remoção são esperados, por volta de 50%, correspondendo justamente a períodos noturnos ou com pouca irradiância, como o amanhecer e o anoitecer.

Esta abordagem garantiu uma base de dados mais consistente e relevante para a análise subsequente e o desenvolvimento dos modelos de desempenho da usina fotovoltaica.

Cálculo de medidas derivadas

A coluna timestamp é única para cada observação, não sendo útil para os modelos de regressão utilizados nesse artigo, pois como será explicado posteriormente, não foram aplicados modelos voltados para séries temporais. Para contornar esse problema, foram criadas mais duas colunas: dia do ano e minutos desde o início do dia. Dessa forma, pode-se ter observações com valores semelhantes que poderão se correlacionar de alguma maneira.

Além disso, existem dados de corrente para cada uma das fases. Para simplificar os modelos, adotou-se o modelo trifásico a partir das correntes A, B e C.

Com isso, foram criadas mais cinco colunas: corrente trifásica da planta e corrente trifásica de cada um dos quatro inversores. Em seguida, as colunas de corrente de fases foram removidas, mantendo-se apenas as correntes trifásicas.

Dado que existem quatro medidas para cada um dos quatro inversores, foi possível simplificar essas 16 medidas adotando medidas totalizadoras de inversores, agrupando as quatro medidas de cada inversor. Isso não altera o objetivo de calcular o desempenho total da planta diante da falta de dados meteorológicos. As quatro novas medidas criadas para substituir as 16 são: corrente trifásica somada dos inversores, frequência média dos inversores, potência somada dos inversores e temperatura interna média dos inversores.

Com a criação dessas medidas para substituir as demais, restaram 11 medidas

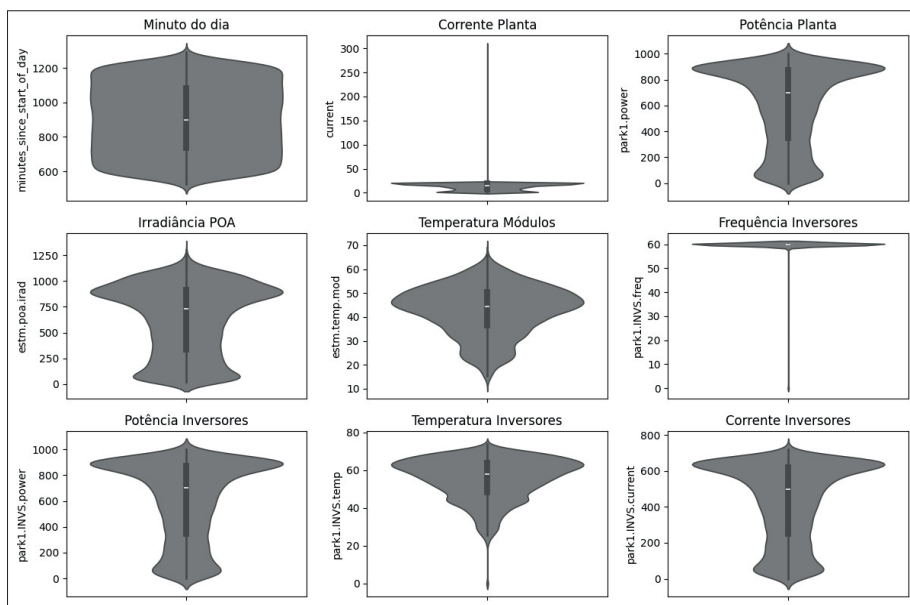
- time:timestamp único de referência, mas que não será utilizado nos modelos

- day of the year: dia do ano
- minutes since start of day: minuto desde o início do dia
- current: corrente total da planta
- park1.power: potência total da planta
- estm.poa.irrad: irradiância (POA) da planta
- estm.temp.mod: temperatura dos módulos (MOD) da planta
- park1.INVS.freq: frequência média dos inversores
- park1.INVS.power: potência total dos inversores
- park1.INVS.temp: temperatura interna média dos inversores
- park1.INVS.current: corrente total dos inversores

Análise Exploratória

Para visualizar o comportamento dos dados, foi montado um violin plot com as variáveis. O violin plot é uma ferramenta útil para entender a distribuição dos dados, pois combina aspectos de um box plot e um kernel density plot, mostrando não apenas a mediana e os quartis, mas também a densidade de probabilidade dos dados em diferentes valores [Hintze and Nelson 1998]. Neste caso, ele possui uma simetria no eixo X, pois não se trata de dados categóricos.

Figura 2 - Violin plot com outliers.

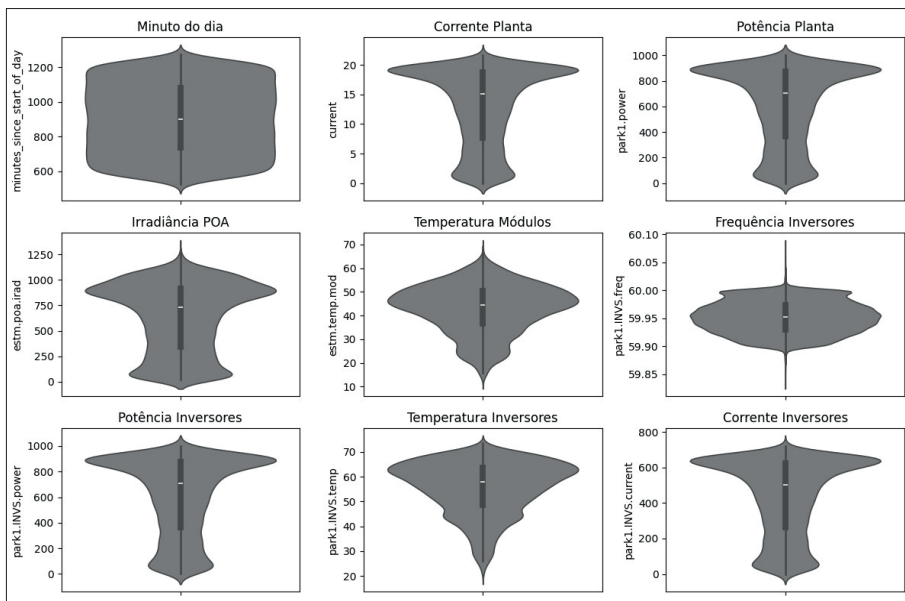


Visualmente, identificou-se a presença de outliers na temperatura dos módulos, na frequência e na temperatura dos inversores. Para remover esses outliers, foram aplicados alguns filtros específicos:

- **Corrente dos inversores:** Deveria ser menor que 50 Amperes, pois valores maiores seriam fisicamente impossíveis devido ao modelo dos equipamentos. Os outliers provavelmente ocorreram devido a ruídos na leitura, logo valores maiores que 50 amperes foram retirados da base de dados original;
- **Frequência dos inversores:** Filtrada em intervalos de 59.5 até 60.5 Hz, que é a faixa usual de frequência da corrente alternada. Valores fora dessa faixa foram retirados da base de dados original;
- **Temperatura interna dos inversores:** Filtrada para valores maiores que 10 graus Celsius. Este valor foi escolhido pois a planta está situada em uma região do Brasil onde os menores valores de temperatura medidos raramente chegam a 17 graus Celsius. Valores abaixo de 10 graus são provavelmente erros de leitura e também foram retirados da base de dados original.

Após a aplicação desses filtros, 413 linhas foram removidas, o que equivale a 1.79% das linhas, restando 22.606 linhas. Em seguida, foi realizado um novo violin plot para verificar a ausência de outliers nas medidas restantes.

Figura 3 - Gráfico de violino sem outliers.



Pode-se perceber uma diferença apenas nos gráficos de corrente dos inversores, frequência dos inversores e temperatura interna dos inversores, pois, nessas medidas, foram identificados os outliers e removidos.

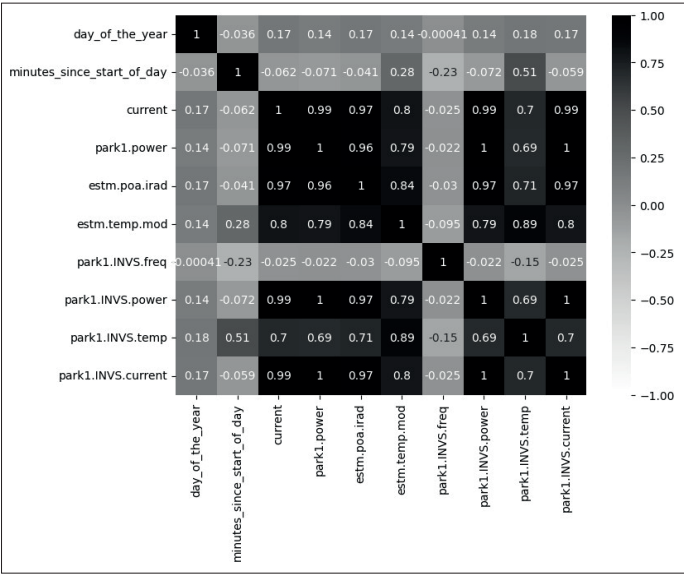
- A distribuição simétrica do gráfico de minuto do dia sugere uma uniformidade na distribuição dos dados ao longo do dia.
- Os gráficos de corrente da planta, potência da planta, potência dos inversores e corrente dos inversores mostram uma concentração majoritária em valores mais altos, com uma cauda pequena indicando alguns valores mais baixos.
- A irradiância apresenta uma distribuição mais variada, com picos em diferentes intervalos, refletindo a variação típica ao longo de um dia ensolarado.
- As distribuições da temperatura dos módulos e dos inversores são

relativamente simétricas, com a maioria dos valores concentrados em torno de um ponto central. Este padrão é coerente, considerando que a temperatura é uma grandeza física que varia mais lentamente em comparação com a irradiância.

- A temperatura dos inversores tende a ser mais elevada do que a dos módulos, devido ao funcionamento contínuo como equipamentos elétricos.
- A distribuição da frequência dos inversores é muito estreita, indicando que a frequência é bastante constante, com variações mínimas, como esperado.

Também foi criado um mapa de calor para visualizar a matriz de correlação entre as variáveis.

Figura 4 - Heatmap de Correlação.



O mapa de calor revela correlações significativas entre variáveis relacionadas à corrente, potência e irradiância, conforme esperado para uma planta solar. A frequência dos inversores e o dia do ano apresentam correlações muito baixas com as demais variáveis.

As temperaturas dos módulos e dos inversores também exibem correlações moderadas a altas com algumas variáveis, sugerindo relações com a

irradiância. É intuitivo que em condições de maior incidência solar, as temperaturas aumentem, afetando tanto a eficiência dos módulos quanto o aquecimento dos inversores.

Por meio da análise exploratória, aliada à remoção de dados indesejados e outliers, foi possível identificar um cenário típico de operação de uma usina solar. Dessa forma, os dados estão preparados para serem utilizados em modelos analíticos.

Preparação final para inserção em modelos

Para simular as falhas na estação meteorológica assim como melhor aproveitamento dos modelos, será preciso preparar os dados de forma adequada.

Foram selecionados os dias que têm pelo menos 30 registros, ou seja 30 observações de 15 minutos, totalizando 7 horas e 30 minutos de dados válidos durante o período de geração. Também foram divididos esses dias em dois grupos: um para treino (80% dos dias) e outro para teste (20% dos dias). Os modelos deverão prever valores para o dia todo, simulando cenários em que a estação meteorológica ficou indisponível todo o dia.

Também foram aplicados **MinMaxScaler** nas features e targets, isso facilitará o treinamento dos modelos facilitando assim o treinamento dos modelos.

Método de tratamento com Imputação pela média

Conforme comentado no item 1.1.3, uma das alternativas para lidar com dados ausentes é substituir pela média das amostras de dados. Para trazer uma melhor aproximação do cenário real de variação das variáveis-alvo, criou-se uma relação de média das features em relação ao minuto do dia, e, a partir disso, os dados dos 20% dos dias com dados faltantes foram substituídos pelas médias minuto a minuto dos outros 80%.

Os gráficos abaixo mostram o comportamento das curvas reais da POA e MOD em relação aos valores previstos com esse método.

Figura 5 - Imputação pela Média - Valores Reais vs Previstos (POA).

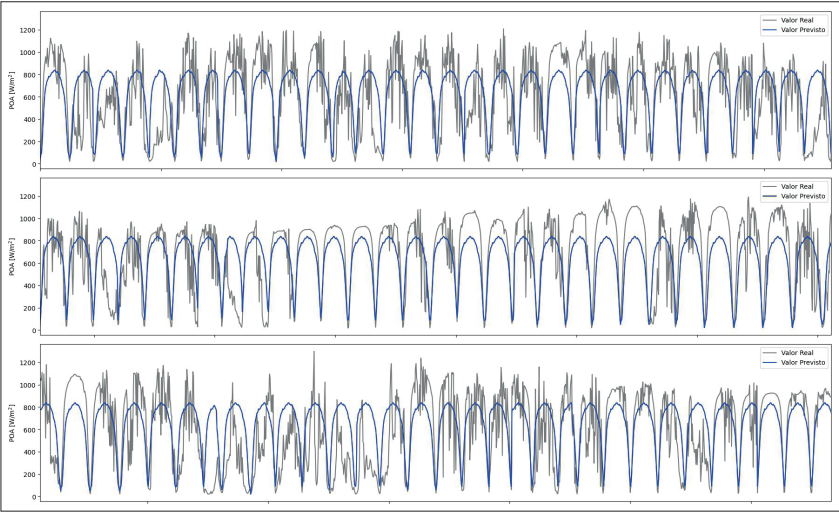
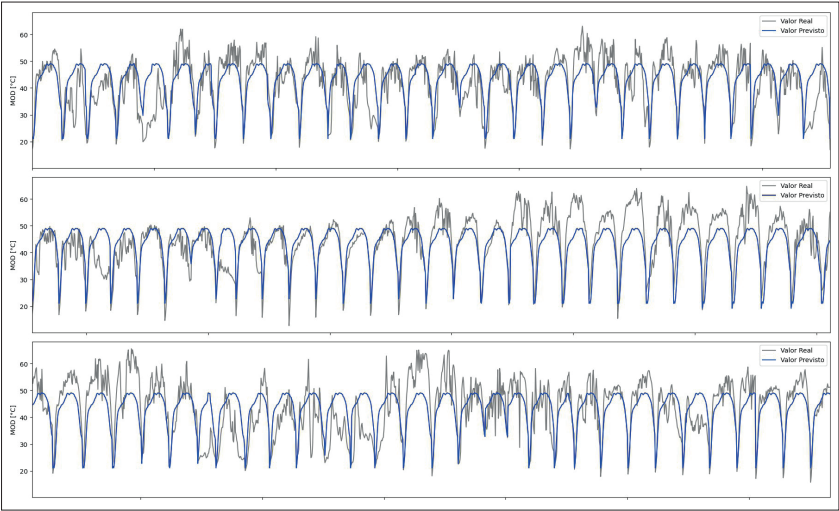


Figura 6 - Imputação pela Média - Valores Reais vs Previstos (MOD).



Pode-se perceber que as curvas de média obtidas repetem a tendência de curva de sino para os dias, tanto para POA quanto para MOD. No entanto, por ser um tratamento muito simples, ele não se encaixa em dias que possuem valores oscilantes, seja por causa da incidência de nuvens ou chuvas.

A tabela a seguir contém os indicadores da eficiência da imputação, sendo seu R^2 e suas raízes quadradas dos erros-médios (RMSE) em W/m^2 para POA e em $^{\circ}C$ para MOD:

Tabela 3 - Imputação pela Média.

Medida	Algoritmo	R^2	RMSE
POA	Mean	0.4523	245.3731
MOD	Mean	0.5260	6.9277

Também foi possível perceber quantitativamente a baixa eficiência do método, pois foram obtidos baixos valores de R^2 e altos de RMSE.

Com os dados previstos, foi possível calcular os PRs para os dias selecionados. O resultado desses cálculos de PR será apresentado no item 3.

Método de tratamento com Regressão

Utilizando a biblioteca em Python scikit-learn, foram testados diversos algoritmos de regressão para observar quais se adequavam melhor a imputação de dados faltantes. Os algoritmos testados foram: KNeighborsRegressor, DecisionTreeRegressor, MLPRegressor, SVR e NuSVR. Além de testar diferentes algoritmos, foi utilizado o GridSearchCV para testar diferentes hiperparâmetros dos algoritmos. A métrica utilizada para definir os melhores modelos, tanto para POA quanto para MOD, foi o R^2 .

Não foram utilizados métodos de regressão voltados para séries temporais, pois a característica do problema segue conceitos físicos previamente conhecidos. Por exemplo, para ter uma geração alta, será preciso bastante irradiância. Logo, cada intervalo de 15 minutos atuará como uma observação individualizada de variáveis independentes que causam efeitos nas variáveis dependentes, sem lidar com questões temporais.

A tabela a seguir contém um resumo dos melhores modelos encontrados para cada algoritmo, seus R^2 e RMSE em W/m^2 para POA e em $^{\circ}C$ para MOD:

Tabela 4 - Modelos de Regressão.

Medida	Algoritmo	Hiperparâmetros	R ²	RMSE
POA	KNeighborsRegressor	n neighbors: 10, weights: distance	0.9725	54.9706
POA	DecisionTreeRegressor	max depth: 8, min samples split: 5	0.9648	62.2523
POA	SVR	C: 0.1, kernel: rbf	0.9489	74.9523
POA	NuSVR	nu: 0.7, kernel: rbf	0.9767	50.5951
POA	MLPRegressor	activation: relu, solver: adam	0.9717	55.7494
MOD	KNeighborsRegressor	n neighbors: 10, weights: distance	0.9182	2.8770
MOD	DecisionTreeRegressor	max depth: 9, min samples split: 9	0.9127	2.9770
MOD	SVR	C: 1, kernel: rbf	0.9058	3.0875
MOD	NuSVR	nu: 0.7, kernel: rbf	0.9304	2.6544
MOD	MLPRegressor	activation: relu, solver: adam	0.9185	2.8722

Em termos de R^2 , o melhor algoritmo foi o NuSVR para ambos os casos. Os gráficos a seguir mostram os resultados das previsões do NuSVR para POA e MOD, respectivamente.

Figura 8 - Regressão com NuSVR para MOD.

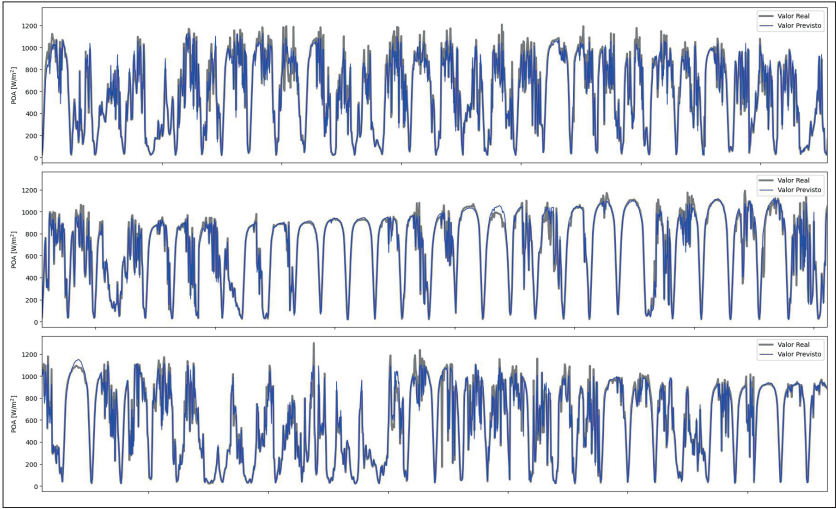
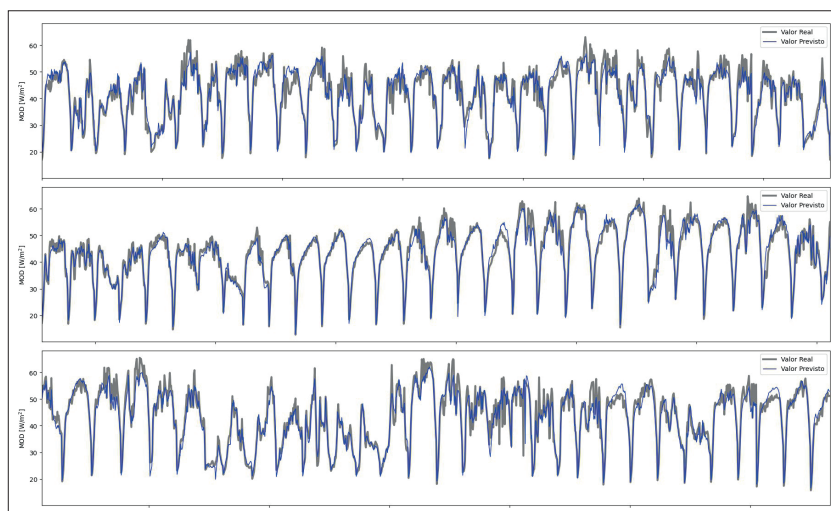


Figura 7 - Regressão com NuSVR para POA.



O desempenho superior em relação ao método de imputação pela média é nitidamente observado tanto visualmente com o ajuste das curvas previstas e reais estando bem próximos, quanto numericamente com o seu maior R^2 e menor RMSE. Os modelos encontrados foram combinados de modo a usar um para POA e outro para MOD, e assim o PR para cada dia de interesse foi calculado. Os resultados dos PRs encontrados poderão ser consultados no item 3.

RESULTADOS

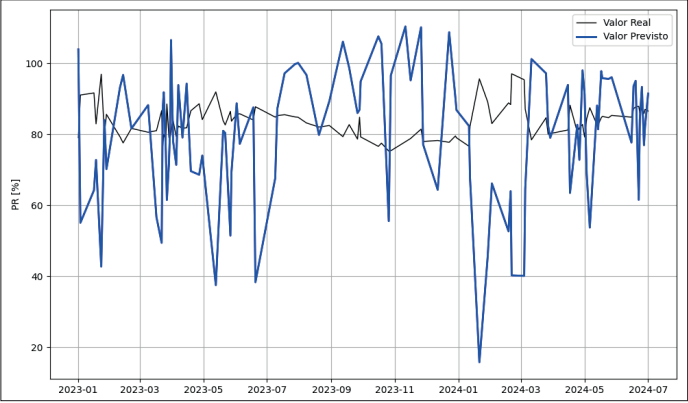
Utilizando a formula do item 1.1.2 foram calculados PR utilizando os diversos modelos citados nos itens anteriores. A tabela abaixo demonstra o modelo utilizado em cada variável, junto com seu R^2 e RMSE em %:

Tabela 5 - Resultado do PR nos Modelos.

POA	MOD	R^2	RMSE
Mean	Mean	-22.6431	22.20
KNeighborsRegressor	KNeighborsRegressor	0.6298	2.7777
KNeighborsRegressor	SVR	0.6294	2.7792
NuSVR	MLPRegressor	0.6288	2.7815
KNeighborsRegressor	DecisionTreeRegressor	0.6285	2.7838

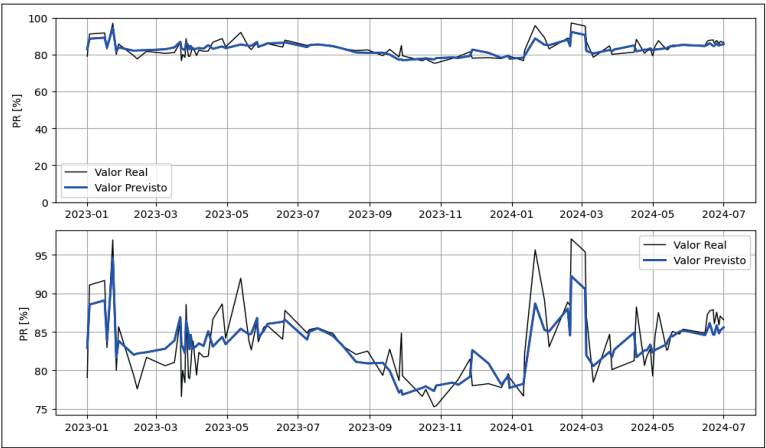
A figura abaixo fornece uma visualização de como estão os valores previstos e reais de PR utilizando o método de imputação pela média:

Figura 9 - PR calculado com Imputação pela Média.



A figura abaixo fornece uma visualização dos valores previstos e reais de PR utilizando o método de Regressão para POA e MOD com o melhor RMSE, em uma visualização com faixas nominais de PR e outra com aproximação da curva no eixo visual.

Figura 10 - PR calculado com Regressão.



Através do MSRE da tabela 5, já tinha ficado claro que o PR, ao imputar valores pela média, havia ficado bem ruim e impreciso, porém a figura 9 deixa

isso ainda mais evidente, mostrando que esse método não é muito adequado para resolver o problema de dados faltantes. O R^2 negativo confirma isso pois ele significa que o modelo está pior do que uma simples média dos valores de PR.

Já para o método utilizando regressão, tanto os resultados numéricos de MSRE quanto os visuais dos gráficos da Figura 10 superam as expectativas do autor, visto que o PR foi calculado para um dia sem nenhum dado meteorológico e foi obtido um MSRE de apenas 2,78%, mesmo o R^2 não estando tão alto, o pequeno erro médio compensa isso e torna o modelo útil em cenários de perdas de dados.

CONCLUSÃO

A perda de dados meteorológicos pode impactar negativamente a análise de desempenho de plantas solares, especialmente no cálculo do Performance Ratio (PR), uma vez que não será possível determinar se o empreendimento está gerando o máximo que deveria e identificar problemas nos equipamentos.

Este trabalho explorou alguns métodos para lidar com dados faltantes, como a imputação pela média e o uso de modelos de regressão. Os resultados indicam que modelos de regressão, como KNeighborsRegressor e DecisionTreeRegressor, são eficazes para preencher lacunas de dados, pois produziram previsões com os menores erros nos cenários avaliados. Esses modelos podem permitir estimativas de desempenho das plantas solares com menores erros, mesmo em cenários de perdas de dados.

Não obstante os resultados obtidos, é interessante buscar métodos mais precisos de previsão em condições reais. Nesse sentido, sugerimos como possíveis linhas de trabalho:

- Testar o uso de métodos mais avançados de imputação, tais como aqueles usados em séries temporais. Em especial, técnicas com arquiteturas recentes de redes neurais, como N-BEATS e N-HITS.
- Tratar dados faltantes captados por outros equipamentos críticos de uma planta solar, como os medidores de faturamento e inversores.

REFERÊNCIAS

- Castro, I. Q. (2014). Uma aplicação de métodos de imputação no estudo de fatores associados ao baixo peso ao nascer.
- Castro, R. M. (2002). Introdução à energia fotovoltaica. *DEEC/Seção de Energia Lisboa: Universidade Técnica de Lisboa. Instituto Superior Técnico*.
- EPE (2024). Consumo de energia elétrica. <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/consumo-de-energia-eletrica>. Acessado em julho de 2024.
- Hackers, D. (2023). Feature engineering: Técnicas para lidar com dados faltantes em um projeto de ciência de dados.
- Hintze, J. L. and Nelson, R. D. (1998). Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184.
- IEC (2021). Photovoltaic system performance - part 1: Monitoring. Standard IEC 61724- 1:2021, International Electrotechnical Commission, Geneva, CH.
- Khalid, A. M., Mitra, I., Warmuth, W., and Schacht, V. (2016). Performance ratio – crucial parameter for grid connected pv plants. *Renewable and Sustainable Energy Reviews*, 65:1139–1158.
- Machado, C. T. and Miranda, F. S. (2015). Energia solar fotovoltaica: uma breve revisão. *Revista virtual de química*, 7(1):126–143.
- Maulud, D. and Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1(2):140–147.
- Mohamad, N. B., Lim, B.-H., and Lai, A.-C. (2021). Imputation of missing values for solar irradiance data under different weathers using univariate methods. *IOP Conference Series: Earth and Environmental Science*, 721(1):012004.
- Solar, C. (2022). Performance ratio (pr) de usinas fotovoltaicas. <https://canalsolar.com.br/performance-ratio-pr-de-usinas-fotovoltaicas/>. Acessado em julho de 2024.

É POSSÍVEL ESTIMAR O RISCO DE ATRASOS OU PARALISAÇÕES DE OBRAS PÚBLICAS NO BRASIL COM O USO DE INTELIGÊNCIA ARTIFICIAL?

Diego de Lira Andrade

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Filipe Alves Neto Verri

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This study investigates the application of data science techniques to predict the risk of public works stoppage in Brazil, an area that has been little explored in the existing literature. Using data from transfer agreements of the Federal Government's General Budget, as well as indicators from the CAIXA SUSTAINABLE MANAGEMENT SEAL, predictive models were built with and without historical evolution to assess the risk of delays in public works. Models such as Gradient Boosting and Random Forest were optimized using Grid Search, with Gradient Boosting showing the best overall performance. The analysis suggests that the inclusion of historical information significantly improves the predictive capacity of the models, highlighting the importance of using past data to predict future risks. Even the model without historical data achieves good predictive performance, which expands its use even for new contracts. The results of this study contribute to better public works management, providing objective support for decision-making and efficient allocation of resources in public projects.

RESUMO

Este estudo investiga a aplicação de técnicas de ciência de dados para prever o risco de paralisação de obras públicas no Brasil, uma área pouco explorada na literatura existente. Utilizando dados de operações de repasse do Orçamento Geral da União, bem como indicadores do SELO CAIXA GESTÃO SUSTENTÁVEL, foram construídos modelos preditivos com e sem histórico de evolução para avaliar o risco de atrasos em obras públicas. Modelos como *Gradient Boosting* e *Random Forest* foram otimizados usando busca em grade, sendo que o *Gradient Boosting* apresentou o melhor desempenho geral. A análise sugere que a inclusão de informações históricas melhora significativamente a capacidade preditiva dos modelos, destacando a importância do uso de dados passados para prever riscos futuros e que mesmo o modelo sem histórico alcança bom desempenho preditivo, o que amplia seu uso mesmo para contratos novos. Os resultados deste estudo contribuem para a melhor gestão de obras públicas, fornecendo subsídios objetivos para a tomada de decisões e a alocação eficiente de recursos em obras públicas.

INTRODUÇÃO

O desenvolvimento da infraestrutura está diretamente associado ao progresso social e ao crescimento econômico. Desde a crise da dívida dos anos 1980, porém, os investimentos públicos em infraestrutura no Brasil diminuíram significativamente [1]. De acordo com o Banco Mundial, o Brasil enfrenta um déficit de financiamento em infraestrutura de aproximadamente US\$ 800 bilhões, o que representa 3,7% do PIB por ano até 2030 [1].

Diversos fatores contribuem para o baixo desempenho dos investimentos públicos, sendo o atraso e a paralisação de obras realidades recorrentes. O Tribunal de Contas da União (TCU) identificou, em seu Acórdão n. 1079/2019, que cerca de 14.000 empreendimentos estavam paralisados [2]. Motivada pelo referido documento, a CAIXA, atuando como Agente Financeiro e Mandatário no repasse de recursos da União, revisou sua base de contratos e constatou que 6.300 obras estavam paralisadas em um universo de 33.000 operações [3].

A definição vigente para obra paralisada envolve contratos de repasse que, após iniciados, não apresentam boletim de medição por um período igual ou superior a 90 dias [3]. Entre as mais de 20 causas identificadas para essas paralisações, a ausência de planejamento se destaca como uma das principais [3].

Desde o Acórdão do TCU, várias iniciativas foram lançadas pelo governo federal para mitigar os riscos de atrasos e paralisações e, em alguns casos, retomar obras suspensas. Nesse cenário, compreender as características dos contratos associados à paralisação pode ajudar na melhor aplicação dos recursos públicos.

Estudou-se uma base de contratos de repasse ativos, totalizando 22 mil operações e cerca de 96 bilhões de reais em investimentos. Segundo o Relatório Integrado da CAIXA de 2023, em dezembro daquele ano, 23.295 operações estavam em carteira, sendo que 2.616 delas estavam paralisadas, o que representa uma redução de 30,92% em comparação a dezembro de 2022 [4, p. 60].

Os contratos de repasse seguem um processo padronizado com quatro fases principais: proposição, celebração, execução e prestação de contas. Embora este trabalho não aprofunde essas etapas, as características presentes na base de dados permitem correlacionar datas às fases mais relevantes do processo [6].

Além dos dados das operações, foram utilizados no estudo indicadores do SELO CAIXA GESTÃO SUSTENTÁVEL (SELO), um reconhecimento dado a

municípios que adotam boas práticas de governança e responsabilidade socioambiental (ESG) na gestão pública local [5].

O foco deste estudo está na aplicação de técnicas específicas para a análise de dados desses contratos, buscando entender melhor como suas características se relacionam ao risco de paralisação. Dada a escassez de estudos sobre monitoramento e controle no gerenciamento de projetos públicos, conforme apontado por Gusmão *et al.* (2023) [7], é essencial que mais pesquisas sejam realizadas para melhorar a eficácia das práticas no setor público.

Projetos de investimento público enfrentam incertezas consideráveis, tornando a avaliação de risco uma ferramenta vital para seu sucesso. A capacidade de quantificar esses riscos é fundamental para garantir a eficiência do gerenciamento de projetos [8].

Os benefícios efetivos de uma política pública só podem ser alcançados com a implementação completa e funcional dos projetos. Portanto, mecanismos robustos para avaliar os riscos de atrasos ou paralisações são cruciais para uma gestão eficiente, apoiando os gestores públicos na tomada de decisões que garantam o cumprimento dos objetivos das políticas públicas.

Questão de Pesquisa

A questão abordada nesta pesquisa está correlacionada com o fato de que, historicamente no Brasil, o Poder Público realiza investimentos em obras públicas e estudos recentes identificam eventos como atrasos e paralisações neste tipo de empreendimento, os quais trazem impacto financeiro aos projetos. Considerando que modelos de inteligência artificial podem ajudar a estimar o risco de ocorrência de atrasos ou paralisações, a seguinte questão de pesquisa foi enunciada: “É possível estimar o risco de atrasos ou paralisações em obras públicas com o uso de Inteligência Artificial?”.

Objetivo Geral

O objetivo geral desta pesquisa é propor um modelo capaz de estimar o risco de atrasos ou paralisações de obras públicas, utilizando-se de algoritmos de classificação com base em dados públicos, visando ao fornecimento de subsídios

objetivos para prevenção e mitigação de riscos e melhor gestão de carteiras de investimentos de recursos públicos.

Objetivos Específicos

Os objetivos específicos desta pesquisa são:

- Definir bases de dados que melhor se aplicam ao objetivo geral, considerando não haver repositório único;
- Definir variável-alvo capaz de capturar os riscos estudados;
- Especificar e conceber um modelo capaz de estimar o risco de atrasos ou paralisações em obras públicas;
- Identificar a importância das características utilizadas no modelo, especialmente as características relacionadas ao histórico das operações e aos indicadores do SELO CAIXA GESTÃO SUSTENTÁVEL.

Contribuições

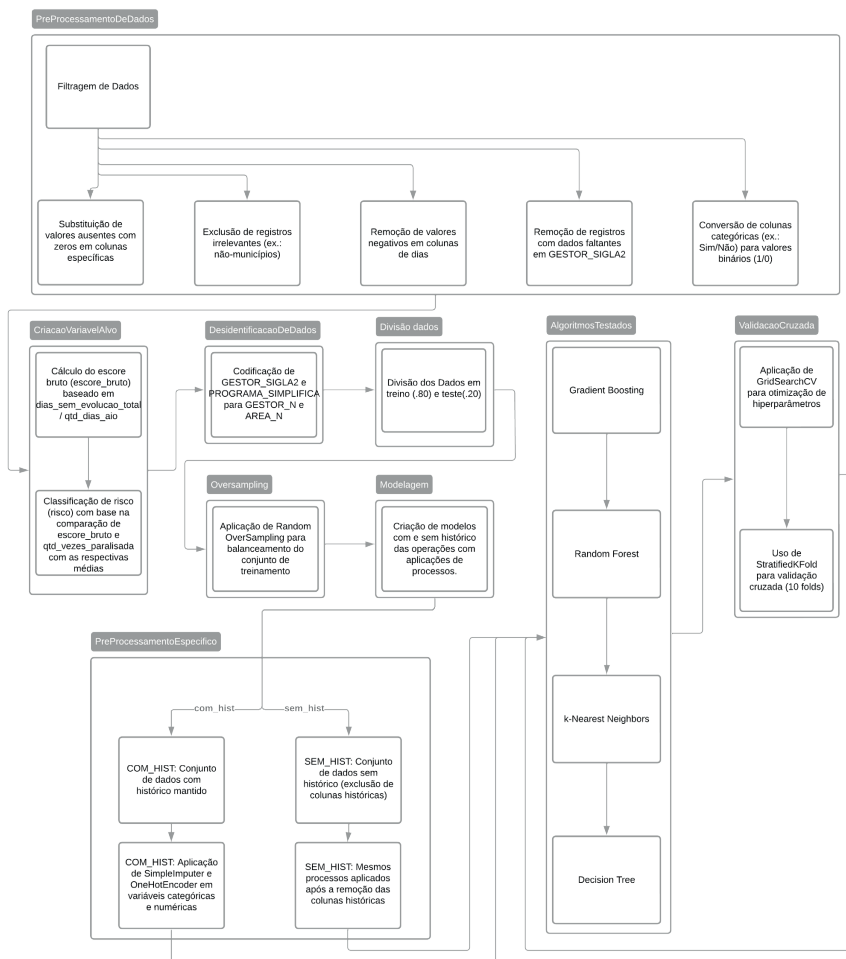
Este trabalho traz uma abordagem objetiva da avaliação de riscos para projetos de investimento público que permite identificar contratos que requeiram maior atenção em seu acompanhamento. Sua aplicação pode ser útil para a gestão de recursos públicos federais tanto para mitigar eventos de atrasos ou paralisações (que trazem prejuízo ao erário), bem como na construção de mecanismos de alocação mais eficientes de recursos públicos.

METODOLOGIA

Considerando que a ciência de dados possui caráter experimental, o planejamento de experimentos é uma etapa essencial para a coleta sistemática de dados de desempenho, com o objetivo de alcançar metas específicas ou testar hipóteses. O procedimento metodológico envolveu etapas de análise exploratória, pré-processamento e modelagem para avaliar o risco de paralisação de obras públicas, começando com a filtragem e desidentificação de dados para proteger a privacidade. Foi definida uma variável-alvo, aplicada técnica de sobre

amostragem, e construídos modelos com e sem histórico de dados, utilizando-se diferentes algoritmos com hiper-parâmetros otimizados por meio de busca em grade. A eficácia dos modelos foi avaliada por validação cruzada, curvas ROC, e o teste de McNemar para comparar o desempenho dos modelos. As variáveis mais importantes foram identificadas, e matrizes de confusão foram usadas para avaliar o desempenho, com o objetivo de determinar a melhor abordagem para prever o risco de paralisação. O procedimento descrito foi representado na Figura 1.

Figure 1 - O diagrama abaixo detalha o fluxo de trabalho realizado desde o pré-processamento até a modelagem, detalhando os algoritmos e técnicas utilizadas no estudo.



Coleta e Preparação dos Dados

A análise realizada neste estudo baseou-se em um conjunto de dados coletados de base de uso restrito, no entanto, a sua origem é pública, especialmente considerando se tratar de obras públicas. A data-base refere-se a junho/2024. Os dados brutos a partir dos quais podem ser geradas as variáveis utilizadas neste trabalho podem ser obtidas na área de downloads da CAIXA [10], sítio do SELO CAIXA GESTÃO SUSTENTÁVEL [5] e SIURB [11].

Os dados referem-se a operações de repasse de obras públicas situadas no Brasil. Os dados foram submetidos a um processo de preparação, que incluiu a limpeza de valores ausentes, a normalização de variáveis contínuas e a codificação de variáveis categóricas. Para as variáveis categóricas, foi utilizada a técnica de *OneHot Encoding*, que foi aplicada às colunas GESTOR e PROGRAMA, para garantir que todas as variáveis fossem adequadamente processadas antes de serem inseridas nos modelos preditivos.

A amostra completa inclui operações em diversos estágios. Os dados relacionados ao histórico de paralisação das operações existe a partir de dezembro/2019, data-base na qual foi iniciado o registro de informações sobre obras paralisadas.

Considerando o objeto de estudo, foram filtrados apenas os dados de operações relacionadas a municípios cuja Autorização de Início de Objeto (AIO) tenha sido emitida. Para tanto, foi realizado um filtro na característica IBGE (código do IBGE), bem como na característica que informa o tempo em dias entre a emissão da AIO e a data-base da amostra (`qtd_dias_aio`).

Além dos dados das operações, foram associados a cada operação, dados dos indicadores SELO CAIXA GESTÃO SUSTENTÁVEL [5] com vistas a identificar associações com 17 de seus 21 indicadores (os que são calculados através de dados públicos) com as demais características de contrato, especialmente considerando que os dados foram desidentificados no presente estudo (2.2).

Métodos de Proteção e Neutralidade dos Dados

Para assegurar a objetividade e integridade deste estudo, optou-se por omitir certas características dos dados coletados. Esta decisão foi tomada com

o intuito de evitar possíveis interpretações enviesadas ou controversas, especialmente no que tange a vieses com influências políticas.

- **Anonimização dos Dados:** Todos os dados foram cuidadosamente anonimizados para proteger a identidade dos indivíduos e entidades envolvidas, informações como número das operações, código do IBGE, nome de Ministérios Gestores ou Programas foram anonimizadas.
- **Agregação dos Dados:** No que concerne a Ministérios Gestores e programas governamentais, dados foram agregados em grupos para garantir a privacidade e reduzir a possibilidade de identificação específica.
- **Tratamento de Dados Sensíveis:** Reconhece-se a natureza sensível de algumas variáveis, e elas foram tratadas de forma a minimizar possíveis vieses interpretativos.
- **Neutralidade na Apresentação dos Dados:** Houve esforço para apresentar os dados de maneira neutra, evitando qualquer sugestão de favorecimento ou detrimento de qualquer grupo ou entidade, em conformidade com as melhores práticas de ética em pesquisa.

Características e descrição dos dados

A seleção de variáveis utilizadas obedeceu ao seguinte critério, tanto em uma modelagem quanto em outra:

- Variáveis que carregam identificação do ente municipal, da operação, do objeto em si ou do programa governamental não foram utilizadas, em acordo com o exposto no item 2.2.
- Variáveis que carregam informações de datas foram tratadas para conversão em dias, usando a data-base como referência, e não foram utilizadas neste estudo.
- Todas as demais foram utilizadas na modelagem com histórico da evolução das operações.
- Para a modelagem sem o histórico, foram desconsideradas as variáveis que carregavam informação histórica da evolução dos contratos. Nos grupos EVOLUÇÃO e TEMPO nenhuma variável foi utilizada; no grupo FINANCEIRO foram desconsideradas as variáveis relacionadas

a desbloqueio (liberação) de recursos.

Um diagrama apresentando as variáveis utilizadas em cada modelagem está representado na Figura 2.

Em que pese o escopo do trabalho utilize-se de dados públicos com vistas a uma avaliação objetiva, com vistas a mitigar risco de avaliações subjetivas acerca dos dados, optou-se por anonimizar os dados que de alguma maneira identifiquem de forma direta o município ou a operação, utilizando-se formas indiretas de capturar as características intrínsecas e extrínsecas das obras, como os indicadores do SELO [5].

Em etapa preliminar realizou-se a análise exploratória dos dados das principais variáveis numéricas utilizadas com vistas a entender melhor sua distribuição.

Criação da Variável-Alvo

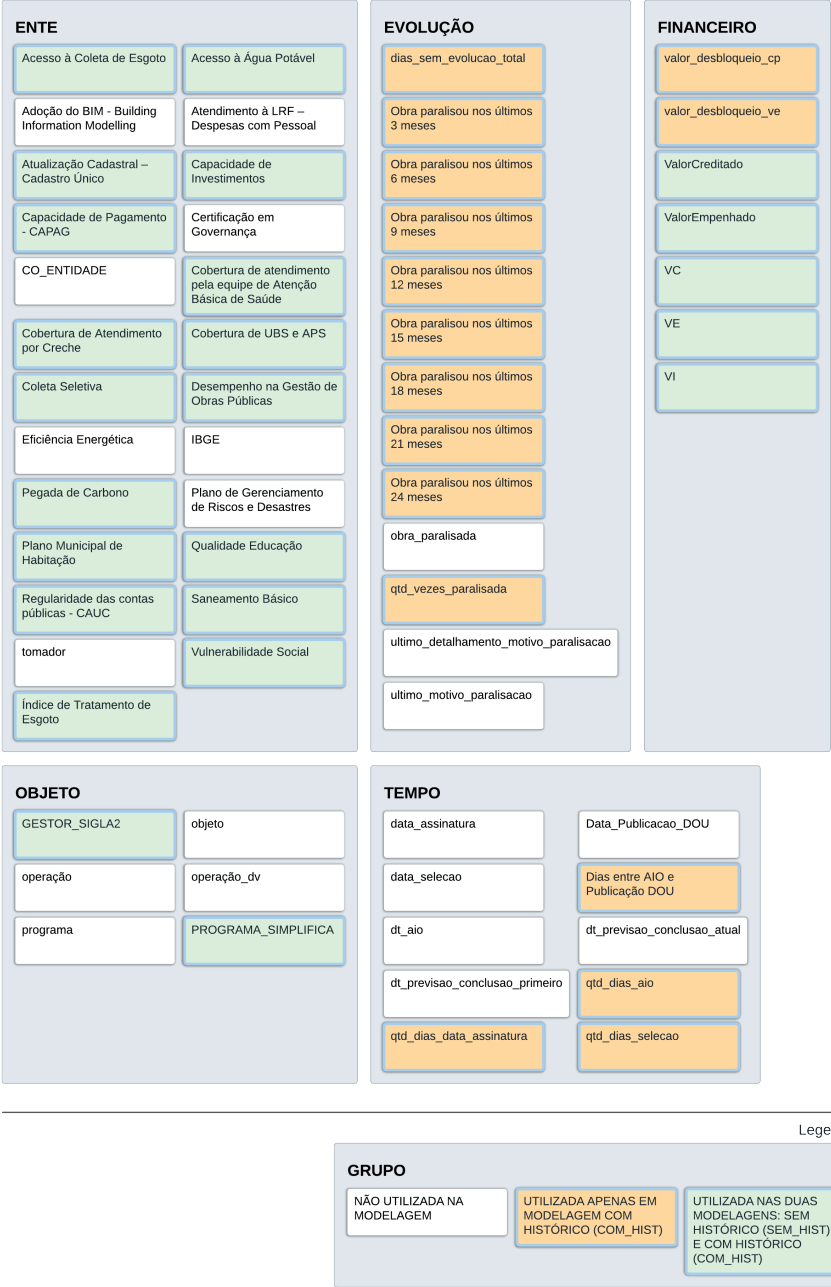
Uma das etapas cruciais deste estudo foi a criação de uma variável-alvo que pudesse servir como um *proxy* para medir a evolução das obras públicas analisadas.

De acordo com o Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK[®]) [12], os riscos são um aspecto da incerteza. Um evento ou condição incerta que, se ocorrer, provocará um efeito positivo ou negativo no projeto. Ainda de acordo com o guia, riscos negativos são chamados de ameaças e os riscos positivos são chamados de oportunidades, sendo ameaça um evento ou condição que, se ocorrer, tem um impacto negativo em objetivos, referindo-se a estratégias para lidar: prevenir, escalar, transferir, mitigar ou aceitar.

A variável-alvo foi construída para capturar o risco de atraso em função do tempo em que a obra ficou paralisada desde que a Autorização de Início de Objeto (AIO) foi emitida, ou média de vezes que paralisou, considerando serem dois marcadores distintos e importantes para identificar situação de maior atenção.

O objetivo é melhor gerenciar as operações diante de indicativos do risco de atraso ou paralisação de obras públicas, com medidas de prevenção ou mitigação.

Figure 2 - Esta figura apresenta um quadro com o esquema das variáveis utilizadas e não utilizadas nas modelagens (com e sem histórico), separadas por grupos.



O processo de criação desta variável envolveu os passos descritos adiante:

Cálculo do Escore Bruto

Inicialmente, foi calculado o escore bruto, que é definido pela seguinte

$$\text{escore_bruto} = \left(\frac{\text{dias_sem_evolucao_total}}{\text{qtd_dias_aio}} \right) \times 1000.$$

Este cálculo permite obter uma métrica quantitativa do tempo de paralisação em relação ao tempo total desde a emissão da AIO, ajustado para facilitar a interpretação.

Tratamento de *Outliers*

Após o cálculo do escore bruto, foi aplicada a técnica do Intervalo Interquartil (IQR) para identificar e tratar possíveis *outliers*. O IQR foi calculado como a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) da distribuição do escore bruto. Valores considerados *outliers* foram aqueles que caíam fora do intervalo definido por:

$$[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR].$$

Esses *outliers* foram removidos ou ajustados para garantir que os modelos preditivos não fossem influenciados por valores anômalos que poderiam distorcer os resultados.

Definição da Variável-Alvo

A variável-alvo, denominada **risco**, foi definida com base nas médias calculadas anteriormente. O risco foi considerado **aumentado** (valor 1) quando fosse atendida pelo menos uma das seguintes condições:

- Condição 1: o escore bruto da obra foi maior que a média do escore bruto, ou

- Condição 2: a quantidade de vezes que a obra foi paralisada foi maior que a média da quantidade de vezes paralisada.

Caso nenhuma das condições acima fosse atendida, o risco foi considerado **padrão** (valor 0).

Essa definição permite identificar de forma clara quais obras apresentam maior risco de atraso, aumentando a atenção para aquelas que apresentam maiores índices de paralisação ou mais dias sem evolução significativa.

Integração com o Modelo

Com a variável-alvo **risco** devidamente calculada e definida, ela foi integrada ao conjunto de dados como a variável que os modelos preditivos iriam tentar prever. Esse passo é crucial para garantir que o modelo esteja focado em prever o risco de atraso das obras.

Construção dos Modelos

Dois modelos preditivos foram desenvolvidos para estimar o risco de atraso em obras públicas. O primeiro modelo utilizou todas as variáveis disponíveis, enquanto o segundo modelo excluiu algumas variáveis previamente selecionadas, a fim de testar a influência dessas variáveis no desempenho geral.

Uma variável-alvo (*alvo*) foi construída como *proxy* para a evolução das obras, utilizando-se duas condições: a primeira relacionada à quantidade de vezes que a operação paralisou (*qtd_vezes_paralisada*); a segunda relacionando quantos dias ela esteve paralisada desde que recebeu a AIO: é a razão entre os dias em que a operação paralisou (*dias_sem_evolução_total*) e o tempo em dias que ela recebeu a AIO (*qtd_dias_aio*). A escala foi ajustada, multiplicando-se o valor por 1000 (mil), em seguida, foi aplicada a técnica do intervalo interquartil (IQR) para a identificação e tratamento de *outliers*.

Enquanto o primeiro modelo buscou entender a influência das variáveis em torno do alvo, o segundo buscou verificar se seria possível fazê-lo desconsiderando a influência dos dados ligados a elementos da história de cada operação, como o tempo desde que ela foi selecionada, o montante financeiro

desbloqueado até a data-base, ou o tempo transcorrido entre a publicação do contrato e a emissão de sua AIO.

Para a otimização dos hiperparâmetros dos modelos, foi empregada a técnica de busca em grade *GridSearchCV*, que permite a busca exaustiva através de uma grade especificada de parâmetros. Essa técnica foi aplicada tanto ao modelo completo quanto ao modelo reduzido.

RESULTADOS E DISCUSSÕES

Na seção de "Resultados e Discussão", apresenta-se uma análise detalhada dos dados e os resultados obtidos a partir dos modelos preditivos aplicados para identificar o risco de paralisação de obras públicas. A análise exploratória dos dados foi conduzida inicialmente para obter *insights* preliminares, incluindo a apresentação das estatísticas descritivas e gráficos relevantes apresentadas em na Figura 3.

Variável-alvo

A distribuição da variável-alvo apresentou desbalanceamento de classes, apresentando melhora nos resultados após aplicação de técnica de sobre amostragem, conforme é possível se observar na Figura 4.

Figure 3 - Análise exploratória das principais variáveis numéricas.

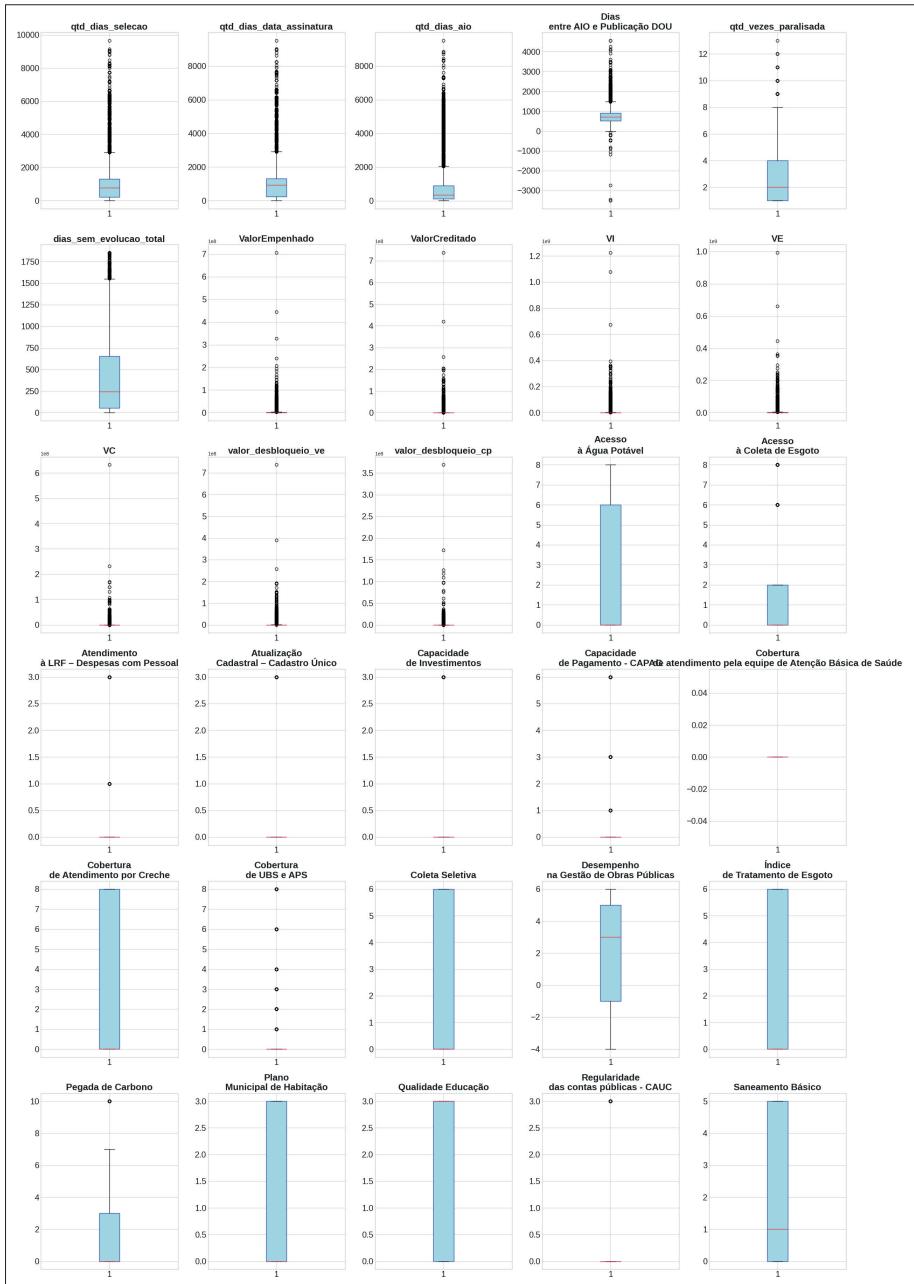
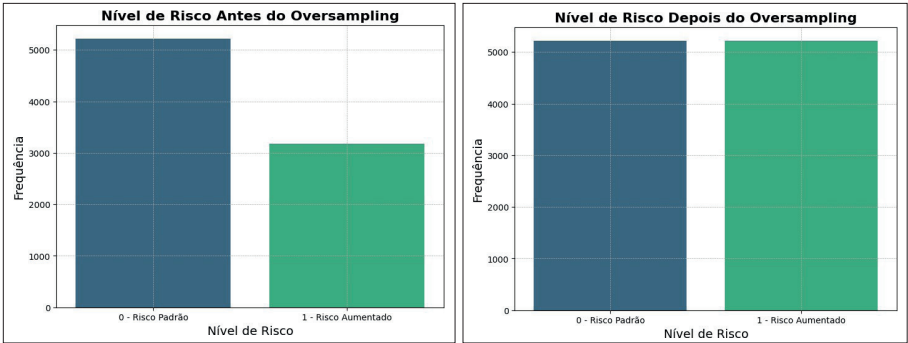


Figure 4 - Distribuição da variável alvo - Níveis de risco associados a obras públicas em operações de repasse, comparando antes e após a sobre amostragem.



Modelos

A avaliação do desempenho dos modelos é fundamental para determinar a eficácia das técnicas preditivas aplicadas na identificação do risco de atraso em obras públicas. Foram testados e comparados diversos algoritmos de aprendizado de máquina utilizando conjuntos de dados com e sem histórico de paralisações, denominados respectivamente como `com_hist` e `sem_hist`. Os modelos avaliados incluem *Gradient Boosting* (`gb`), *Random Forest* (`rf`), *k-Nearest Neighbors* (`kNN`) e *Decision Tree* (`tree`).

Métricas de Desempenho

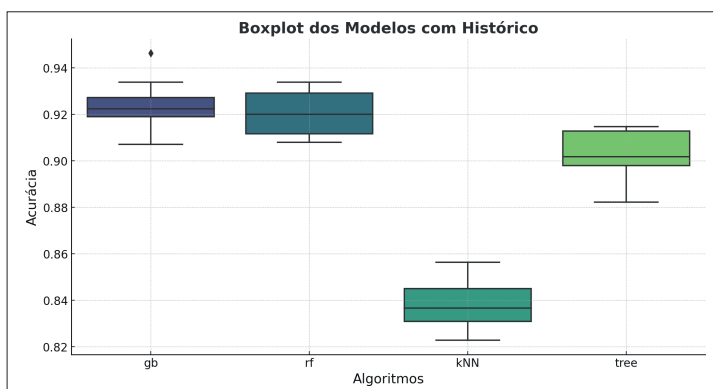
O desempenho dos modelos foi avaliado utilizando validação cruzada com 10 divisões (*folds*), calculando-se a acurácia média e o desvio padrão para cada modelo em ambos os conjuntos de dados. As tabelas a seguir resumem os resultados obtidos.

Desempenho nos Dados com Histórico (com hist)

Observa-se que tanto o *Gradient Boosting* quanto o *Random Forest* alcançaram a maior acurácia média de 92%, indicando um desempenho superior em relação aos outros modelos. O *k-Nearest Neighbors* apresentou a menor acurácia média de 84%, enquanto o *Decision Tree* obteve 90%.

Table 1 - Acurácia dos modelos no conjunto com histórico.

Modelo	Acurácia Média	Desvio Padrão
Gradient Boosting (gb)	0.92	± 0.01
Random Forest (rf)	0.92	± 0.01
k-Nearest Neighbors (kNN)	0.84	± 0.01
Decision Tree (tree)	0.90	± 0.01

Figure 5 - Desempenho dos modelos nos dados com histórico.**Table 2** - Acurácia dos modelos no conjunto sem histórico.

Modelo	Acurácia Média	Desvio Padrão
Random Forest (rf)	0.86	± 0.01
Gradient Boosting (gb)	0.85	± 0.01
Decision Tree (tree)	0.80	± 0.01
k-Nearest Neighbors (kNN)	0.79	± 0.01

Desempenho nos Dados sem Histórico (sem hist)

No conjunto sem histórico, o *Random Forest* obteve a melhor acurácia média de 86%, seguido de perto pelo *Gradient Boosting* com 85%. O *Decision Tree* e o *k-Nearest Neighbors* apresentaram desempenhos inferiores, com 80% e 79% respectivamente.

Análise Comparativa dos Modelos

A comparação entre os modelos evidencia que a inclusão de variáveis históricas melhora significativamente o desempenho preditivo. No conjunto

com_hist, os modelos alcançaram acurácias mais altas em comparação com o conjunto sem_hist, indicando que informações sobre paralisações passadas são relevantes para prever riscos futuros.

Tanto o *Gradient Boosting* quanto o *Random Forest* mostraram-se consistentes em ambos os conjuntos de dados, mas com desempenho superior quando o histórico é considerado. O *k-Nearest Neighbors* teve o desempenho mais baixo nos dois cenários, sugerindo que este algoritmo é menos adequado para este tipo de problema ou que requer ajustes adicionais de hiper-parâmetros, conforme (Figura 7) e (Figura 8).

Detalhamento dos Melhores Modelos

Ambos os modelos, Gradient Boosting no conjunto com_hist e Random Forest no conjunto sem_hist, seguiram pipelines idênticos em termos de pré-processamento e escalonamento.

Figure 6 - Desempenho dos modelos nos dados sem histórico.

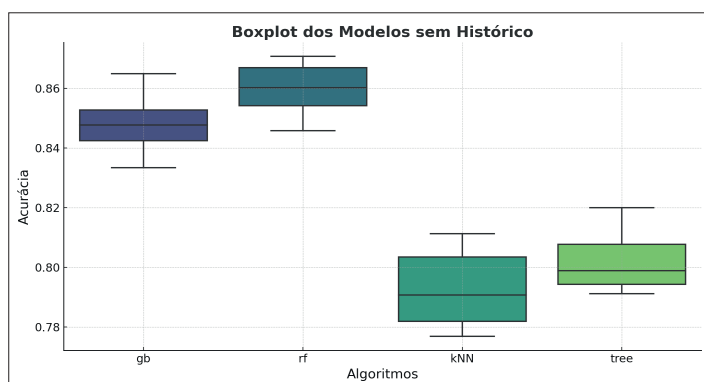


Figure 7 - Boxplot comparando resultados modelagem com e sem variáveis inerentes a dados do histórico da operação.

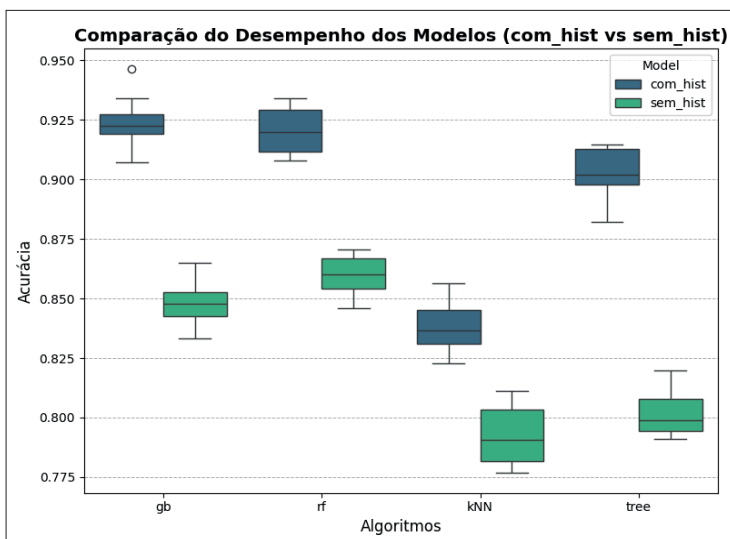
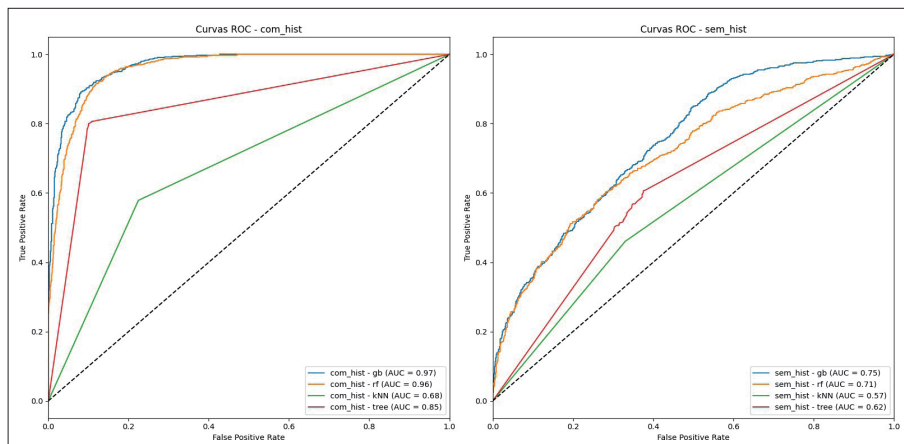


Figure 8 - Curvas ROC para modelagem com e sem variáveis inerentes a dados do histórico da operação.



SimpleImputer, codificação *one-hot* para variáveis categóricas, e normalização com *MinMaxScaler*. As diferenças principais entre eles estão nos algoritmos utilizados: o *Gradient Boosting* aplicou um *GradientBoostingClassifier* com 200 estimadores, enquanto o *Random Forest* utilizou um *RandomForestClassifier* com profundidade máxima de 20 e 200 estimadores.

Ambas as modelagens combinaram um pré-processamento com algoritmos eficientes, que foram capazes de capturar as relações complexas nos dados, tanto com histórico quanto sem histórico, demonstrando robustez e capacidade de predição satisfatória.

Matriz de Confusão

As matrizes de confusão para os modelos com e sem histórico indicam a eficácia dos modelos em classificar corretamente os exemplos:

- **Com histórico:** O modelo conseguiu classificar corretamente 1202 obras que não paralisaram e 705 que paralisaram nas condições estudadas, com menos falsos negativos (90) e falsos positivos (103).
- **Sem histórico:** Houve um aumento significativo em falsos negativos (315) e falsos positivos (372), indicando que o modelo sem histórico tem mais dificuldade em classificar corretamente as obras.

Figure 9 - Matriz de confusão - Com histórico.

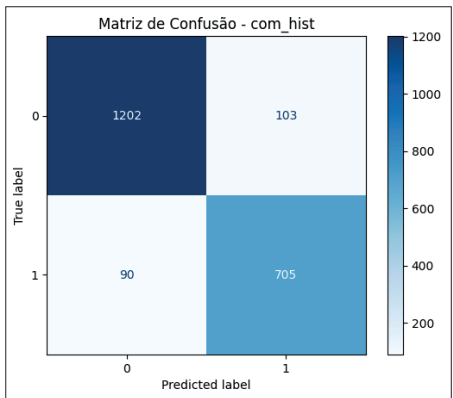
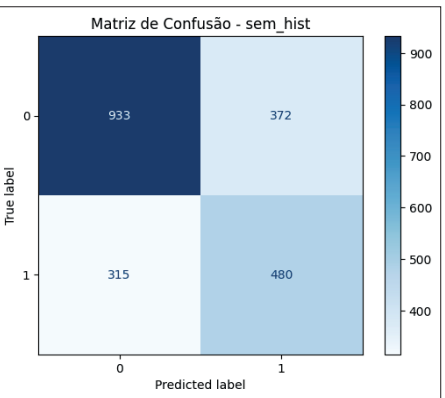


Figure 10 - Matriz de confusão - Sem histórico.



Teste de McNemar

Considerando a necessidade de comparação entre as diferentes estratégias de modelagem, aplicou-se o Teste de McNemar para avaliar a significância estatística das diferenças de desempenho entre os modelos com e sem histórico [9]. Utilizou-se a aproximação qui-quadrado, apropriada devido ao elevado número de discordâncias observado na matriz de contingência, que foi construída a partir da

comparação pareada entre os valores preditos pelos modelos *Gradient Boosting* no conjunto `com_hist` e *Random Forest* no conjunto `sem_hist`. Os resultados obtidos foram:

- **Estatística:** 352.65
- **Valor-p:** 0.0

Com um valor-p de 0.0, que é menor que o nível de significância de 0.05, o teste indica que há uma diferença estatisticamente significativa entre os resultados dos desempenhos dos modelos com e sem histórico. Entretanto, ao observar as métricas de avaliação de cada modelagem e considerando a opinião de especialistas, conclui-se que ambos os modelos podem ser aplicados para a estimativa da variável-alvo "risco".

Importância das Variáveis

A análise da importância das variáveis é um passo essencial para entender quais fatores têm maior influência nas previsões do modelo. Esta análise permite identificar as variáveis que mais contribuem para o desempenho do modelo, ajudando a focalizar esforços em dados que realmente impactam os resultados e a tomar decisões mais informadas na gestão das obras públicas.

A técnica utilizada para extrair a importância das variáveis foi baseada no uso de modelos de aprendizado de máquina que oferecem o atributo *feature importances*, presente no pacote *Scikit-learn*. Este atributo é comumente presente em modelos baseados em árvores de decisão, como *Random Forest* e *Gradient Boosting*. Para identificar a importância das variáveis, foi utilizada a função `feature_importances` presente nos modelos selecionados. Este atributo mede a contribuição relativa de cada variável na decisão do modelo e foi aplicado ao modelo que obteve melhor resultado na avaliação das métricas, quais sejam: *Gradient Boosting*, para o modelo com histórico e *Random Forest* para o modelo sem histórico. As variáveis transformadas foram extraídas e suas importâncias foram ordenadas em ordem decrescente para facilitar a visualização.

Para visualizar a importância das variáveis, gráficos de barras foram utilizados, onde o eixo das ordenadas representou as variáveis transformadas e o eixo das abscissas a importância relativa de cada variável.

Importância das Variáveis com Histórico (com hist)

Para o modelo que inclui histórico (`com_hist`), as variáveis com maior importância são as seguintes:

- **qtd dias seleção (0.4538)**: Esta variável, que representa o tempo desde a seleção da obra até a data-base, teve a maior importância no modelo. Isso sugere que quanto mais tempo a obra tem desde que foi selecionada, maior a probabilidade de haver atrasos ou paralisações.
- **valor desbloqueio ve (0.1534)**: O valor desbloqueado de repasses (valor de repasse) também se mostrou uma variável significativa. Obras com maior montante financeiro desbloqueado já superaram marcos de evolução que podem influenciar diretamente no prazo de sua execução.
- **Dias entre AIO e Publicação DOU (0.0861)**: O tempo entre a emissão da AIO e a publicação no Diário Oficial da União também foi uma variável importante, indicando que atrasos inerentes aos procedimentos administrativos comuns em operações com recursos públicos podem impactar o andamento das obras.
- **Desempenho na Gestão de Obras Públicas (0.0616)**: Esta variável reflete o desempenho geral da gestão das obras para cada ente tomador de recursos municipal, e também mostrou ter uma influência relevante no risco de atrasos.
- **Obra paralisou nos últimos 12 meses (0.0418)**: O fato de a obra ter paralisado nos últimos 12 meses foi outro fator significativo, sugerindo que interrupções recentes têm um grande impacto no risco futuro.

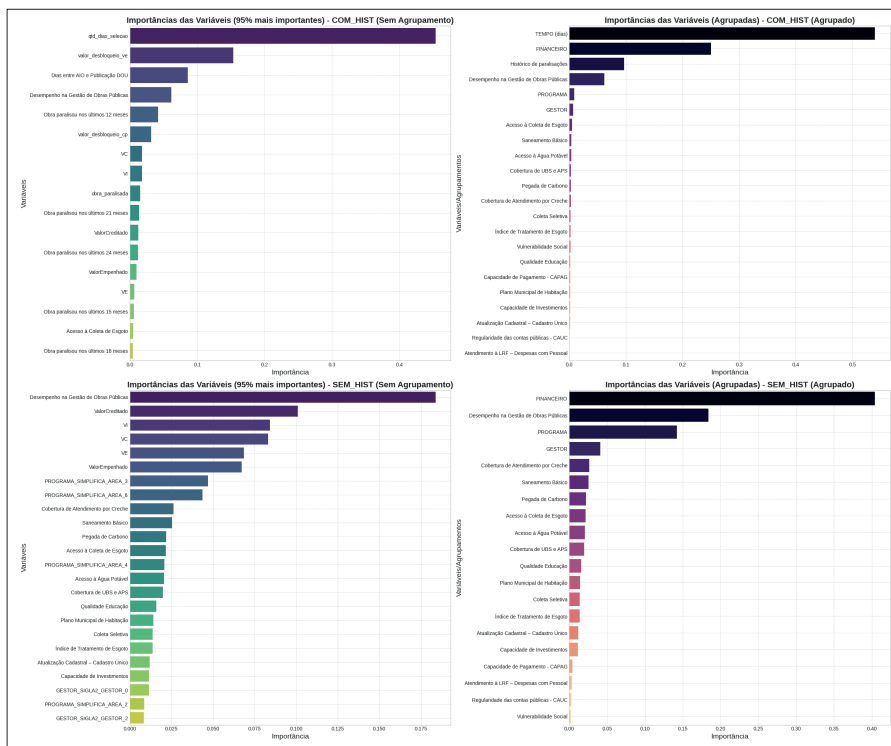
Outras variáveis, como `valor_desbloqueio_cp` (valor de contrapartida), `VC` (valor de contrapartida) e `VI` (valor de investimento), também apresentaram importâncias menores, mas ainda assim contribuíram para as previsões do modelo.

Importância das Variáveis sem Histórico (sem hist)

No modelo sem histórico (`sem_hist`), as variáveis mais importantes foram:

- **Desempenho na Gestão de Obras Públicas (0.1838):** Mesmo sem o histórico, o desempenho geral na gestão das obras de cada ente tomador (municípios, no caso em tela) figurou como a variável mais influente, indicando que a qualidade da gestão de obras públicas do ente municipal é um fator crítico para evitar atrasos.
- **ValorCreditado (0.1010):** O valor creditado à obra foi a segunda variável mais importante, sugerindo que o crédito de recursos para as obras desempenha um papel crucial na sua execução.

Figure 11 - Importância das características com histórico e sem histórico de operações - com e sem agrupamento por categorias exceto SELO.



VI (0.0843) e **VC (0.0831)**: Assim como no modelo com histórico, os valores de investimento e contrapartida continuam sendo variáveis de destaque, embora com um peso ligeiramente menor nesta modelagem.

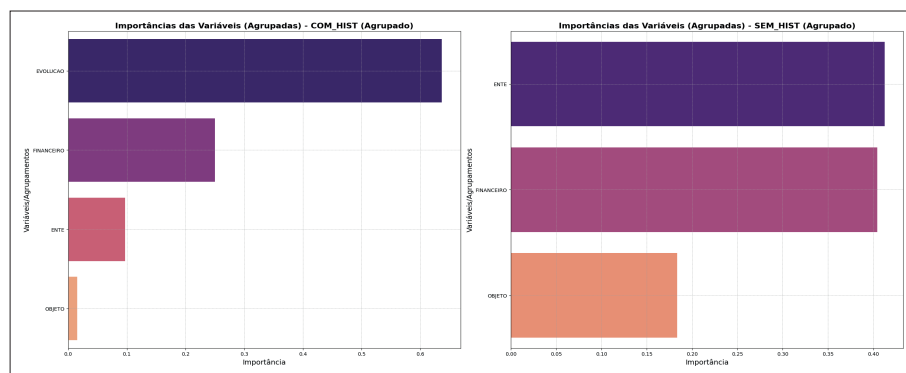
VE (0.0685) e **ValorEmpenhado (0.0673)**: Estes valores financeiros também mostraram-se relevantes, indicando que o comprometimento financeiro com a obra afeta diretamente sua execução.

Outras variáveis como PROGRAMA_SIMPLIFICA_ÁREA_3 e Cobertura de Atendimento por Creche também apareceram com alguma relevância, mas em menor grau comparados às principais variáveis mencionadas.

Os resultados demonstram que os modelos de aprendizado de máquina são capazes de prever o risco de atraso em obras públicas com elevado grau de acurácia, especialmente quando informações históricas estão disponíveis. O *Gradient Boosting* e o *Random Forest* destacam-se como as abordagens mais eficazes, apresentando desempenho superior consistentemente (Figura 8).

A significância estatística apontada pelo Teste de McNemar indica que, embora haja uma melhoria na acurácia com a inclusão de dados históricos, essa melhoria pode não ser mandatória para justificar a utilização de mais variáveis para todas as situações, dado que a escolha de incluir ou não dados históricos deve considerar não apenas o desempenho preditivo, mas também a disponibilidade e o custo de obtenção desses dados, bem como a necessidade da aplicação do algoritmo, por exemplo: se para avaliar uma carteira em evolução (com dados históricos disponíveis) ou uma amostra de novos projetos.

Figure 12 - Importância das características com histórico e sem histórico de operações - agrupamento por macro categorias.



Em síntese, a combinação de técnicas adequadas de pré-processamento e a seleção com base em critérios técnicos reconhecidos como boas práticas de algoritmos de aprendizado de máquina permitem a construção de modelos preditivos eficientes para auxiliar na gestão de carteiras de operações de obras públicas, podendo ser útil para a identificação precoce de riscos e a tomada de decisões pertinentes.

A análise das importâncias das variáveis mostra que, tanto no modelo com histórico quanto no modelo sem histórico, as variáveis financeiras (como ValorCreditado, VI, VC, VE) e de desempenho (como Desempenho na Gestão de Obras Públicas) são consistentes em termos de importância. Isso destaca a necessidade de monitorar de perto os aspectos financeiros e de gestão das obras públicas de cada ente municipal para mitigar riscos de atrasos ou paralisações.

No modelo com histórico, a variável `qtd_dias_seleção` se destacou significativamente, sugerindo que o tempo desde a seleção da obra é um preditor muito forte de atrasos ou paralisações, provavelmente devido a causas como o aumento dos custos financeiros decorrentes de efeitos inflacionários ou relacionados ao valor do dinheiro no tempo. Já no modelo sem histórico, a importância relativa de variáveis relacionadas à gestão municipal, demonstrando que o indicador de desempenho de obras públicas do SELO CAIXA GESTÃO SUSTENTÁVEL pode ser observado para avaliar o risco de atraso ou paralisação de obras especialmente em obras em estágios mais iniciais, dado que as variáveis históricas não estão presentes.

Na Figura 12, pode-se observar a importância das variáveis agrupadas por EVOLUÇÃO, FINANCEIRO, ENTE e OBJETO, para os dois modelos, com exceção de EVOLUÇÃO para o modelo sem histórico. Depreende-se que, considerando o histórico, o fator determinante para o risco é a EVOLUÇÃO, seguido de FINANCEIRO.

Já na situação sem histórico, o ENTE (municipal) e os aspectos FINANCEIROS são bem mais importantes para entender o risco da operação.

Em uma situação real, a modelagem que não leva consideração dados históricos pode ser útil para entender onde melhor direcionar esforços de gestão para evitar problemas futuros, e olhar para o indicador de desempenho de obras públicas do SELO pode ser útil para isso.

Essa análise da importância das variáveis utilizadas em cada modelagem, com ou sem histórico, se constitui elemento relevante na busca da mitigação de riscos de fatores que impactam o sucesso de obras, tanto com vistas a futuro desenho de alocação de recursos, quanto na melhoria contínua dos processos de gestão das obras públicas.

Neste mesmo diapasão, no aspecto agrupado objeto, observar a área do objeto, o programa e o Ministério Gestor das operações podem trazer informações relevantes associadas ao risco de paralisação ou atraso de obras.

CONCLUSÃO

Foi possível estimar, com o uso de algoritmos de inteligência artificial, o risco de paralisação ou atraso de obras públicas no país. Adicionalmente pode ser verificada a importância das características contidas na base de dados de obras públicas estudada, associadas ao risco de paralisações em obras públicas relacionadas a operações de repasse. Foram utilizadas e comparadas duas abordagens: uma considerando características relacionadas ao histórico das operações (*com histórico*) e outra utilizando a mesma base, mas sem a inclusão dessas características (*sem histórico*). As abordagens resultaram em dois modelos de predição com acurácia aproximada de 0,91 (*com histórico*) e 0,86 (*sem histórico*).

O modelo de **Gradient Boosting** e **Random Forest** mostraram-se os mais robustos e consistentes, apresentando as maiores AUCs em ambos os conjuntos de dados, sendo o **Gradient Boosting** o mais eficaz para o conjunto de dados *com histórico* e o **Random Forest** para o conjunto de dados *sem histórico*. Por outro lado, os modelos de **KNN** e **Decision Tree** mostraram desempenho inferior em ambos os conjuntos de dados.

Dessa forma, foram desenvolvidos dois modelos com a capacidade de estimar o risco de paralisação em obras públicas de repasse com base em um conjunto de dados. A primeira modelagem levou em conta um maior número de variáveis, incluindo as relacionadas ao histórico das operações, já a segunda modelagem utilizou parte das variáveis utilizadas na primeira, desconsiderando características mais diretamente relacionadas ao histórico das operações.

Tal achado é relevante como apoio à gestão pública no sentido que permite identificar com antecedência quais operações merecem uma maior atenção na

atuação, bem como quais municípios podem precisar de mais assistência técnica na execução de suas obras, o que pode contribuir para diminuição do estoque de operações paralisadas no país.

CONSIDERAÇÕES FINAIS

O presente estudo é um ponto de partida para novas linhas de investigação que podem aprofundar, detalhar ou ampliar os resultados obtidos. Apresenta-se algumas possibilidades vislumbradas para os futuros estudos:

Outros recortes de estudo devem ser investigados, como o conjunto de obras concluídas desde o início do monitoramento de registros sobre paralisações. Isso permitirá capturar um ciclo de execução completo, comparando projetos incluídos em um mesmo período temporal.

A utilização de dados de outros órgãos gestores públicos ou outras fontes de recurso poderão ser valiosas para validar e comparar a metodologia deste trabalho em diferentes contextos.

A continuidade do monitoramento dos registros de acompanhamento das operações, incluindo paralisações e atrasos contratuais, é considerada essencial. Isso permitirá a realização de novos estudos com uma base de dados mais ampla, possibilitando a exploração de mais possibilidades de pesquisa.

A construção ou consolidação de mais dados estruturados de obras públicas, bem como sua publicização são determinantes para a realização de novos estudos que levem em conta variáveis mais detalhadas sobre ESCOPO, PRAZO, CUSTO e QUALIDADE das obras públicas com o intuito de melhor estimar os riscos de projetos, bem como possibilitar a mensuração do impacto financeiro de atrasos ou paralisações em obras, o que é crucial para a melhoria da gestão e alocação de recursos públicos.

As propostas para pesquisas futuras podem tanto complementar os achados do presente estudo, como trazer informações valiosas para o aprimoramento das melhores práticas de gestão de obras públicas.

Agradecimentos

Ao Professor Dr. Filipe Alves Verri Neto. Ao Engenheiro Civil Antemildo Batista de Andrade.

À CAIXA ECONÔMICA FEDERAL, através de sua Gerência Nacional de Soluções em Operações de Governo e aos meus colegas de trabalho.

REFERÊNCIAS

World Bank. (2023). *Infrastructure Needs in Brazil: The Road to 2030*. Washington, DC: World Bank. Disponível em: <https://documents1.worldbank.org/curated/en/099140006292213309/pdf/P1745440133da50c0a2630ad342de1ac83.pdf>.

Tribunal de Contas da União, *Acórdão 11196/2018*, Processo: 1119620181. Disponível em: <https://pesquisa.apps.tcu.gov.br/documento/acordao-completo/1119620181.PROC/%20DTRELEVANCIA%20desc,%20NUMACORDAOINT%20desc/0/%20?uuid=e62497d0-8152-11e9-acc9-c9d47db6992f>. Acesso em: 28 ago. 2024.

Caixa Econômica Federal. *Obras paralisadas:proposições construtivas para a boa evolução das obras públicas e para evitar paralisações*. Disponível em: <https://www.caixa.gov.br/Downloads/desenvolvimento-urbano-obras-paralisadas-proposicoes-construtivas/cartilha-obras-paralisadas-proposicoes-construtivas.pdf>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal, *Relatório Integrado 2023*, p. 60. Disponível em: <https://www.caixa.gov.br/Downloads/caixa-governanca/relatorio-integrado-2023.pdf>. Acesso em: 28 ago. 2024.

Caixa Econômica Federal, *Gestão Sustentável-Selo*, disponível em: <https://www.caixa.gov.br/sustentabilidade/gestao-sustentavel/selo>, acesso em: 19 ago. 2024.

Tribunal de Contas da União, *Convênios e Outros Repasses*, 6ª ed., 2011. Disponível em: <https://www.gov.br/plataformamaisbrasil/pt-br/manuais-e-cartilhas/arquivos-e-imagens/convenios-e-outros-repasses-6a-edicao-tcu>. Acesso em: 28 ago. 2024.

GUSMÃ O, G. M.; HOLANDA, J. T.; FIALHO, H. V. M. L.; MOREIRA, F. O. P.; GOMES, D. A.; CEOLIN, A. C.; CABRAL, R. M. Monitoramento e controle no gerenciamento de projetos do setor público brasileiro: uma revisão sistemática da literatura. *Revista Gestão e Secretariado (GeSec)*, v. 14, n. 4, p. 6494-6505, 2023. DOI: <http://doi.org/10.7769/gesec.v14i4.2062>.

PAULA, C. P.; CORDEIRO, G. A.; RAMPASSO, I. S.; ORDÓ ÑEZ, R. E. C.; ANHOLON, R. Métodos Quantitativos para Gestão de Risco em Projetos: Uma Revisão da Literatura. *GEPROS. Gestão da Produção, Operações e Sistemas*, v. 14, n. 2, p. 129-148, 2019.

VERRI, Filipe A. N. *Data science project: an inductive learning approach*. [s.l.]: [s.n.], 2024. Disponível em: <https://github.com/verri/dsp-book>. Acesso em: 14 ago. 2024.

Caixa Econômica Federal, *Orçamento Geral da União – Base de Dados*, disponível em: <https://www.caixa.gov.br/site/paginas/downloads.aspx>, acesso em: 19 ago. 2024.

Acompanhamento de Operações, SIURB. *Caixa Econômica Federal*. Disponível em: <https://webp.caixa.gov.br/siurb/ao/pag/index.asp>. Acesso em: 26 ago. 2024

Project Management Institute, *Guia PMBOK® – 7ª Edição*, 2021.

Caixa Econômica Federal. Acesso à Água Potável. *Sistema Nacional de Informações sobre Saneamento (SNIS) / Ministério do Desenvolvimento Regional*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 4-5. Disponível em: <http://app4.mdr.gov.br/serieHistorica/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Acesso à Coleta de Esgoto. *Sistema Nacional de Informações sobre Saneamento (SNIS) / Ministério do Desenvolvimento Regional*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 5-6. Disponível em: <http://app4.mdr.gov.br/serieHistorica/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Atendimento à LRF – Despesas com Pessoal. *Tesouro Transparente / Secretaria do Tesouro Nacional (STN)*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 14-15. Disponível em: <https://www.tesourotransparente.gov.br/temas/estados-e-municipios/capacidade-de-pagamento-capag>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Atualização Cadastral – Cadastro Único. *Ministério da Cidadania*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 11-12. Disponível em: <https://cecad.cidadania.gov.br/painel03.php>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Capacidade de Investimentos. *Índice FIRJAN de Gestão Fiscal/ FIRJAN*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 14. Disponível em: <https://www.firjan.com.br/ifgf/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Capacidade de Pagamento - CAPAG. *Tesouro Transparente / Secretaria do Tesouro Nacional (STN)*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 15-16. Disponível em: <https://www.tesourotransparente.gov.br/temas/estados-e-municipios/capacidade-de-pagamento-capag>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Cobertura de atendimento pela equipe de Atenção Básica de Saúde. *Cadastro Nacional de Estabelecimentos de Saúde / Ministério da Saúde e e-Gestor Atenção Primária / Ministério da Saúde*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 9-10. Disponível em: <http://tabnet.datasus.gov.br/cgi/tabcgi.exe?cnes/cnv/estabbr.defe> <https://egestorab.saude.gov.br/paginas/ acessoPublico/relatorios/relHistoricoCobertura.xhtml>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Cobertura de Atendimento por Creche. *Observatório da Criança / Fundação ABRINQ*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 9. Disponível em: <https://observatoriocrianca.org.br/cenario-infancia/temas/educacao-infantil/1081-taxa-bruta-de-matricula-em-creches?filters=177>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Cobertura de UBS e APS. *Cadastro Nacional de Estabelecimentos de Saúde / Ministério da Saúde e e-Gestor Atenção Primária / Ministério da Saúde*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 9-10. Disponível em: <http://tabnet.datasus.gov.br/cgi/tabcgi.exe?cnes/cnv/estabbr.defe> <https://egestorab.saude.gov.br/paginas/ acessoPublico/relatorios/relHistoricoCobertura.xhtml>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Coleta Seletiva. *Sistema Nacional de Informações sobre Saneamento (SNIS) / Ministério do Desenvolvimento Regional*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 6. Disponível em: <http://app4.mdr.gov.br/serieHistorica/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. Desempenho na Gestão de Obras Públicas. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 17-18. Disponível em: <https://www.caixa.gov.br/sustentabilidade/selo-gestao-sustentavel/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Índice de Tratamento de Esgoto. Sistema Nacional de Informações sobre Saneamento (SNIS) / Ministério do Desenvolvimento Regional*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 5. Disponível em: <http://app4.mdr.gov.br/serieHistorica/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Pegada de Carbono. Emissão Líquida de tCO₂e do município no banco de dados SEEG e IBGE*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 19-20. Disponível em: <https://plataforma.seeg.eco.br/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Plano Municipal de Habitação. Instituto Brasileiro de Geografia e Estatística (IBGE)*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 12. Disponível em: <https://www.ibge.gov.br/estatisticas/sociais/saude/10586-pesquisa-de-informacoes-basicas-municipais.html?=&t=destaques>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Qualidade Educação. Índice de Desenvolvimento da Educação Básica (IDEB) / Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP)*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 11. Disponível em: <http://ideb.inep.gov.br/resultado/>. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Regularidade das contas públicas - CAUC. Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi) / Secretaria do Tesouro Nacional (STN)*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 15. Disponível em: https://siconfi.tesouro.gov.br/siconfi/pages/public/sti/iframe_sti.jsf. Acesso em: 21 jul. 2024.

Caixa Econômica Federal. *Saneamento Básico. Sistema Nacional de Informações sobre Saneamento (SNIS) / Ministério do Desenvolvimento Regional*. In: Guia Selo CAIXA Gestão Sustentável, versão 1.4, p. 7-8. Disponível em: <http://app4.mdr.gov.br/serieHistorica/>. Acesso em: 21 jul. 2024.

ESTUDO DE DESEMPENHO DE MODELOS DE LINGUAGEM DE GRANDE ESCALA EM MARATONAS DE PROGRAMAÇÃO

Vitor Laguerra Netto Gioia

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)
vitorlngioia@gmail.com

Cesar Augusto Cavalheiro Marcondes

Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

This study investigates the performance of Large Language Models (LLMs) in solving problems presented in the Maratona de Programação da Sociedade Brasileira de Computação (SBC) 2023 edition. The research aimed to assess the capability of these models to generate correct code for the challenges stated in the competition, using the Python language. Two state-of-the-art LLMs were selected: GPT-4 and Llama 3.1. The methodology involved preparing prompts from problem statements, code testing and applying metrics to evaluate the performance of the generated code. The results indicate that while the models achieved reasonable success rates, they still face limitations, particularly with more complex problems. This study contributes to understanding the potential and constraints of LLMs in specific programming contexts, suggesting pathways for future research and the enhancement of these technologies.

RESUMO

O presente estudo investiga o desempenho de Modelos de Linguagem de Grande Escala (LLMs) na resolução de problemas enunciados na Maratona de Programação da Sociedade Brasileira de Computação (SBC) edição 2023. A pesquisa teve como objetivo avaliar a capacidade desses modelos em gerar códigos corretos para desafios propostos na competição, utilizando a linguagem Python. Para tanto, foram escolhidos dois LLMs: o GPT-4o e o Llama 3.1, ambos considerados estado da arte em seus respectivos contextos. A metodologia envolveu a preparação de prompts a partir dos enunciados de problemas, testagem de códigos com arquivos fornecidos pela maratona e a aplicação de métricas para avaliação da performance dos códigos gerados. Os resultados indicam que, embora os modelos tenham apresentado taxas de sucesso razoáveis, a geração de código ainda enfrenta limitações, especialmente em problemas mais complexos. Este estudo contribui para a compreensão do potencial e das restrições dos LLMs em contextos específicos de programação, sugerindo caminhos para futuras pesquisas e aprimoramento dessas tecnologias.

INTRODUÇÃO

A evolução acelerada dos modelos de linguagem de grande escala (LLMs) emerge como uma transformação no vasto campo de inteligência artificial. O início da área de estudo data de meados do século XX e desde então, foi marcada por avanços irregulares no tempo, de forma que alguns pesquisadores dividem sua história em fases de desenvolvimento[Muthukrishnan *et al.* 2020].

No seu período mais recente, houve um marcante e acelerado crescimento do campo, quando passaram a existir uma variedade de ferramentas sofisticadas capazes de executar inclusive tarefas complexas. Os LLMs, nesse contexto, impactaram domínios diversos, incluindo processamento de linguagem natural (PLN), saúde, educação, negócios e mais recentemente em tarefas de programação.

Perante essas possibilidades de aplicação, a presente pesquisa visa o aprofundamento da utilização dos modelos de linguagem de grande em escala em tarefas de programação. Em face dessa finalidade, nesta seção será abordado um contexto geral da evolução dos LLMs desde suas primeiras iniciativas em tarefas diversas até os trabalhos mais modernos especializados em programação. Ao fim, será enunciado os objetivos gerais e específicos

Contextualização

O estágio inicial da história dos LLMs, nos anos de 1960, foi marcado pelo surgimento de alguns modelos que simulavam conversas humanas. Um exemplo foi o projeto ELIZA [Weizenbaum 1966] que possuía um contexto muito limitado de conversação. Dessa época para o estágio atual, observou-se um avanço relevante do poder computacional e sofisticação dos algoritmos. Alguns marcos de toda essa jornada foram o surgimento do campo de redes neurais nos anos de 1980 e de técnicas de aprendizado profundo na década de 2010 [LeCun *et al.* 2015]

A contribuição mais recente que impulsionou o campo foi o desenvolvimento dos *transformers* [Vaswani *et al.* 2017]. Trata-se de uma arquitetura de aprendizado profundo que utiliza mecanismos de atenção e que viabilizou ferramentas robustas para lidar de forma eficiente com longas cadeias de dados sequenciais, típicas de textos em linguagem natural.

No contexto do surgimento dessas técnicas, alguns modelos demonstraram uma capacidade distinta em resolver tarefas como geração de texto, sumarização e análise de sentimento. O lançamento da versão *Generative Pre-trained Transformer* (GPT-2) da OpenAI foi um marco para esse período [Radford *et al.* 2019]. Desde então, foram liberadas novas versões da própria companhia (GPT-3 em diante) e de outras empresas, que sucessivamente mostraram melhorias de desempenho no que diz respeito ao processamento de entradas e saídas, bem como à coerência das respostas [Brown 2020].

Além da própria evolução da capacidade de raciocínio e processamento, com o aprimoramento de técnicas de formulação de entrada, conhecida também como engenharia de *Prompt*, percebeu-se o aumento de performance na resolução de tarefas, incluindo as de objeto desse estudo, as de programação [Brown 2020]. Relacionado a esse tipo, observa-se um desempenho dos modelos nas seguintes aplicações: geração de código, complementação de um código existente, tradução de uma linguagem para outra, documentação e detecção e correção de falhas.

A complexidade da tarefa de geração de código varia com a dificuldade do problema e, em um contexto de projeto de software, com o nível em que ela se propõe em resolver. Alguns temas ainda são considerados problemas em aberto, como a complementação em nível de repositório [Zhang *et al.* 2023]. Isso porque os modelos de linguagem enfrentam desafios ao tentar generalizar respostas quando lhes é fornecidos repositórios não observados em treinamento.

Como consequência disso, para o tipo de tarefa em nível de função existem *bench-marks* para comparação de modelos sendo amplamente utilizados há pelo menos alguns anos, como o *HumanEval* [Chen *et al.* 2021]. Enquanto que para níveis de complexidade de estrutura superiores, como o *RepoCoder* [Zhang *et al.* 2023], observaram-se estudos apenas muito recentemente. [Peng *et al.* 2023] [Austin *et al.* 2021]

Problema de Pesquisa

O problema endereçado nesta pesquisa relaciona-se a esses trabalhos recentes, utilizando ferramentas existentes para resolução de tarefas de geração de código. Selecionando alguns desafios da Maratona de Programação da Sociedade Brasileira de Computação (SBC), enuncia-se o seguinte problema

de pesquisa: **LLMs geram bons códigos para problemas da Maratona de Programação da SBC?**

Objetivo Geral

O objetivo desta pesquisa é “estudar o desempenho de LLMs na resolução de problemas enunciados nas Maratonas de programação”. Se identificados bons resultados no estudo, verifica-se a possibilidade de utilizá-los em problemas mais complexos, seja em dificuldade, seja em um projeto estruturado de software, contribuindo na geração de código em múltiplos arquivos.

Objetivos Específicos

Os objetivos específicos apresentam, de forma detalhada, as ações que se pretende alcançar e estabelecem estreita relação com as particularidades relativas ao objetivo geral já apresentado. Desta forma, os seguintes objetivos específicos foram identificados:

- Aplicar modelos LLM em diferentes configurações para resolução de problemas de Maratonas de Programação;
- Especificar métricas para avaliação dos modelos nas tarefas enunciadas;
- Observar o desempenho dos modelos nas particularidades dos problemas - critérios de entrada e saída e enunciados em língua portuguesa brasileira.

DESENVOLVIMENTO

Com o objetivo de levantar as ferramentas necessárias para avaliação de LLMs, realizou-se um estudo inicial de regras da Maratona de Programação, de como os problemas são enunciados e como são avaliados pela comissão do evento. Além desses critérios, levantou-se o que havia de disponível de modelos e técnicas em artigos e materiais disponibilizados pelas instituições responsáveis e parceiras.

Desafios escolhidos

Escolheu-se como conjunto de problemas a ser resolvido o da primeira fase da edição da Maratona de Programação SBC de 2023 [SBC 2023a]. A maratona possui 13 enunciados no "Caderno de Problemas", que devem ser implementados segundo alguns critérios detalhados tanto nesse documento quanto no de "Informações".

Para os fins desta pesquisa, nem todos os critérios foram seguidos rigorosamente, por ter sido desenvolvida fora do ambiente da maratona e suas plataformas de avaliação e submissão. A seguir serão detalhados os critérios obedecidos e ignorados pelos algoritmos implementados.

Critérios obedecidos

- Linguagem de programação aceita;
- Formatos de entrada e saída do programa;
- Execução dos casos de testes disponibilizados;

Quanto à linguagem de programação para resolução dos problemas e que será, portanto, refletida em códigos inferidos pelos modelos, escolheu-se Python. Além de ser uma das possíveis estabelecidas pela comissão organizadora, trata-se de uma linguagem amplamente utilizada nos treinamentos dos LLMs.

Os formatos de entrada e saída já seriam naturalmente obedecidos, visto que os casos de testes disponibilizados na página da edição 2023 possuem a exata formatação estabelecida por esses critérios. Assim como consta no "Caderno de Problemas", a entrada apresenta as seguintes características:

1. A entrada de seu programa deve ser lida da entrada padrão;
2. A entrada é composta de um único caso de teste, descrito em um número de linhas que depende do problema.
3. Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco;
4. Cada linha, incluindo a última, contém exatamente um caractere final-de-linha;

5. O final da entrada coincide com o final do arquivo. Sobre a saída:
6. A saída de seu programa deve ser escrita na saída padrão.
7. Quando uma linha da saída contém vários valores, estes devem ser separados por um único espaço em branco; a saída não deve conter nenhum outro espaço em branco.
8. Cada linha, incluindo a última, deve conter exatamente um caractere final-de-linha.

Quanto aos testes, avaliou-se a corretude da execução do código gerado pelo LLM em um formato diferente do sugerido no arquivo de informações, o qual instrui o uso da interface *web* do Boca. A maneira escolhida para execução foi implementar um Shell script que executa o código gerado a partir de entradas disponibilizadas na página inicial do evento e compara saídas obtidas com as esperadas, as quais também estavam disponíveis (aos pares com as entradas).

Critérios ignorados

- Limites de tempo de execução;
- Limites de memória;
- Limite de tamanho do arquivo fonte;
- Limite de tamanho do arquivo de saída;
- Formato de submissão.

Dentro do ambiente do desafio, todos esses critérios são devidamente validados ao se obedecer as instruções de submissão dos códigos à plataforma Boca. Como esta não foi utilizada pelo presente estudo, implementou-se uma simplificação para avaliar primordialmente a corretude dos códigos gerados, como já previamente introduzido na seção anterior.

Ferramentas e recursos utilizados

Como anteriormente explicado, escolheu-se a linguagem de programação Python, especificamente na versão 3.8 para geração de códigos pelos modelos

LLMs selecionados. Em conformidade com essa definição, a linguagem também foi utilizada para:

1. Preparação e organização dos arquivos de testes disponibilizados pelo site da maratona no diretório do projeto;
2. Preparação dos prompts a partir dos enunciados no "Caderno de Problemas";
3. Requisição aos modelos enviando os prompts como entrada;
4. Avaliação geral de cada modelo para o conjunto de tarefas especificadas.

Como os arquivos baixados pelo site são compactados em dois níveis, utilizou-se um *script* Python para descompactá-los e copiá-los para diretórios mais próximos à raiz do projeto, de forma a facilitar o acesso para *scripts* que fossem executados posteriormente.

Quanto à preparação de prompts, os enunciados dos problemas foram copiados manualmente do arquivo PDF "Caderno de Problemas" para arquivos texto sem qualquer tratamento. Com o objetivo de formatar e remover o que poderiam ser ruídos para processamento do texto pelo modelo, implementou-se um outro *script*. Mais detalhes sobre a preparação da entrada e sua importância na seção a seguir.

Desenvolveu-se um terceiro código com a finalidade de concentrar todas as configurações dos modelos e *prompts* a serem requisitados. Esse arquivo também é responsável por definir a quantidade de amostras geradas por problema e arquivo de saída para cada uma dessas respostas.

Por último, implementou-se um *script* para avaliação dos modelos segundo métrica utilizada frequentemente na literatura para o tipo de tarefa de geração de código. Para este em específico, além das bibliotecas-padrão da linguagem, foi utilizada a *numpy* na sua versão 2.0.1. O formato de cálculo será descrito com maior profundidade numa seção específica sobre o assunto mais adiante.

Além dessas ferramentas, utilizou-se um *script* Shell para execução dos códigos gerados com as entradas fornecidas. Todo o código pode ser encontrado no repositório GitHub <https://github.com/vgioia/llm-code-inference>. Os modelos LLMs utilizados serão detalhados em uma seção específica sobre o assunto adiante.

Os códigos foram executados em uma máquina Linux Ubuntu 20.04.6 LTS 64-bit, processador Intel® Core™ i7-10610U CPU @ 1,80GHz×8 e memória 15,2 GiB, com modelos sendo executados nos servidores em que hospedam suas versões API.

Prompting

Como previamente explicado, implementou-se um *script* dedicado para a devida preparação das entradas para os modelos. Relacionada a essa etapa existe uma subárea importante para o campo de LLMs que é a Engenharia de Prompt.

Pesquisas mostraram que diferentes estruturas de *prompts* podem influenciar de forma relevante a resposta de um modelo, destacando a importância do desenho da entrada para saídas acuradas e apropriadas ao seu contexto [Brown 2020]. O desenvolvimento de técnicas de engenharia de *prompts* levaram a exploração de estratégias de *zero-shot* e *few-shot learning*, por meio das quais LLMs podem executar tarefas para que não foram explicitamente treinados [Radford *et al.* 2019].

No aprendizado *zero-shot*, o modelo recebe somente uma descrição da tarefa que deve executar, devendo gerar saídas sem qualquer exemplo fornecido. Dessa forma, em tal aprendizado, observa-se a habilidade do modelo em executar o tipo de tarefa unicamente pelo seu conhecimento preexistente.

Enquanto que no aprendizado de *few-shot*, o texto de entrada possui alguns poucos exemplos - na ordem de até poucas dezenas. Esses exemplos demonstram as saídas esperadas para cada entrada informada no *prompt*. Nos estudos quanto ao tema, observou-se uma melhora significativa com a utilização de técnicas do aprendizado *few-shot* em relação ao *zero-shot* associados aos LLMs [Brown 2020].

Embora os enunciados dos problemas não forneçam códigos-resposta como orienta as técnicas de aprendizado *few-shot*, o formato em que são apresentados com entradas para os códigos e respectivas saídas esperadas sugere a importância do tratamento de *prompts*.

A seguir, utilizando como exemplo o problema I do caderno, observa-se o formato do enunciado:

"Você se encontra em um misterioso mundo binário, onde um vetor de N dígitos binários aguarda pelo seu exame minucioso. Cada dígito é zero ou um, criando um padrão único em toda a paisagem. Sua missão é descobrir os padrões ocultos deste reino binário, desvendando o significado de subvetores com um número ímpar de uns. O vetor de dígitos é denotado como $b_1, b_2, \dots, b_N$. Sua tarefa é embarcar em uma jornada para descobrir os subvetores enigmáticos – segmentos de dígitos consecutivos – e determinar a contagem de subvetores que abrigam um número ímpar de uns. Ao percorrer essa paisagem binária, lembre-se de que um subvetor é definido por seus dígitos iniciais e finais. Por exemplo, na sequência $[b_1, b_2, b_3]$, os subvetores incluem $[b_1]$, $[b_2]$, $[b_3]$, $[b_1, b_2]$, $[b_2, b_3]$, e $[b_1, b_2, b_3]$. Sua missão é projetar um algoritmo que determine o número total de subvetores contendo um número ímpar de uns nesta sequência binária. Não se esqueça de que a resposta pode não caber em um número inteiro de 32 bits.

Entrada: A primeira linha da entrada contém o inteiro N ($1 \leq N \leq 105$), representando o comprimento da sequência binária. A segunda linha contém os N dígitos binários $b_1, b_2, \dots, b_N$ ($b_i \in \{0, 1\}$), representando os elementos da sequência.

Saída: Seu programa deve imprimir uma linha contendo a quantidade de subvetores contendo uma quantidade ímpar de uns.

Percebe-se que, ao final do enunciado, há dois exemplos de entradas e respectivas saídas esperadas. Visto a importância da formatação desses exemplos, prepararam-se cuidadosamente todos os *prompts* para que entradas e saídas fornecidas tivessem valores e quebras-de-linha exatamente conforme o esperado pelas regras da maratona.

Figura 1 - Exemplos de entradas e saídas Problema I da Maratona de Programação SBC 2023.

Exemplo de entrada 1	Exemplo de saída 1
3 0 1 0	4
Exemplo de entrada 2	Exemplo de saída 2
10 1 0 0 1 1 0 1 1 1 0	30

Modelos

Para inferência dos códigos, escolheram-se dois modelos de linguagem de grande escala: o GPT-4o e LLaMa3.1 de 70 bilhões de parâmetros. Dois foram os motivos para seleção desses LLMs. O primeiro por serem modelos estado

da arte no período do estudo. O segundo foi com objetivo de comparação de desempenho entre um comercializado, caso do da OpenAI e um *open-source* desenvolvido pela companhia Meta Platforms, Inc.

Conforme consta na plataforma da OpenAI [OpenAI 2024b] o GPT-4o é o modelo mais avançado da empresa, que possui capacidade de inferência ainda comparável a do anterior GPT-4 Turbo. Entretanto, possui um desempenho melhor em outras linguagens que não o inglês que seus predecessores - o que é muito útil para o estudo, visto que os enunciados são em português. Além disso, é mais eficiente e menos custoso.

Quanto ao modelo *open-source* escolhido, o LLaMa 3.1 possui 3 versões treinadas com 8, 70 e 405 bilhões de parâmetros [Meta 2024c]. Também são os modelos mais recentes lançados pela Meta e com melhores performances nos *benchmarks* demonstrados no descritivo do modelo no GitHub.

Ao se consultar comparações de desempenho entre LLMs [OpenAI 2024a], nota-se que é a versão de 405 bilhões de parâmetros a qual é usualmente equiparada ao GPT-4o. Contudo, não foram encontradas alternativas livres de custo para execução do modelo em servidores na nuvem. Além disso, a máquina utilizada para o estudo não possuía capacidade suficiente de processamento para execução da versão disponível para *download* do modelo. A versão de 70 bilhões de parâmetros, todavia, possuía um meio de utilização disponibilizado pela companhia Groq, Inc [Meta 2024b] e foi por esse motivo selecionada.

Avaliação de modelos

Diferentemente de outros tipos de tarefa, avaliar o desempenho de um LLM ao performar tarefas de geração de código pode seguir uma abordagem objetiva. Por meio da implementação de testes, é possível afirmar que um código executa corretamente se as saídas obtidas são idênticas às esperadas - explicitadas pelos testes.

Dessa forma, como parte da avaliação, é suficiente utilizar os arquivos que já são disponibilizados no site da maratona. Para cada problema, uma quantidade de testes que varia de 15 (problema C) a 100 (problema M) são fornecidos. Os exemplos do enunciado, estão incluídos no total de casos de testes, ou seja, se o enunciado do problema C contém 2 exemplos de entrada e saída,

são disponibilizados 15 testes, dentre os quais 13 são casos que o modelo não foi explicitamente orientado a resolver.

Associada à utilização desses casos de testes, o presente estudo irá calcular a métrica $pass@k$ [Chen *et al.* 2021] para avaliar a efetividade dos modelos performando as tarefas enunciadas. Definida pela equação 1, essa métrica oferece um estimador não-viesado da acurácia do modelo ao executar um conjunto de tarefas.

$$pass@k := \mathbb{E}_{Problemas} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right] \quad (1)$$

Para calcular essa métrica, gera-se uma quantidade $n \geq k$ de amostras por problema, avalia-se o montante de corretas c para cada um deles e calcula-se, a um determinado k , a expressão interna ao valor esperado. O trecho de código definido no Listing 1 demonstra uma função numericamente estável para determinar essa expressão.

Listing 1. Função numericamente estável para cálculo do estimador $pass@k$.

```
1 def pass_at_k(n, c, k):  
2     if n - c < k:  
3         return 1.0  
4     return 1.0 - np.prod(1.0 - k / np.arange(n - c + 1, n + 1))
```

Após determinar valor da expressão para cada um dos problemas, determina-se a média e atribui-se à métrica $pass@k$ para a configuração de modelo e problemas.

Análise e Discussão de Resultados

Seguindo os passos já previamente detalhados e novamente enumerados abaixo, obteve-se a métrica $pass@k$ para cada configuração modelo-problemas. São os seguintes:

- Preparação de arquivos de teste;
- Preparação dos prompts;
- Inferência;

- Teste do código inferido com os testes do item 1;
- Avaliação do modelo.

Os itens de 3 a 5 foram executados para cada uma das configurações, enquanto os resultados dos dois primeiros foram reutilizados. Para inferência, os modelos foram requisitados com hiper-parâmetros na sua configuração original com exceção da *temperature* que foi explicitada para 1 no LLaMa, assim como é utilizada no GPT-4o. Houve 5 execuções por problema, o que considerando 13 enunciados para a edição, resultou em 170 inferências no total (85 por modelo).

A cada processamento de uma entrada-teste, salvou-se em um dos 85 arquivos de saída, uma linha com três possíveis valores: PASSED, FAILED e TIMEOUT. Esse terceiro foi utilizado considerando um limite máximo de 8 segundos por execução, tempo que coincide com o maior valor estabelecido nas informações da maratona (problema J).

Para exemplificação das saídas nessas etapas, considere o problema I enunciado anteriormente em 2.3. Utilizando o modelo GPT-4o, obtiveram-se como saídas, dentre as cinco para o modelo, os exemplos no Listing 2 e no Listing 3:

Listing 2. Saída I 1 com GPT4o.

```

1  ```python
2  import sys
3  input = sys.stdin.read
4
5  data = input().split()
6  N = int(data[0])
7  binary_sequence = [int(x) for x in data[1:]]
8
9  count_odd = 0
10 current_count = 0
11 prefix_counts = {0: 1}
12
13 for bit in binary_sequence:
14     current_count ^= bit
15     if current_count in prefix_counts:
16         count_odd += prefix_counts[current_count]
17         prefix_counts[current_count] += 1
18     else:
19         prefix_counts[current_count] = 1
20
21 print(count_odd)
22  ```

```

Listing 3. Saída I 2 com GPT4o.

```
1  ```python
2  import sys
3  input = sys.stdin.read
4  data = input().split()
5
6  N = int(data[0])
7  b = list(map(int, data[1:]))
8
9  odd_count = 0
10 even_count = 1
11 current_sum = 0
12 total_odd_subvectors = 0
13
14 for bit in b:
15     current_sum += bit
16     if current_sum % 2 == 0:
17         total_odd_subvectors += odd_count
18         even_count += 1
19     else:
20         total_odd_subvectors += even_count
21         odd_count += 1
22
23 print(total_odd_subvectors)
24 ```
```

Utilizando os dois códigos para executar os 21 testes de entrada e comparar saídas obtidas com as esperadas para o problema, observou-se um comportamento completamente diferente dentre as amostras. Para o primeiro, todos os testes falharam, enquanto que para o segundo, todos executaram com sucesso.

Por outro lado, em outros casos, obtiveram-se resultados mais heterogêneos para os testes, com *timeouts* dispersos entre falhas e sucessos, por exemplo. Ainda que para algumas das amostras houvesse alguns *timeouts* dentre somente sucessos, indicando provavelmente corretude do código, não se considerou tal inferência como correta durante o cálculo da métrica. Como exemplo, para o problema E, uma das saídas ao se utilizar o modelo LLaMa3.1:70b foi o código no Listing 4. Para esse caso, obtiveram-se 90,9% de sucessos contra 9,1% de *timeouts*.

Listing 4. Saída para E 1 com Llama3.1:70b.

```
1  ```python
2  import heapq
3
4  n, k = map(int, input().split())
5  flores = list(map(int, input().split()))
6
7  flores_invertidas = [(-x, x) for x in flores]
8  heapq.heapify(flores_invertidas)
9
10 for _ in range(k - 1):
11     _, mais_polens = heapq.heappop(flores_invertidas)
12     if mais_polens == 0:
13         break
14     mais_polens -= sum(map(int, str(mais_polens)))
15     heapq.heappush(flores_invertidas, (-mais_polens, mais_polens))
16
17 if flores_invertidas:
18     _, mais_polens = heapq.heappop(flores_invertidas)
19     polens_coletados = sum(map(int, str(mais_polens)))
20 else:
21     polens_coletados = 0
22
23 print(polens_coletados)
24  ```
```

Ao fim das inferências e execução dos testes, resultou-se em 170 arquivos com registros de sucesso, falha e tempo estourado linha a linha. Esse conjunto de dados foi utilizado para cálculo das métricas em 3 situações por modelo: $k = 1$, $k = 3$ e $k = 5$. A tabela 1 mostra o resumo dos resultados

Tabela 1 - Resumo das métricas de desempenho pass@k.

Métrica	gpt-4o	llama3.1
pass@1	24.62%	13.85%
pass@3	30.00%	15.38%
pass@5	30.77%	15.38%

Pelos dados apresentados, o modelo GPT-4o teve notadamente um desempenho superior a versão do LLaMa utilizada e será, portanto, utilizado para comparação com as estatísticas da Maratona de Programação SBC 2023. É importante ressaltar que o raciocínio para comparação a seguir não é justo, visto que a competição acontece em um ambiente com critérios mais rigorosos. O objetivo é apenas ilustrar que o desempenho do modelo foi considerável.

No *website* da maratona [SBC 2023b], constam 663 equipes que pontuaram na competição. Aqueles que submeteram com sucesso 4 de 13 tarefas obtiveram

pontuação de 30,77%, valor igual ao da melhor métrica obtida. Dos que alcançaram essa quantidade, a primeira aparição está na 104ª colocação. Dessa forma, caso estivesse sob os mesmos critérios de avaliação, o modelo figuraria numa posição igual ou superior que 84,62% das equipes.

Diante dos resultados, é importante ressaltar alguns pontos que poderiam trazer melhores métricas em novas configuração de execução das tarefas. Um primeiro ponto é encontrar um meio acessível para uso da versão de 405 bilhões de parâmetros do LLaMa

Isso porque, em alguns *benchmarks* de geração de código [Meta 2024a], o modelo conseguiu ter desempenho superior ao GPT-4o. Além dessa versão, modelos desenvolvidos por outras instituições podem ser testados com a finalidade de tentar obter melhores desempenhos.

Outro ponto interessante para destaque é que em algumas tarefas, ambos os modelos falharam em executar os próprios casos de testes fornecidos no enunciado. Dessa forma, poder-se-ia implementar uma configuração de retentativa em formato de *chat* continuado ou mesmo descartando a resposta e iniciando uma nova inferência. Assim, estabelecendo uma quantidade máxima de retentativas, possivelmente mais códigos corretos seriam gerados.

CONCLUSÃO

Os resultados obtidos nesta pesquisa demonstram que modelos de linguagem de grande escala (LLMs) possuem um potencial significativo para a geração automática de código em desafios de programação, como aqueles enunciados na Maratona de Programação da Sociedade Brasileira de Computação (SBC).

A aplicação dos modelos GPT-4o e LLaMa 3.1 apresentou desempenho considerável, ao se considerar acurácia medida pela métrica *pass@k*. Embora ambos os modelos tenham conseguido gerar soluções corretas para um número significativo de problemas, foi observado que o GPT-4o se destacou na comparação das métricas para todos valores de *k*.

A escolha do Python como linguagem de inferência mostrou-se adequada, tanto por sua familiaridade nos treinamentos dos modelos quanto pela facilidade de manipulação dos dados e execução dos testes. Além disso, a metodologia adotada para a preparação de *prompt* e avaliação dos resultados, utilizando

casos de testes e a métrica *pass@k*, foi eficiente para determinar a capacidade dos modelos em gerar códigos funcionais e corretos.

Entretanto, o estudo também identificou limitações importantes para destaque. A não consideração de aspectos como tempo de execução, uso de memória e outros critérios de desempenho refletiu em uma avaliação menos rigorosa do que a aferida pela comissão do evento.

Adicionalmente, visto que houve uma quantidade significativa de problemas não resolvidos pelos modelos, sugere-se a exploração mais ampla de diferentes LLMs e maior variação de parâmetros para estudos futuros. Outra alternativa, é a elaboração de outras configurações para tentativa de respostas que falharam nas entradas enunciadas. Dentro dessa estratégia ou mesmo no *prompt* inicial, é também sugerida uma maior exploração de outros formatos de entrada, considerando técnicas mais sofisticadas de engenharia de *prompt*.

Em suma, a pesquisa corrobora a hipótese de que LLMs podem ser uma ferramenta muito útil para a resolução de problemas de programação em competições como a maratona. A continuidade deste estudo, para aprimoramento do desempenho em desafios competitivos e mesmo ampliando o escopo para contemplar desafios de maior complexidade, poderá oferecer percepções valiosas sobre a aplicabilidade dos LLMs na geração de código em seu âmbito geral.

REFERÊNCIAS

- Austin, J., Odena, A., and *et al.* (2021). Program synthesis with large language models. *CoRR*, abs/2108.07732.
- Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint ar- Xiv:2005.14165*.
- Chen, M., Tworek, J., and *et al.* (2021). Evaluating large language models trained on code.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436– 444.
- Meta (2024a). Apresentação do modelo llama 3.1 e versões no *website* da meta platforms, inc. <https://ai.meta.com/blog/meta-llama-3-1/>. Acesso: 28 ago. 2024.
- Meta (2024b). Receitas de engenharia de *Prompt* utilizando llama3.1 disponibilizada no repositório github do modelo. https://github.com/meta-llama/llama-recipes/blob/main/recipes/quickstart/Prompt_Engineering_with_Llama_3.ipynb. Acesso: 28 ago. 2024.
- Meta (2024c). *Model card* do modelo llama 3.1 da meta platforms, inc. https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md. Acesso: 28 ago. 2024.

Muthukrishnan, N., Maleki, F., and *et al.* (2020). Brief history of artificial intelligence. *Neuroimaging Clinics of North America*, 30(4):393–399. Machine Learning and Other Artificial Intelligence Applications.

OpenAI (2024a). Comparativo de modelos estado da arte disponibilizado pela openai. <https://openai.com/index/hello-gpt-4o/>. Acesso: 28 ago. 2024.

OpenAI (2024b). Modelo gpt-4o da openai. <https://platform.openai.com/docs/models/gpt-4o>. Acesso: 28 ago. 2024.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.* (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

SBC (2023a). Maratona de programação da sociedade brasileira de computação - primeira fase 2023. <https://maratona.sbc.org.br/hist/2023/primfase23/>. Acesso: 28 ago. 2024.

SBC (2023b). Tabela de pontuação das equipes participantes da maratona de programação sbc 2023. <https://maratona.sbc.org.br/hist/2023/primfase23/contest/Score.html>. Acesso: 28 ago. 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Zhang, F., Chen, B., Zhang, Y., Keung, J., Liu, J., Zan, D., Mao, Y., Lou, J.-G., and Chen, W. (2023). Repocoder: Repository-level code completion through iterative retrieval and generation.

INSTANCE SEGMENTATION ALGORITHM FOR DETECTING AND CLASSIFYING DEFECTS IN AIRCRAFT FUSELAGE SKINS WITH CONVOLUTIONAL NEURAL NETWORKS

Juan Pedro Baena Cassal

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

juanpedro.bc22@gmail.com

Elcio Hideiti Shiguemori

Computer Science Division - Instituto Tecnológico de Aeronáutica (ITA)

ABSTRACT

The use of computer vision in visual inspection automated to identify nonconformities in the Manufacturing and Maintenance of products has become well established in various industrial sectors. When employing Deep Learning techniques with Computer Vision, these sectors have achieved remarkable results, continually improving quality and reducing inspection times compared to traditional manual methods. However, some industries, such as Aeronautics faces challenges in implementing these technologies due to the critical nature of their final products. This study addressed the specific challenges of applying such technologies for product inspections in the Aerospace Industry. To this end, experiments were developed using state-of-the-art algorithms with convolutional neural networks to create instance segmentation models to detect and classify nonconformities in aircraft fuselage, with the aim of understanding and comparing the results obtained with the application to a set of data created for the research, containing cracks, dents, scratches and dirt on aircraft fuselages. The analyzes showed good results for the task that analyzed the performance of models created with YOLOv8 and YOLOv9, identifying a better result from the latter and evidence of points of common difficulties such as the detection of small defects.

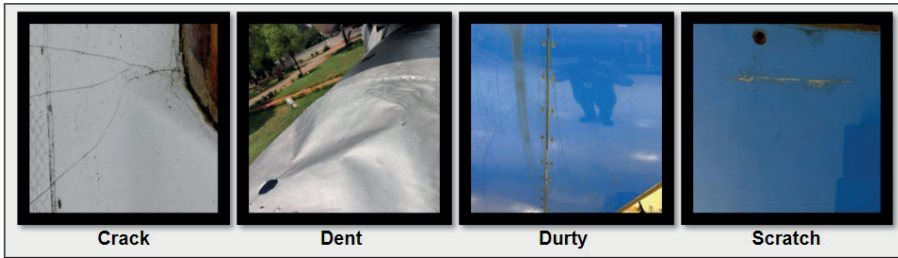
INTRODUCTION

In the aeronautical industry, where precision is paramount due to the critical nature of the final product[1], detecting surface defects is one of the most crucial tasks in aircraft maintenance as it directly impacts the quality of the fuselage, which is vital for the aircraft's stealth performance[2]. Similarly, in aircraft manufacturing, visual inspection ensures that product requirements are met before delivery to customers. Indeed, the visual inspection process is essential for controlling the quality of finished products[3] and ensure the safe operation of aircraft in their future activities.

With the increasing demand for product quality and documentation across many industrial sectors, automating visual inspections through machine vision has become a key technology[4]. This automation, supported by the high volume of data accessible via visual equipment, enables quick detection and flagging of defective goods, recognizing their defects, and allowing swift and efficient intervention in Industry 4.0[5], enabling increased quality and greater flow of inspection cadences. Currently, advanced artificial intelligence techniques, particularly Deep Learning, are proving to be more efficient than traditional techniques in the field of computer vision, pushing the limits of digital image processing[6].

This innovative approach to visual inspections using computer vision can significantly impact aircraft production and maintenance operations[1]. Manual inspection of fuselages presents particular challenges, depend on the expertise of inspectors, often cause safety hazards to inspectors due to the height of the aircraft, it requires a high level of constant focus for long periods, due to the dimensions of the aircraft, leading to cognitive fatigue and potential errors which can have a high cost for companies[7]. Automating the fuselage inspection process can be extremely difficult to approve due to the criticality of the process, however the use of automation as a support tool can be an effective solution to overcome the challenges of current processes, performing image analysis to recognize unwanted characteristics and objects, overcoming environmental constraints[8], reducing failures, and even mitigating safety risks by using robots or drones to acquire images in hazardous areas.

Figure 1 - Nonconformities in aircraft fuselage.



In this study, images of defects in aircraft fuselage were collected, characteristics in four distinct categories: cracks, scratches, dents and dirt. These images were used to train convolutional neural networks based on the latest versions of the YOLO architecture, highly recognized for good performance in inspection tasks due to their high accuracy, excellent real-time performance and efficient use of computational resources[9]. This architecture also allows the training of models for instance segmentation, an advanced technique that allows detecting, segmenting, identifying and locating objects simultaneously, which is ideal for locating defects accurately[10], considering the different formats within each class. The Figure 1 presents examples of the nonconformities analyzed in this study.

Conceptually, dirt refers to contaminants or undesirable particles that accumulate on a surface. It is considered the least severe non-conformity and has the simplest resolution[11]. A scratch, on the other hand, is a surface imperfection characterized as a mark or groove caused by a harder object coming into contact with a softer surface[12]. When the painted surface becomes chipped due to scratching, the loss of the protective coating allows the base metal to become more susceptible to corrosion[11]. Conversely, cracks are superficial fissures that develop in a material and have their initiation and growth influenced by propagation energy derived from mechanical, thermal or chemical effects[13]. Cracks have the potential to compromise the strength and durability of the material, making them one of the most serious non-conformities due to the direct risks they pose to the operational safety of aircraft[14]. For this study, dents are considered concave and convex deformations in the fuselage structure that can cause weaknesses and impair aerodynamics.

For the aeronautics industry, the primary reference document characterizing such nonconformities is the Air Transport Association of America Specification 100

(ATA-100)[15]. This standard is used in the aviation industry to organize and classify technical information about aircraft and components, standardizing maintenance processes. According to this document, Chapter 51 - Standard Practices and Structures General, inspired by the Luftfahrt Bundesamt (German Federal Aviation Authority), aircraft damage is classified from Class 1 to Class 4, with Class 1 being the most severe. Depending on their severity, cracks can be classified as Classes 1 to 3, while scratches fall under Class 4. In the same document, the impact of dirt is not mentioned, however in the context of inspections, according to FAA Advisory Circular AC 43-4B - Federal Aviation Administration, a US regulatory agency, before inspections it is necessary to clean the surfaces that will be inspected, this is important to remove any type of dirt that could hide non-conformities[11], which is why dirt is also considered in this work as a class of non-conformity.

RELATED WORK

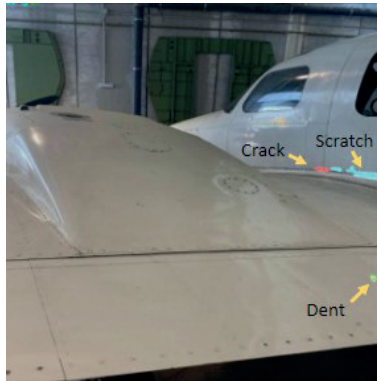
Recently, a lot of work is being carried out in the area of detecting and classifying defects on product surfaces, which can be diverse, not only for aeronautical products, but the techniques and approaches are similar with customizations to suit each case. For example, Wang *et al.* 2022[16] carried out work on detecting surface defects in aircraft glass covers based on improved YOLOv4 (You Only Look Once, version 4), a real-time object detection algorithm that uses deep neural networks to identify and localize objects in images or videos with high accuracy and speed, in this case it was adapted to use FReLU - Flexible Rectified Linear Units activation function[17], which can adaptively adjust the ReLU[18] output by a rectified point to capture negative information and provide zero-like property, instead of Mish[19], a self-regularized non-monotonic activation function, to better adaptively capture spatial correlation, however, the version of YOLO used is already considered outdated as it still uses an anchor system, proving to be less efficient than its subsequent versions[20].

Liu *et al.* 2022[21] uses YOLO4 applied to an Unmanned Aerial Vehicle to carry out inspections on aircraft fuselage, due to the lack of real images of aircraft fuselage defects, the work proposed using a transfer learning, a machine learning technique in which a model trained on a task is reused or adapted to improve performance on a new task, taking advantage of previously acquired knowledge[22],

where, first, the network is pre-trained on a publicly available dataset of defects in similar materials and then, the head of the network is frozen and the classification model is adjusted using a dataset collected from real defects on fuselage surfaces, a strategy which achieved high accuracy according to the author. Rice *et al.*[23], using instance segmentation to identify and classify defects, projected defects into a 3D system containing their location on the aircraft, while Zhang *et al.* 2020[24], carried out the identification of defects in aircraft fuselages by comparing the results of YOLO version 1, which treats object detection as a single regression problem[25], Faster-RCNN with integrate region proposal networks (RPNs)[26] and Mask-RCNN, which extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition[27], as a result, the tests concluded that the prediction recognition accuracy of YOLO for skin crack, skin peeling, skin deformation and skin tear are 72%, 84%, 68% and 70%, results superior to those obtained by the other models studied.

Lan *et al.* 2023[28] specialized the improved YOLOv8 model, one of the last released versions of the YOLO architecture[29], for detecting defects in PCBs, the modification in the model architecture consisted of the addition of a Crossscale Fusion Module (CFM), added between the backbone and the neck of the network, parts that extract features and make final predictions, this module was used for the fusion of resources from three adjacent layers and thus achieved an improvement in mAP (mean Average Precision) performance, a metric used to evaluate the accuracy of detection models. Tao *et al.* 2023[30] performed semiconductor wafer detection using YOLOv8 and obtained superior results compared to Single Shot MultiBox Detector (SSD), a real-time object detection model that performs object detection in a single step, and models created with YOLOv5, fifth generation of YOLO architecture. The good result is due to an improvement attributed to the new features of YOLOv8, such as the C2f module, a specialized block that combines multiple convolutional layers with skip connections and Spatial Pyramid Pooling Fusion (SPPF), which enhances feature extraction by pooling input features at multiple scales. Zhu *et al.* 2023[31] modified the original architecture of YOLOv8 by adding a small object detection head and Convolutional Block Attention Module (CBAM), module that enhances feature representation by focusing on important parts of the input data and with this the proposed model achieved better utilization of multi-level resources to assist in detecting small defects.

Figure 2 - Small defects.



CHALLENGES OF CLASSIFYING NONCONFORMITIES IN AIRCRAFT FUSELAGE

Initially, the task of detecting and classifying nonconformities in aircraft fuselages is expected to include the following challenges mentioned in the following subitems.

Structuring the minimum Dataset for analysis

To begin studying the possibility of automation, one of the main challenges in analyzing nonconformities in aircraft fuselages is creating a dataset with a significant amount of data to allow a model to identify patterns. This challenge is due to the complexity and diverse nature of defect shapes, which present a high degree of variation and complexity across different target classes. Therefore, it is essential to map as many features as possible to provide enough data for a model to achieve good performance in identifying and classifying defects[32].

The difficulty in finding aircraft defect image datasets can be attributed to the critical nature of the product involved, making this type of image extremely sensitive. Public disclosure of such content can be extremely damaging to a company's image, as it can negatively influence passengers' perception of aircraft safety[33]. Passengers may begin to avoid certain aircraft if they are aware of the types of damage they have previously suffered, even if the aircraft has been fully repaired and met *all* necessary safety standards. Another reason is related to the potential use of these images for automation, which can be considered an industrial secret, as few companies have the potential of generating such

images, and using them in the development of technology designed to automate inspections, which can become a competitive advantage.

Identification of small defects

A challenge for current manual processes has been not to let small defects go unnoticed, which can also be a challenge for computer vision[34]. In the Figure 2 it is possible to observe a small defect that could go unnoticed by a tired eye during an inspection.

Figure 3 - Dirt vs Scratches.

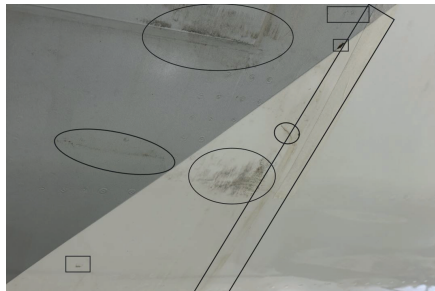


Figure 4 - Reflections/Glare vs Defects.



Delimitation of visibly similar classes

A significant challenge in classifying nonconformities in aircraft fuselages is distinguishing between classes that exhibit a high degree of similarity. For instance, certain types of scratches can be difficult to differentiate from cracks

and dirt. Although cracks often have branching patterns, in some cases, they can appear very similar to scratches, and both classes can be mistaken for dirt particles. In Figure 3, the circular shapes represent scratches, and the rectangular shapes represent dirt, both classes being very close to each other in the same location and displaying considerable visual similarities.

False identification of Glare as a defect

When capturing images of shiny surfaces, such as glass, metal, plastic, or even surfaces painted with glossy and reflective paints, as is common with most aircraft, excessive glare or unintended reflections can occur. These visual disturbances are commonly found in images of aircraft in hangars illuminated by strong artificial lights. During the images or videos capture, depending on the relative position of the light source, the camera, and the object, part of the light is reflected back into the camera, causing unwanted glare in the final image. In Figure 4, it is evident how difficult it is to distinguish between a defect in the fuselage and a reflection. In the image, the rectangular contours indicate unwanted reflections, while the circular contours mark scratches.

Figure 5 - Overlap of nonconformities due to dirt.

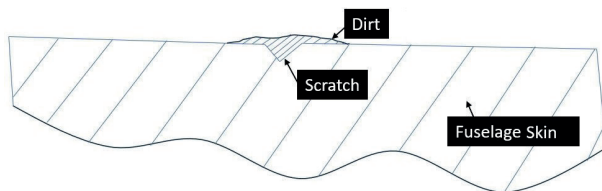
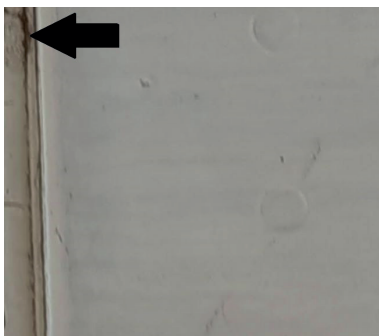


Figure 6 - Real case of overlap of nonconformities due to dirt.



Overlap of different types of nonconformities by dirt

Some nonconformities may be difficult to classify because they are overlaid by dirt debris. In certain situations, surfaces with scratches and cracks present these defects masked by a superficial layer of dirt, making it difficult to see deeper damage to the fuselage, as illustrated in Figure 5.

In Figure 6, a real case of nonconformity concealment due to a crust of debris covering the damage is depicted, making it impossible to identify or characterize the hidden nonconformity.

PROPOSED METHOD

The research followed the methodology illustrated in Figure 7. The process began with the preparation of a dataset, followed by a preprocessing stage to ensure the data was suitable for model training. Each model underwent a rigorous training process, after which the models were evaluated using images from the test set. Performance metrics were generated to assess the effectiveness of each model. Additionally, a detailed point-by-point evaluation was conducted on challenging cases to further analyze the performance of the best models.

Preparation of the Dataset

To collect a minimum volume of images that enable analysis, different sources were aggregated to form a single dataset containing the four classes of interest: Cracks, Dents (Convex and Concave), Scratches, and Dirt. As shown in Figure 8, the images in the final dataset were collected from three sources: public datasets found on the Roboflow platform[35], datasets purchased from the IEEE Data Port[36], and a small dataset assembled by acquiring images of defects on aircraft fuselages from public exhibitions. Due to the scarcity of images for the crack and dent classes, images of concrete cracks and car dents were added, comprising a total of 23% and 30% of these images in their respective classes where they were added.

For this research, images were acquired using the camera of an iPhone 14 Pro Max with an optical sensor configured to obtain images in the most compatible mode, which saves RGB channel images in JPEG format with dimensions of

4032x2027 pixels and a vertical and horizontal resolution of 72 dpi. As a capture methodology, the acquisition of 9 images per non-conformity was standardized, taken from different positions, with 3 variations of optical zoom (1x, 1.5x, and 2x) for each, generating 27 unique views of the same target, increasing the data diversity. Figure 9 illustrates nine red dots representing the positions where the camera performed the acquisitions, with distances between positions established using a digital laser tape measure with a precision of 2 mm.

Figure 7 - Experiments process.

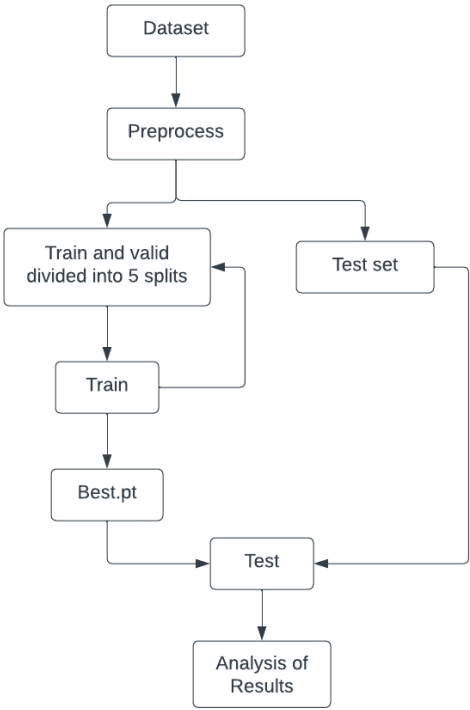
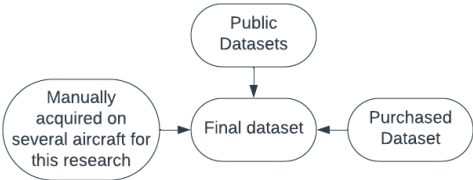


Figure 8 - Final Dataset Composition.



ed, 2190 images were collected, containing 1620 instances of scratches, 1616 of cracks, 1604 of Dirty and 1488 of Dent.

Data Labelling

For the creation of a model capable of recognizing and classifying non-conformities, a supervised machine learning methodology will be used for class identification. In this type of approach, the target element contained in an image of various elements must be indicated. This procedure is known as labeling and involves using a framework to delineate the boundaries of the target in the images to obtain coordinates of where the element of interest is located within the analysis image[37].

Figure 9 - Positioning for image acquisition.

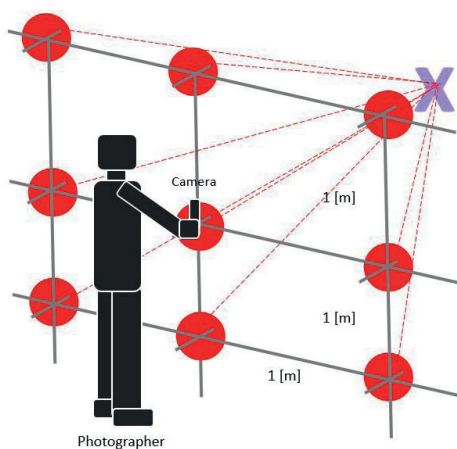
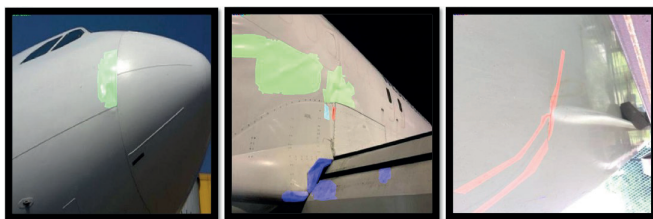


Figure 10 - Example of labeling in images.



To obtain the location of defects of different classes, instance segmentation was utilized for identification and classification, a technique that provides distinct labels for separate instances of objects belonging to the same class and assigns a unique mask to each specific instance[38]. This technique is particularly useful

in scenarios where it is necessary not only to identify object classes but also to count and delineate individual instances.

As a labeling framework, the Roboflow platform[39] was utilized due to its ability to allow users to have datasets in private mode, quickly merge datasets covering the same theme available on the platform, export to YOLO format, easy access to various types of preprocessing techniques such as data augmentations, used to artificially increase the size and diversity of a training dataset and tools for smart labeling, facilitating the delineation of boundaries of complex shapes. This is a significant advantage for reducing the time required for labeling large quantities of images. Figure 10 displays defect labeling on aircraft fuselage images.

As a result of the labeling, annotation files were generated for each image in YOLOv8 format and compatible with YOLOv9, containing for each line the coordinates of the delimitations of a polygon that demarcates, in turn, a target object within the respective image, to which, the annotation file belongs, and each file can have n lines, each one for an instance that can belong to any of the 4 classes of interest. The Equation 1 represents the format of an annotation, where X is the class number and x and y are the coordinates in the image of the points that form the n-sided polygon.

Figure 11 - Augmentation techniques used. Table 1. Instance and image count per class.

$$x\ x1\ y1\ x2\ y2\ x3\ y3\ \dots\ xn\ yn\ (1).$$

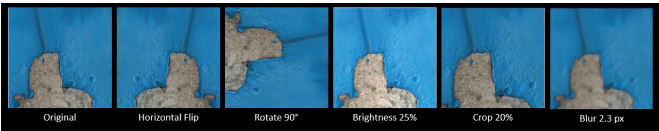
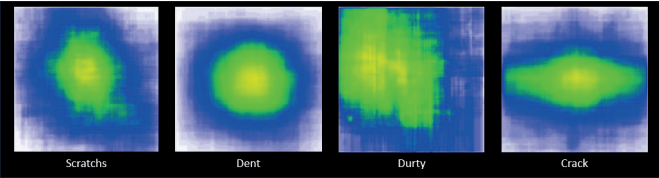


Figure 12 - Heatmap of the average location of classes in images.



Data Augmentation

Data augmentation is a strategy used in the field of machine learning, especially in activities related to computer vision, with the purpose of expanding the

amount of data available for training a model[40]. For this work, new images were created using 5 augmentation approaches, horizontal flip, rotation by 90°, with an increase in brightness by 25%, taking into account the heatmap of the locations of the targets in the images, Crop of 20% of the edges was applied. of the images and finally, 2.5px blurring was applied to simulate images with low resolutions.

For each new image, it was not necessary to carry out new labeling, only the original coordinates of each polygon were transformed into the new images created through basic geometric conversions. This strategy avoided hours dedicated to labeling the new images created. In the Figure 11 it is possible to check each of the augmentation techniques used.

After the data augmentation process, the generated dataset had 8515 images. The Table 1 presents the quantitative results generated by listing defect instances by class and by images.

Analysis of the final dataset

As a final result, it is possible to observe a slightly unbalanced dataset with the Dents class containing a little less representation. In the Figure 12 it is possible to observe the heatmap of the location of defects in the images, as a result, it can be concluded that the majority of defects are located more towards the center of the images, with the exception of the Dirt class, this enabled the use of data augmentation techniques such as cropping at the edges as it is known that there would be no significant loss of information when removing the outermost parts of the images.

Regarding the proportions of the images in the dataset, 1710 images have a square shape, 459 a wide format, 18 tall format and the rest ultra wide. As shown in Figure 9, it is also possible to notice that there is a concentration with the majority of images close to 640px times 640px and for this reason, this range is the most suitable dimension for standardizing the size of images.

Figure 13 - Image Ratios and sizes.



Through a mapping of image sizes, it was observed that there is an average image size of 0.41 mp, varying from 0.11 mp to 12.19 mp, which indicates that the images, on average, have a relatively low resolution and a wide range of resolution diversity. This diversification can be beneficial for machine learning applications because exposing models to a variety of resolutions during training can help with generalization by incorporating learning from images with fewer and more details into the model. The second consideration is that there is a large variation in the size of defects, this favors The detection and classification model learns to identify nonconformities at multiple scales.

Data preprocessing

Pre-processing aimed to prepare the images from the final dataset to be better used in the training process. To achieve this, the steps mentioned below were carried out:

- **Data Cleaning:** As a first approach to pre-processing, it was checked whether the image labeling was correct, considering that many of the images were acquired with labeling, cases were eliminated that were

judged as not pertinent to the research focus, duplicate images were removed and class names were standardized so that there were no name variations within the same classes.

- **Balancing:** Originally in the dataset there were a much larger number of instances of scratches and dirty, so to balance the number of instances of each class the Under-sampling approach was used to reduce the number of images which contained these two classes so that there was an approximate leveling in quantity with cracks and dents.
- **Data Transformation:** Many Deep Learning models, especially convolutional neural net-works (CNNs), are designed to work with fixed-size inputs. This way, the size of the images was standardized to 640x640px to allow the training of these models in a consistent way. In addition to standardizing the size, the images were self-oriented to discard the rotations identified in the data saved from the Exchangeable Image File Format EXIF and standardize the order of the pixels.

Technique and Pre-trained model Selection

To address the task of identifying and classifying non-conformities (NCs), instance segmentation was chosen as the core technique. Instance segmentation is advantageous for defect detection as it not only identifies and classifies each defect but also accurately maps its precise shape and boundaries, providing detailed information that is crucial for effective quality control.

Given the demands of this technique, the YOLOv8 and YOLOv9 algorithms were selected due to their good efficiency in performing instance segmentation. These models are known for their ability to detect objects quickly and accurately, making them ideal for the complex task of defect segmentation and classification in diverse industrial contexts. Its advanced architecture allows location and segmentation of defects, with the expectation that even small and intricate non-conformities are detected and categorized correctly.

The YOLO (You Only Look Once) network is currently considered as state of the art for computer vision, Unlike other region-based two-stage detection models such as Fast RCNN[41], which is an object detection model capable of improving on previous R-CNN approaches by integrating region proposal generation and classification into a single network and its evolution, the Faster RCNN[42], in this

way, detection is treated as a regression problem, which brings great benefits to real-time processing[43].

The trajectory of this algorithm until reaching the YOLOv8 and v9 versions began in 2016 with the generation of YOLOv1[44], which performed better than traditional models of the time, for example, the models made with the SSD - Single Shot MultiBox Detector algorithm[45], but it still did not perform well for detecting objects close to each other and very small groups., in 2017 the YOLOv2 version[46] sought to improve training and increase performance, introducing Overfeat and SSD (Single Shot Detector), in addition to adopting a fully convolutional model with Anchors that trains with complete and multi-scale images, YOLOv2 also, similar to Faster R-CNN, adjusts the references in the bounding boxes instead of directly predicting the width and height, however, despite these modifications, the model still predicts the x and y coordinates directly. In 2018, the YOLOv3[47] version was released, similar to YOLOv2 with some improvements that made it possible to achieve multiscale fusion through a pyramidal network structure, resulting in an even greater improvement in the detection of small targets. In 2020, YOLOv4 was launched[48], for the first time without original author J. Redmon and included the YOLOv3 header in the network and improvements in several aspects, which made it possible to improve the detection performance of targets of different scales. In the same year, Ultralytics' landmark YOLOv5 was launched with superior performance to YOLOv4, YOLOv5 improved the use of trained neural networks in video processing applications, in addition to offering a wide spectrum of architectures with different levels of complexity[49]. Subsequently, YOLOv6[50] and YOLOv7[51] were developed through several improvements and optimizations at an algorithmic level, YOLOv7 has surpassed all known object detectors in terms of speed, covering a speed range from 5 to 160 FPS, with the higher the FPS, the faster the model is capable of process and detect objects in real time. In 2023, YOLOv8[52] was launched by the company Ultralytics, which had also launched YOLOv5, the new model being superior in performance to the previous ones and offering five scaled versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) and YOLOv8x (extra-large), such sizes offer varying balances between speed and accuracy. Smaller models are faster and lighter, ideal for real-time use on limited hardware. Larger models provide greater accuracy but require more computing power. The algorithm

also supports multiple vision tasks such as object detection, segmentation, pose estimation, tracking and classification[53]. The Table 2 shows the characteristics of each model, being:

- **Depth Multiple (d):** The parameter adjusts the depth of the neural network by scaling the number of layers in the architecture. A smaller d results in a shallower network with fewer layers, which typically leads to faster training and inference, but can reduce the model's ability to learn complex features. On the other hand, a larger d creates a deeper network with more layers, increasing the model's ability to capture intricate patterns in the data, but requiring more computational resources and potentially increasing the risk of overfitting. The choice of d must balance computational efficiency with the complexity of the task in question.

Table 2 - Model Specification.

Model	d (depth multiple)	w (width multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

- **Width Multiple (w):** This parameter controls the width of the network, that is, the number of channels in each layer. It is multiplied by the default number of channels to get the actual number of channels in each layer. A smaller w results in a narrower network, while a larger w results in a wider network.
- **Ratio (r):** This parameter adjusts the balance between the network's width (number of channels) and depth (number of layers). A larger r increases the network's width relative to its depth, resulting in more channels per layer, which can enhance the model's ability to capture fine details and process more information at each layer. However, this also increases the computational load. Conversely, a smaller r makes the network deeper with fewer channels, which might improve the ability to learn hierarchical features but may also limit the model's capacity to capture broad, detailed patterns. The choice of r should be tailored to the specific needs of the task and the available computational resources.

In 2024 YOLOv9[54] was launched, that version incorporated innovative techniques such as Programmable Gradient Information (PGI), a technique in neural networks where the gradient information, which is used for updating the model parameters during training, is adjusted or programmed in a way that enhances the learning process and the Generalized Efficient Layer Aggregation Network (GELAN), a neural network architecture designed to enhance the learning process by efficiently aggregating information across multiple layers, such architectural changes promoted an improvement in performance in this new generation.

Training of the neural network

Training was carried out with the four pre-trained YOLOv8 models[55] of different types of complexity mentioned below:

- YOLOv8n-seg
- YOLOv8s-seg
- YOLOv8m-seg
- YOLOv8l-seg

For YOLOv9, training was carried out with the pre-trained model YOLOv9c-seg[56]. All training was conducted with the aim of specializing a network for the task of instance segmentation. For this purpose, several aspects and hyperparameters were standardized to maintain comparative coherence between the model results, including the batch size of 16 images, staying within the memory limits of the GPU used. Patience of 30 epochs before Early Stopping for all training, ensuring that the model doesn't stop training too early and miss out on potential improvements, data augmentation was disabled and other advanced techniques were applied, such as Automatic Mixed Precision (AMP) to reduce memory usage and increase computational efficiency while maintaining model accuracy.

The training process involved the analysis of a dataset consisting of 8300 images for training and 340 images for validation. The model was configured to perform segmentation into 4 specific classes, so the specific configuration of the layers, such as convolutions and transformation blocks, were adjusted to meet the requirements of the segmentation task.

SGD was used as an optimizer, an abbreviation for Stochastic Gradient Descent[57], which is widely adopted in deep learning with convolutional neural networks[58]. Its application is especially relevant in large-scale problems, in which data sets can be extensive, and the calculation of gradients across the entire database can require prohibitive computational resources in terms of time and memory[59].

The training was carried out by a machine equipped with an NVIDIA GeForce RTX 4070 Ti GPU with 12 GB of dedicated memory, details are shown in the Table 3. During the process, automatic mixed precision checks (AMP) were performed, ensuring efficient and error-free execution.

Table 3 - Hardware and System Specifications.

Component	Specification
GPU	NVIDIA GeForce RTX 4070 ti
CPU	Intel Core(TM) i7-13700K 13th Gen, 8 cores, 16 threads, 3.5 GHz
RAM	32 GB DDR4, 3200 MHz
Operating System	Windows 11 Pro
Training System	Pytorch 2.1.0 with CUDA 118
Motherboard	ASUS Tuf Z690-plus D4 Gaming

Validation

To validate the training, the k-fold cross-validation method[60] was used, a robust and generalizable technique to evaluate the performance of machine learning models in each epoch, guiding the training direction, improves model evaluation by reducing variance and providing a more reliable performance estimate, ensures better use of data by testing the model on multiple subsets (splits). This technique involves dividing the data set into k subsets (or "folds") of equal size, maintaining the original distribution of classes in the data. The model is trained *K* times, each time using k-1 subsets as training data and the remaining subset as validation data[61]. This ensures that each observation is used for both training and validation, maximizing the use of the dataset and minimizing the possibility of bias or overfitting[62]. In the end, the results are calculated as the averages of the performance metrics obtained in each training and validation iteration. This approach is particularly useful when working with unbalanced data sets that are highly varied in terms of the complexity of instances within the same class, avoiding a split that could allocate easier instances to the validation subset, a common

problem in simpler validations such as holdout validation[62]. In the Table 4, one can observe the number of instances of each class, by division and by training and validation subsets, while the Table 5 presents this same division in terms of percentage, totaling the sum of the training, validation and test subsets at 100%.

Test

To test the final models, a subset of image and label defects was used, balanced with the 4 classes of interest that were collected in a stratified manner from the dataset formed after pre-processing and did not participate in the training processe, in this way, separating the testing set from the training process ensures an unbiased assessment of the model’s performance on unseen data, prevents overfitting, and accurately assesses the model’s ability to generalize to new data[63]. The Table 6 shows the number of instances and percentage of the 851 images that make up the test set.

Table 4 - Number of instances per class in the training and validation sets.

Split	Train				Validation			
	Crack	Scratch	Dent	Dirty	Crack	Scratch	Dent	Dirty
Split 1	4022	4536	3993	4195	970	857	933	1150
Split 2	4007	4341	3857	4302	985	1052	1069	1043
Split 3	4002	4311	3968	4283	990	1082	958	1062
Split 4	3995	4120	3987	4307	997	1273	939	1038
Split 5	3942	4264	3899	4293	1050	1129	1027	1052

Table 5 - Percentage of instances per class in the training and validation sets.

Split	Train (%)				Validation (%)			
	Crack	Scratch	Dent	Dirty	Crack	Scratch	Dent	Dirty
Split 1	68.83	77.24	73.28	84.48	13.40	14.35	16.76	15.22
Split 2	68.19	77.64	71.54	82.22	13.73	15.32	17.26	14.96
Split 3	69.02	78.02	73.18	83.16	13.80	15.82	15.72	15.43
Split 4	68.23	76.56	73.50	84.78	14.06	19.04	15.63	15.17
Split 5	67.85	77.44	71.11	83.56	14.71	17.29	16.13	15.43

Evaluation Metrics

The metrics used to validate the proposed models are listed below and provide a comprehensive assessment of model performance, crucial for reliable defect detection in industrial environments:

- **Box Precision:** The precision of the bounding boxes, which measures the proportion of predicted bounding boxes that are correct relative to true bounding boxes.
- **Box Recall:** The bounding box recall, which measures the proportion of true bounding boxes that were correctly predicted.
- **mAP50 (Box):** The average mean precision (mAP) calculated with an intersection over union (IoU) of 0.5, metric measures the overlap between the predicted bounding box and the ground truth bounding box, which is a common metric for evaluating the accuracy of bounding boxes.
- **mAP50-95 (Box):** The mAP calculated over an IoU range of 0.5 to 0.95, which provides a more comprehensive measure of the accuracy of the bounding boxes.
- **Mask Precision:** The accuracy of the masks, which measures the proportion of predicted masks that are correct relative to true masks.
- **Mask Recall:** Mask recall, which measures the proportion of true masks that were correctly predicted.
- **mAP Mask:** The calculated mAP for masks, which evaluates the accuracy of masks based on a given metric such as IoU.
- **mAP50 Mask:** The calculated mAP for masks with an IoU of 0.5, a metric specific to masks.

To evaluate the model, Recall was chosen as the primary metric, as shown in Equation 2. This choice reflects the model's emphasis on minimizing missed defect identifications rather than false positives. Recall effectively penalizes the model for failing to identify actual defects, ensuring that missed detections are prioritized in the evaluation.

Table 6 - Percentage of test instances per class.

Classe	Total de Instâncias de Teste	Porcentagem (%)
Crack	496	21.87
Scratch	583	25.71
Dent	554	24.43
Dirty	634	27.96

To enrich the evaluation, the standard confusion matrix developed by Ultralytics for YOLO models was also utilized[64]. This matrix is particularly advantageous as it is tailored specifically to assess YOLO-based models, ensuring an accurate evaluation of their performance. The matrix measures bounding box detection, classifying a bounding box as a True Positive (TP) if its confidence score exceeds 0.25 and the Intersection over Union (IoU) with the ground truth box is above 0.5. A False Positive (FP) is a predicted box with a confidence score over 0.25 but an IoU below 0.5, indicating a defect that was incorrectly identified. A False Negative (FN) occurs when a ground truth box lacks any predicted boxes with an IoU above 0.5, signifying a missed defect. In the matrix, the last column represents FN, emphasizing the importance of not overlooking any actual defects, while the last row represents FP.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

After the models were created, they were used to detect and classify the nonconformities in the images contained in the test folder. The original labels were then compared with the masks generated by the models to evaluate the practical results.

Experimental Results

The metrics presented in Figure 14 provide a detailed view of the models' performance in terms of detection and segmentation. The recall of the bounding box and the mask were defined as the most significant metrics.

It is also shown in the Figure 14, the YOLOv9 C model stood out as the best overall performance among the evaluated models. This model achieved the following results:

- **Box Precision:** 88.8%
- **Box Recall:** 80.7%
- **mAP50:** 85.6%
- **mAP50 95:** 69.9%
- **Mask Precision:** 87.6%
- **Mask mAP:** 83.8%

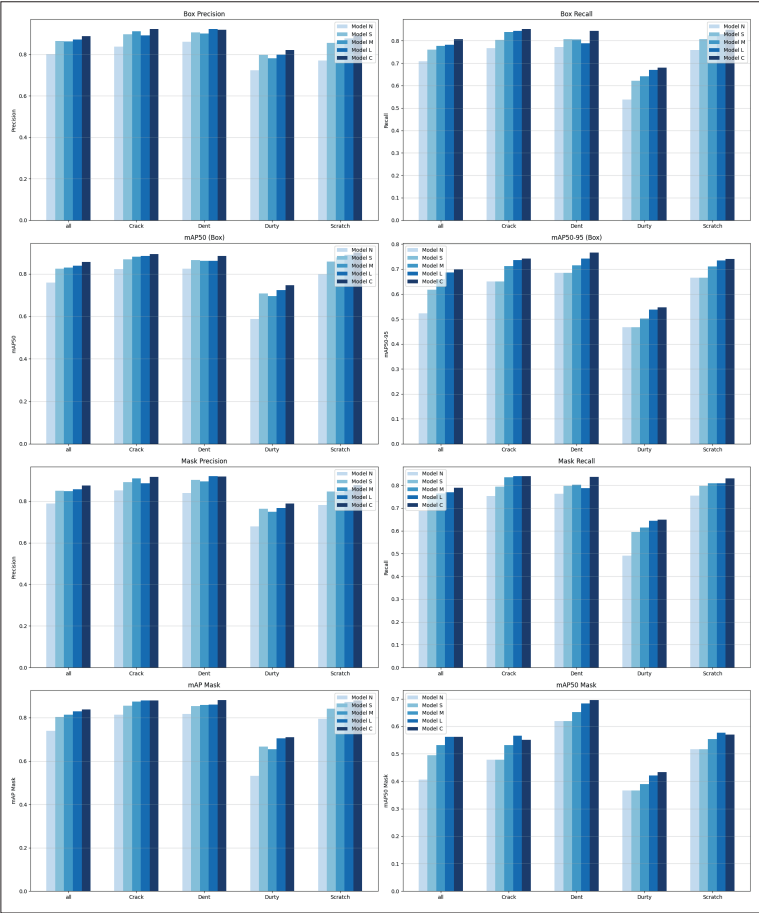
These metrics indicate that the YOLOv9 C model not only correctly detects object instances but also has a high precision rate in segmenting these instances.

Class Performance

Crack

- **Models:** All models showed good performance for this class, with box precision ranging from 83.7% (Model N) to 92.3% (YOLOv9 C).
- **mAP50:** Ranged from 82.3% (Model N) to 89.3% (YOLOv9 C).
- **Mask Precision:** Ranged from 85.2% (Model N) to 91.7% (YOLOv9 C).
- **Interpretation:** The 'Crack' class is detected and segmented with high precision and recall, indicating that the models can correctly identify cracks in most cases.

Figure 14 - Comparison between models N, S, M, L of YOLOv8 and C of YOLOv9.



Dent

- **Models:** 'Dent' exhibited very high metrics across all models, with box precision ranging from 86.1% (Model N) to 91.9% (YOLOv9 C).
- **mAP50:** Ranged from 82.4% (Model N) to 88.4% (YOLOv9 C).
- **Mask Precision:** Ranged from 83.9% (Model N) to 91.8% (YOLOv9 C).
- **Interpretation:** While 'Dent' shows high precision in detection and segmentation, the variability in recall suggests that some models struggle to consistently detect all instances of dents in some models

Dirty

- **Models:** 'Dirty' consistently showed lower recall values compared to other classes, with box recall ranging from 53.8% (Model N) to 68.1% (YOLOv9 C).
- **mAP50:** Ranged from 58.7% (Model N) to 74.7% (YOLOv9 C).
- **Mask Precision:** Ranged from 67.9% (Model N) to 78.9% (YOLOv9 C).
- **Interpretation:** The 'Dirty' class is more challenging for the models, showing lower recall, indicating that the models struggle to detect all instances of dirt.

Scratch

- **Models:** 'Scratch' showed high precision and recall values in the more advanced models, with box precision ranging from 77.0% (Model N) to 89.1% (YOLOv9 C).
- **mAP50:** Ranged from 79.8% (Model N) to 90.0% (YOLOv9 C).
- **Mask Precision:** Ranged from 78.2% (Model N) to 87.8% (YOLOv9 C).
- **Interpretation:** 'Scratch' is well detected and segmented, especially by more advanced models, indicating that the models have a good ability to identify and segment scratches.

It is also possible to observe lower results from the segmentation mask creation metrics compared to the metrics that measured detection with bounding boxes, which indicates greater difficulty for the models in performing segmentation of masks than in detecting and demarcating objects with bounding boxes.

Figure 15 presents the results of the normalized confusion matrix, focusing on detection outcomes from the best YOLOv8 model, specifically YOLOv8l-seg. A normalized confusion matrix displays the proportion of each classification relative to the total number of actual instances in each class, allowing for a clearer comparison of model performance across classes. It highlights the model's strengths and weaknesses in a more interpretable way, making it evident that there is still room for reducing False Negatives in the detection process.

In Figure 16 the results of using the YOLOv8 model L on a test batch are shown, demonstrating the model's ability to accurately identify, segment, and classify defects. A test batch refers to a selected subset of data that is used to

evaluate the model's performance after training. This batch includes a variety of examples that the model has not seen before, allowing for an assessment of how well the model generalizes to new, unseen data.

Evaluating the models in action, some difficulty of the models in identifying small objects was identified, in Figure 17, and shown an example of a situation in which the YOLOv8l-seg model failed to identify a small non-conformity , arrows show NCs that should have been identified.

It was identified that in some situations the models confuse glare with one of the defect classes, in Figure 18 the problem is visible when a glare is classified as a scratch.

The models achieved positive performances in the detection and classification of morpho-logically complex defects. Figure 19 shows the application of the YOLOv8l-seg model in complex images of defects from the test set.

The Table 7 shows the result of the visual analysis carried out by applying the YOLOv8l-seg model to the test set, examining the number of images with occurrences of the main problems studied.

In the Figure 20 it is possible to see the superior performance of YOLOv9 in relation to YOLOv8 for the scenarios of detection of small objects and false identification of glare as a defect.

Figure 15 - YOLOv8l-seg confusion matrix normalized.

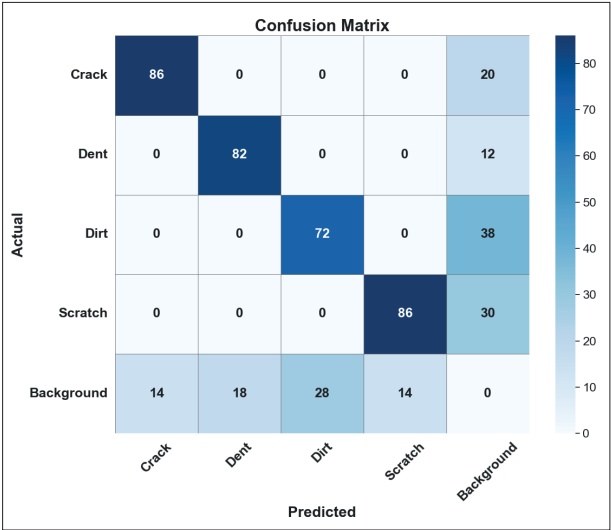
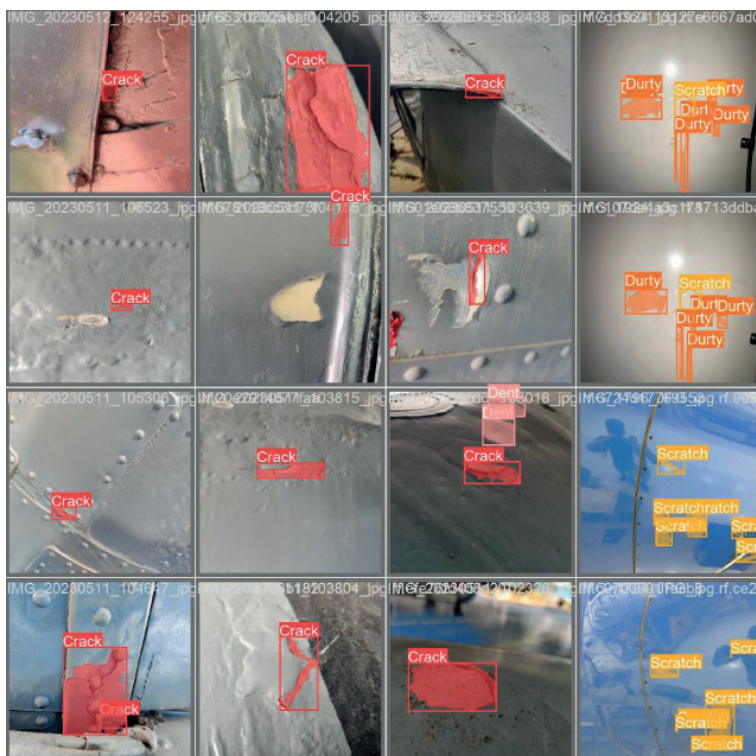


Figure 16 - YOLOv8l-seg test batch.



In Table 8 it is possible to observe the results of the same tests carried out in Table 7, but now applying YOLOv9c-seg. The results show a significant improvement in YOLO C.

Table 8 - . Visual Analysis Results.

Problem	Number of Occurrences
Incorrect classification between classes	4
Failure to identify small defects	16
Identification of glare as defect	2

Figure 17 - Difficulty identifying small objects.

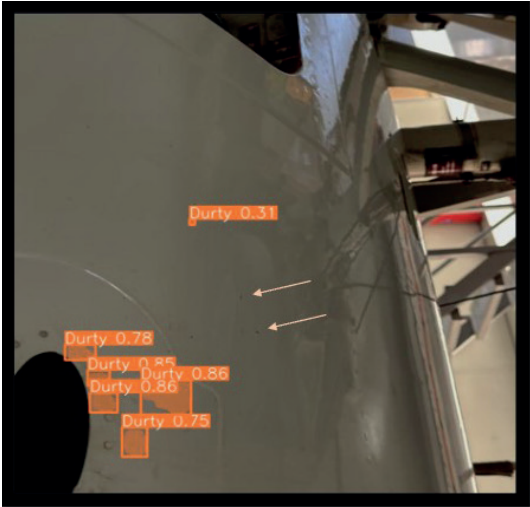


Figure 18 - Identification of glare as NC.

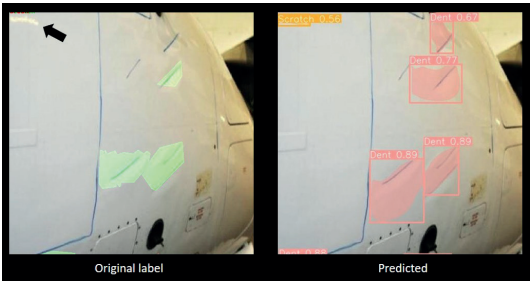


Figure 19 - Application of the L model on test set images.

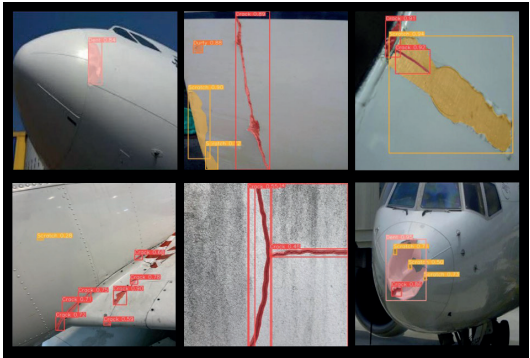


Figure 21 shows the results of the normalized confusion matrix resulting from the model created with YOLOv9c-seg. A general improvement in false negatives is visible, which is a critical point for the task.

Table 7 - Visual Analysis Results.

Problem	Number of Occurrences
Incorrect classification between classes	3
Failure to identify small defects	23
Identification of glare as defect	5

Figure 20 - Comparison between YOLOv8 and YOLOv9 results.

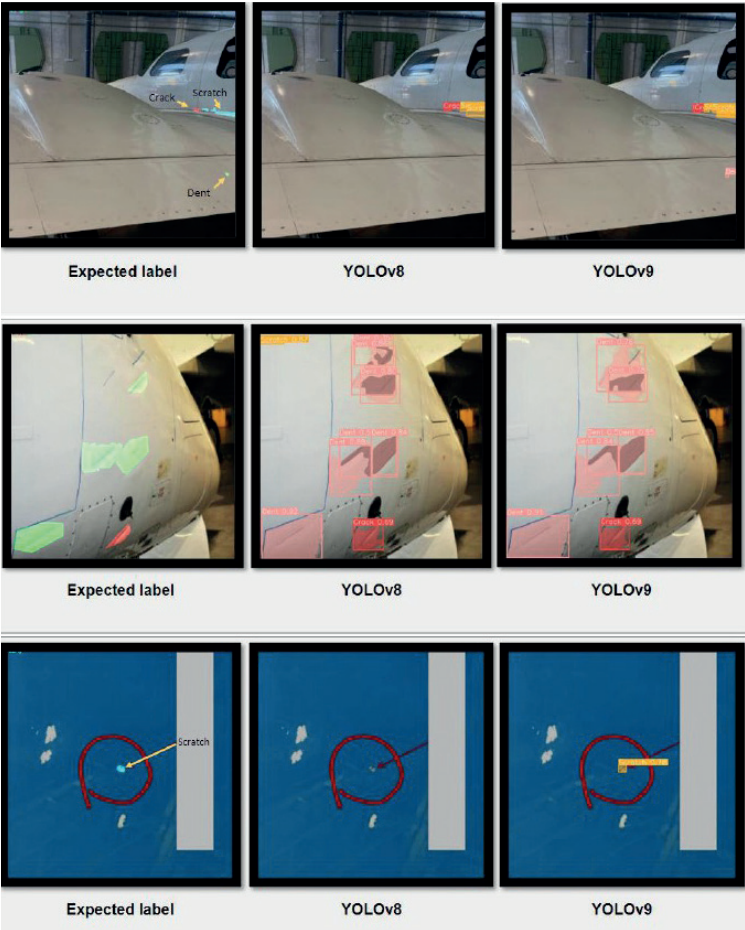
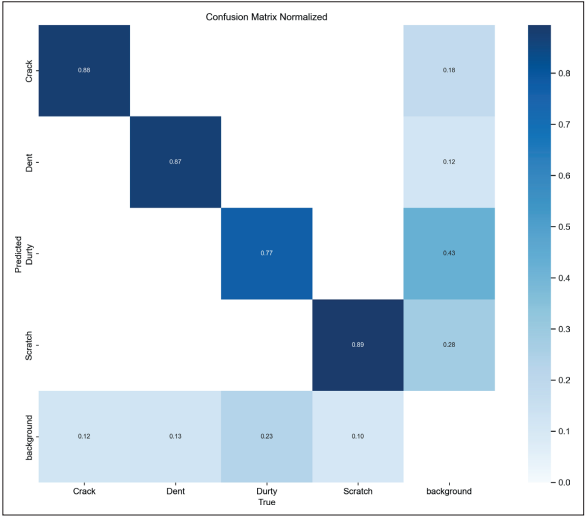


Figure 21 - YOLOv9c-seg confusion matrix normalized.



In the Figure 22 it is possible to observe the precision and recall curve at various confidence limits of the best model generated from YOLOv9c-seg, with the adopted confidence being 0.25% to maintain a high value of Recall.

Figure 22 - YOLOv9c-seg Precision Recall plot.

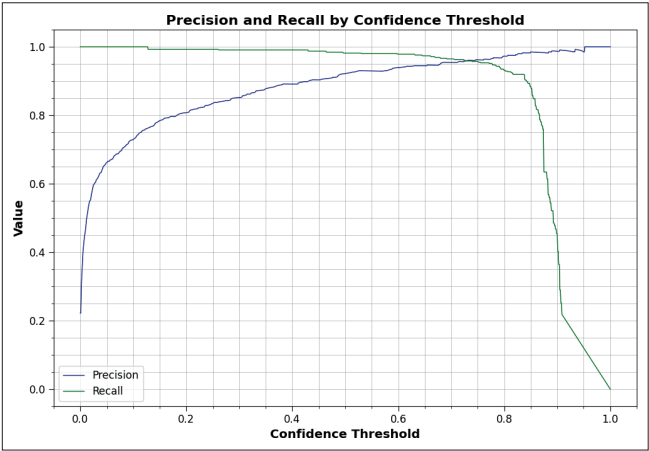
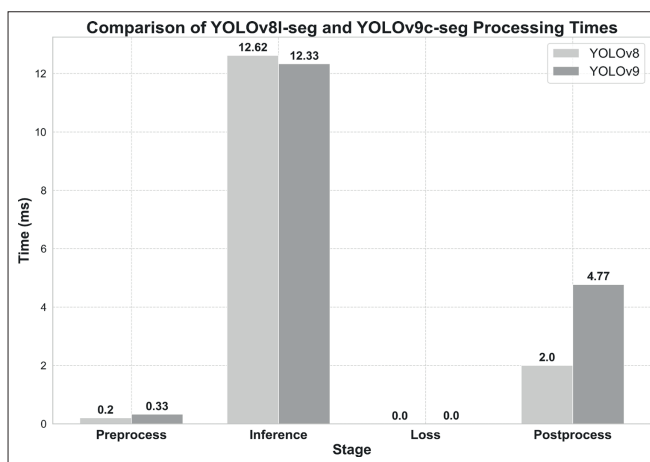


Figure 23 - Comparison of model speeds.



In the Figure 23 the speed of the two best models obtained through YOLO-v8l-seg and YOLOv9c-seg is shown.

CONCLUSION

Through the analysis of the metrics resulting from the tested models and visual assessments of the practical outcomes, it is possible to conclude that Deep Learning techniques integrated with computer vision can achieve good performance in detecting and classifying defects in aircraft fuselages, with high precision and recall. However, it is also clear that there are challenges that can be addressed to further improve performance. The analyses identified that the class with the greatest difficulty in generalization was the "Dirty" class, which consistently achieved lower metric levels compared to the other classes across all models tested. This result suggests the hypothesis that this class might have the highest morphological complexity compared to the others.

The objective of the initial tests was to determine the ideal neural network complexity for the task. It was found that, in most cases, the larger and complex models yielded the best results. This is evident from the comparison charts in Figure 14, where the evaluation metrics for each model are presented side by side. Proportionally, this fact leads to the understanding that the nature of the task is also highly complex.

It is visible that for the Dent class, if only Recall was analyzed, there was a decline in the result in the more complex models of YOLOv8, and this result may be related to the fact that this was the class with the smallest number of instances and images, it was even necessary to introduce 30% of images of dents on car surfaces to increase diversification, but evidence showed that this was still not enough for the YOLOv8 and, it seems, the more complex models of YOLOv8 generation may be suffering from overfitting. When analyzing the results in YOLOv9, it is possible to see that the same does not occur and the class Dents has a great result, as for the other classes, which indicates that YOLOv9 was able to retain more information and used the data set more effectively, which indicates that the architectural changes that PGI and GELAN introduced were very positive.

From the practical analysis of the application of the models with the results recorded in Table 7 and Table 8, it is possible to conclude that there was some difficulty in identifying small-scale defects, and in a small number of instances, the model incorrectly classified certain classes or wrongly identified areas of shine as defects. However, in all these cases errors were less frequent than expected, indicating generally good performance despite these challenges.

When comparing the results of the models created from YOLOv8 with the model developed from YOLOv9, a general improvement in performance is observed there was an improvement in reduction in errors misidentifying glare as defects and in the identification of small objects. When analyzing the confusion matrices, YOLOv9 visibly obtained a better result in reducing FN occurrences, which is the most critical item in the task.

Regarding the speed comparison of the models, the model generated with YOLOv8l-seg and YOLOv9c-seg were compared, such comparisons show small differences in results between the models in the pre-processing and inference time and a larger difference in the post-processing time. processing, with YOLOv9 obtaining longer times, which must be taken into consideration for real-time applications.

The analyses indicate that the models performed well with the generated dataset, particularly in the specific task of detecting and classifying nonconformities in aircraft fuselages. This suggests significant potential for using similar techniques to develop automated tools for quality inspections. However, continuous learning is essential, regularly expanding the dataset and allowing the models to become

familiar with as many real-world defect cases as possible. This ongoing adaptation will enable the models to improve continuously, ensuring greater accuracy and efficiency in inspections.

Discussion and Future Work

The 'Dirty' class consistently showed lower recall values, suggesting that future work should focus on improving the detection and segmentation of this specific class. This can be achieved through greater diversity in training data, hyperparameter optimization, or using advanced data augmentation techniques to improve representation of this class in training datasets. For the incorrect classification of glare as a non-conformity, the possibility of using polar lenses to avoid glare in the images could be investigated, while for the misclassification between classes, it could be mitigated with a thorough review of the labeling of the images in the dataset. For future studies. Other defects in aircraft fuselage that constantly impact production lines and aeronautical maintenance could also be included, such as paint failures, oxidation, missing screws, among others.

Since the instance segmentation technique precisely demarcates defects according to their shapes, it would also be valuable to explore methods for conducting dimensional qualification of non-conformities after detection and classification. This approach could enhance the overall assessment by providing detailed measurements and further insights into the severity and nature of each defect.

Furthermore, for future research, it would be beneficial to increase computational power by using more advanced GPUs or parallel processing. This would allow optimized searches for the best hyperparameters, further enhancing the performance of the models. Finally, real-time field testing with the model embedded in automation tools such as augmented vision glasses and drones is also recommended, which was not covered by this research.

REFERENCES

- [1] Dogru, Anil. Bouarfa, Soufiane. Arizar, Ridwan. Aydogan, Reyhan *Using Convolutional Neural Net-works to Automate Aircraft Maintenance Visual Inspection*. MDPI, Application of Multiagent Systems and Artificial Intelligence Techniques in Aviation (Volume II), 22, 2020.
- [2] Dong, Jingxuan. Wang, Jun. Li, Yida. Liu, Yuanpeng. Gong, Xiaoxi *A UAV-Based Aircraft Surface Defect Inspection System via External Constraints and Deep Learning*. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 71, 15, 2022.
- [3] Chouchene, Amal. Carvalho, Adriana. Lima, Tânia. Charrua-Santos, Fernando. Osório, Gerardo. Barhouni, Walid. *Artificial Intelligence for Product Quality Inspection toward Smart Industries: Quality Control of Vehicle nonconformities*. International Conference on Industrial Technology and Management, 9th, 5, 2020.
- [4] Beyerer, Jürgen. León, Fernando. Frese, Christian. *Machine Vision, Automated Visual Inspection: Theory, Practice and Applications*. Springer, 2016.
- [5] Javaid, Mohd. Haleem, Abid. Singh, Ravi. Rab, Shanay. Suman, Rajiv. *Exploring impact and features of machine vision for progressive industry 4.0 culture*. Sensors International, Volume 3, 11, 2022.
- [6] O'Mahony, Niall. Campbell, Sean. Carvalho, Anderson. Harapanahalli, Suman. Hernandez, Gustavo. Krpalkova, Lenka. Riordan, Daniel. Walsh, Joseph. *Deep Learning vs. Traditional Computer Vision*. Science and Information Conference, 144, 2019.
- [7] Mosca, Nicola. Patruno, Cosimo. Renò, Vito. Nitti, Massimiliano. Stella, Ettore. *Qualitative comparison of methodologies for detecting surface defects in aircraft interiors*. IEEE 8th International Workshop on Metrology for AeroSpace, 6, 2021.
- [8] H. Golnabi, A. Asadpour. *Design and application of industrial machine vision systems*. Published 1 December 2007. *Robotics and Computer-integrated Manufacturing*, 23(6): 626-633.
- [9] AL-KARKHI, Ameera. *A Study Comparing YOLOv8 and Detector2 for Object Segmentation Defect Identification*. Sheridan College/ Faculty of Applied Science and Technology. 7 Feb 2024.
- [10] DING, Meng; WU, Boer; XU, Juan; KASULE, Abdul Nasser; ZUO, Hongfu. *Visual inspection of aircraft skin: Automated pixel-level defect detection by instance segmentation*. College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China and Key Laboratory of Civil Aircraft Health Monitoring and Intelligent Maintenance of CAAC, Nanjing, China. Chinese Journal of Aeronautics, Chinese Society of Aeronautics and Astronautics & Beihang University. Received 19 November 2021; revised 20 January 2022; accepted 23 February 2022; available online 14 May 2022. Available at: www.sciencedirect.com.
- [11] U.S. Department of Transportation Federal Aviation Administration. *Advisory Circular: Inspection, Prevention, Control, and Repair of Corrosion on Avionics Equipment*. AC N°: 43-206, Initiated by: AFS-300, 7 Aug 2024.
- [12] Peter Martin. *Introduction to Surface Engineering and Functionally Engineered Materials*. Scrivener, 2011.
- [13] MARAZANI, Tawanda; MADYIRA, Daniel M.; AKINLABI, Esther T. *Repair of cracks in metals: A review*. Procedia Manufacturing, 14th Global Conference on Sustainable Manufacturing, GCSM 3-5 October 2016, Stellenbosch, South Africa.

- [14] Ashby, M. F., e Jones, D. R. H. (2005). *Engineering Materials 1: An Introduction to Properties, Applications, and Design*. Elsevier, 2012.
- [15] ATA. ATA Specification 100. 2002.
- [16] J. Wang, S. Wei, K. Wang, and J. Li, *Surface Defect Detection of Aircraft Glass Canopy Based on Improved YOLOv4*, 2022 21st International Symposium on Communications and Information Technologies (ISCIT), Xi'an, China, 2022, pp. 222-226, doi: 10.1109/ISCIT55906.2022.9931195.
- [17] QIU, Suo; XU, Xiangmin; CAI, Bolun. *FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks*. School of Electronic and Information Engineering, South China University of Technology, Guangzhou, P.R.China. arXiv:1706.08098v2 [cs.CV], 29 Jan 2018.
- [18] AGARAP, Abien Fred M. *Deep Learning using Rectified Linear Units (ReLU)*. arXiv:1803.08375v2 [cs.NE], 7 Feb 2019.
- [19] MISRA, Diganta. *Mish: A Self Regularized Non-Monotonic Activation Function*. Landscape, KIIT, Bhubaneswar, India. arXiv:1908.08681v3 [cs.LG], 13 Aug 2020.
- [20] LIU, Yijian; YANG, Junli; CUI, Wenqian. Simple, fast, accurate object detection based on anchor-free method for high resolution remote sensing images. **Beijing University of Posts and Telecommunications**, 2020.
- [21] Yuanpeng Liu, Jingxuan Dong, Yida Li, Xiaoxi Gong, and Jun Wang. *A UAV-Based Aircraft Surface Defect Inspection System via External Constraints and Deep Learning*. *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022, 5019315.
- [22] ZHUANG, Fuzhen; QI, Zhiyuan; DUAN, Keyu; XI, Dongbo; ZHU, Yongchun; ZHU, Hengshu; XIONG, Hui; HE, Qing. *A Comprehensive Survey on Transfer Learning*. arXiv:1911.02685 [cs.LG], DOI <https://doi.org/10.48550/arXiv.1911.02685> 10.48550/arXiv.1911.02685.
- [23] Mark Rice, Liyuan Li, Gu Ying, Marcus Wan, Eng Thiam Lim, Gao Feng, Jamie Ng, Melissa Nicole Teoh Jin-Li, and V. Shana Babu. *Automating the Visual Inspection of Aircraft*. Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore, Defence, Science and Technology Agency, Singapore.
- [24] Deyin Zhang, Mingwei Tan, Li Wang, Penghui Wei, Conghan Chen, and Wenxuan Hong. *Investigation of Aircraft Surface Defects Detection Based on YOLO Neural Network*, 2020 7th International Conference on Information Science and Control Engineering (ICISCE), 2020, doi: 10.1109/ICISCE50968.2020.00165.
- [25] JIANG, Peiyuan; ERGU, Daji; LIU, Fangyao; CAI, Ying; MA, Bo. *A Review of Yolo Algorithm Developments*. *Procedia Computer Science*, Volume 199, 2022, Pages 1066-1073.
- [26] REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv:1506.01497v3 [cs.CV], 6 Jan 2016.
- [27] HE, Kaiming; GKIOXARI, Georgia; DOLLAR, Piotr; GIRSHICK, Ross. *Mask R-CNN*. Facebook AI Research (FAIR). arXiv:1703.06870v3 [cs.CV], 24 Jan 2018.
- [28] LAN, Hong *et al.* PCB Defect Detection Algorithm of Improved YOLOv8. In: **8th International Conference on Image, Vision and Computing**. 2023.
- [29] SOHAN, Mupparaju; SAIRAM, Thotakura; RAMIREDDY, Ch. Venkata. *A Review on YOLOv8 and its Advancements*. School of Computer Science and Engineering, VIT-AP University, Amaravati, India.

- [30] TAO, Qian; CHEN, Yiyang; CHEN, Hongtian. A Detection Approach for Wafer Detect in Industrial Manufacturing based on YOLOv8. In: **CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)**. 2023. ISBN: 979-8-3503-3775-4.
- [31] ZHU, Liming; ZHANG, Jun; ZHANG, Qiang; HU, Hongtao. CDD-YOLOv8: A Small Defect Detection and Classification Algorithm for Cigarette Packages. In: **IEEE 13th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)**. 2023. ISBN: 979-8-3503-1519-6.
- [32] HUANG, Gao; LIU, Zhuang; VAN DER MAATEN, Laurens; WEINBERGER, Kilian Q. *Densely Connected Convolutional Networks*. Cornell University, Tsinghua University, Facebook AI Research. arXiv:1608.06993v5 [cs.CV], 28 Jan 2018.
- [33] ENERSTVEDT, Olga Mironenko. *Protection of Privacy and Data Protection in Aviation Security*. In book: *Aviation Security, Privacy, Data Protection and Other Human Rights: Technologies and Legal Principles* (pp.19-63), September 2017. DOI:10.1007/978-3-319-58139-22.
- [34] SABERIRONAGHI, Alireza; REN, Jing; EL-GINDY, Moustafa. *Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review*. Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON, Canada.
- [35] ROBOFLOW. *Search Results for "Fuselage Defects"*. Disponível em: <https://universe.roboflow.com/search?q=fuselage%2520defects>. Accessed on: 26 ago. 2024.
- [36] IEEE DATAPORT. *AircraftFuselageDet2023: Aircraft Fuselage Defect Detection Dataset*. Disponível em: <https://ieee-dataport.org/documents/aircraftfuselagedet2023-aircraft-fuselage-defect-detection-dataset>. Accessed on: 26 ago. 2024.
- [37] Sager, Christoph; Janiesch, Christian; Zschech, Patrick. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 2021. DOI: 10.1080/2573234X.2021.1908861.
- [38] Hafiz, Abdul Mueed; Bhat, Ghulam Mohiuddin. A Survey on Instance Segmentation: State of the art. *Journal Name*, 2023. DOI: DOI_do_Artigo.
- [39] ROBOFLOW. Disponível em: <https://roboflow.com/>. Accessed on: 26 ago. 2024.
- [40] Goodfellow, I., Bengio, Y., e Courville, A. (2016). *Deep Learning*. MIT Press.
- [41] Girshick, R. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7-13 December 2015*; pp. 1440-1448. [Google Scholar]
- [42] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22-29 October 2017*; pp. 2961-2969. [Google Scholar]
- [43] ATIK, Muhammed Enes; DURAN, Zaide. Comparison of YOLO Versions for Object Detection from Aerial Images. **Istanbul Technical University, Faculty of Civil, Geomatics Engineering Department, Istanbul, Turkey**. ISSN: 2148-9173, Vol. 9, Issue 2, June 2022.
- [44] REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. *You Only Look Once: Unified, Real-Time Object Detection*. University of Washington, Allen Institute for AI, Facebook AI Research. Disponível em: <http://pjreddie.com/yolo/>. arXiv:1506.02640v5 [cs.CV], 9 May 2016.

- [45] LIU, Wei; ANGUELOV, Dragomir; ERHAN, Dumitru; SZEGLY, Christian; REED, Scott; FU, Cheng-Yang; BERG, Alexander C. *SSD: Single Shot MultiBox Detector*. arXiv:1512.02325v5 [cs.CV], 29 Dec 2016.
- [46] REDMON, Joseph; FARHADI, Ali. *YOLO9000: Better, Faster, Stronger*. University of Washington, Allen Institute for AI. Disponível em: <<http://pjreddie.com/yolo9000/>>. arXiv:1612.08242v1 [cs.CV], 25 Dec 2016.
- [47] REDMON, Joseph; FARHADI, Ali. *YOLOv3: An Incremental Improvement*. University of Washington. arXiv:1804.02767v1 [cs.CV], 8 Apr 2018.
- [48] BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. Institute of Information Science, Academia Sinica, Taiwan. arXiv:2004.10934v1 [cs.CV], 23 Apr 2020.
- [49] DHRUTHI L et al. *State-of-the-Art Object Detection: An Overview of YOLO Variants and their Performance*. Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru - 560056, Karnataka, India. In: **4th International Conference on Smart Electronics and Communication (ICOSEC)**. 2023. ISBN: 979-8-3503-0088-8.
- [50] LI, Chuyi et al. *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. Meituan Inc. arXiv:2209.02976v1 [cs.CV], 7 Sep 2022.
- [51] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. *YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors*. arXiv:2207.02696v1 [cs.CV], 2022.
- [52] SOLAWETZ, Jacob; FRANCESCO. *What is YOLOv8? The Ultimate Guide*. 2023. 30 de abril de 2023. Disponível em: <<https://blog.roboflow.com/whats-new-in-yolov8/>>.
- [53] TERVEN, Juan; CÓRDOVA-ESPARZA, Diana-Margarita; ROMERO-GONZÁLEZ, Julio-Alejandro. *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. *Mach. Learn. Knowl. Extr.*, 2023, 5, 1680-1716. Disponível em: <<https://doi.org/10.3390/make5040083>>.
- [54] WANG, Chien-Yao; YEH, I-Hau; LIAO, Hong-Yuan Mark. *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*. Institute of Information Science, Academia Sinica, Taiwan; National Taipei University of Technology, Taiwan; Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan. arXiv:2402.13616v2 [cs.CV], 29 Feb 2024.
- [55] ULTRALYTICS. *YOLOv8-seg Models Documentation: YOLOv8n-seg, YOLOv8s-seg, YOLOv8mseg, YOLOv8l-seg*. Available at: <https://docs.ultralytics.com/yolov8/>. Accessed on: 26 Aug. 2024.
- [56] ULTRALYTICS. *YOLOv9c-seg Model Documentation*. Available at: <https://docs.ultralytics.com/yolov9/>. Accessed on: 26 Aug. 2024.
- [57] ZHANG, Jiawei. *Gradient Descent based Optimization Algorithms for Deep Learning Models Training*. IFM Lab Tutorial Series 1, Information Fusion and Mining Laboratory. First Version: February 2019; Revision: March 2019. arXiv:1903.03614v1 [cs.LG], 11 Mar 2019.
- [58] O'SHEA, Keiron; NASH, Ryan. *An Introduction to Convolutional Neural Networks*. Department of Computer Science, Aberystwyth University, and School of Computing and Communications, Lancaster University. arXiv:1511.08458v2 [cs.NE], 2 Dec 2015.

- [59] Bottou, L. (2010). *Large-Scale Machine Learning with Stochastic Gradient Descent*. In: Lechevallier, Y., Saporta, G. (eds) Proceedings of COMPSTAT'2010. Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-2604-3_16.
- [60] BERRAR, Daniel. *Cross-validation*. Data Science Laboratory, Tokyo Institute of Technology, Tokyo, Japan.
- [61] Carvalho, André Carlos Ponce de Leon Ferreira *et al.* *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*. Publisher: Capa Comum, 12 março 2021
- [62] KOHAVI, Ron. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. Computer Science Department, Stanford University, Stanford, CA 94305. Available at: <http://robotics.stanford.edu/~ronnyk/>.
- [63] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. *Deep Learning*. MIT Press, 2016. Available at: <http://www.deeplearningbook.org>.
- [64] Ultralytics. *YOLOv8 Performance Metrics - Confusion Matrix*. Available at: <https://docs.ultralytics.com/guides/yolo-performance-metrics/#introduction>. Accessed on: 26 Aug. 2024.

EXPLORING INTERPRETABILITY IN MACHINE LEARNING MODELS FOR HEALTHCARE DECISION SUPPORT SYSTEMS

Leandro Zanon Siqueira

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

Lilian Berton

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

ABSTRACT

This project aims to study Machine Learning applications in health-care decision support systems. The main objective is to use an explainable method (SHAP/LIME) to understand how black-box models make their decisions, promoting transparency between systems decisions in such a critical context. The research problem was set as a binary classification problem using the MIMIC-IV-ED dataset, we intend to predict patients with chest pain will be discharged based on features collected during triage and first medical contacts. The pipeline integrates a comprehensive preprocessing step, where numerical features are standardized, and categorical features are one-hot encoded. The models tested include logistic regression, random forest, support Vector Machines, XGBoost, and LightGBM. To address the class imbalance, the pipeline compares a random under-sample and over-sample method. Hyperparameter optimization is performed using random search to choose the best parameters for each model, and model parameters are fine-tuned to enhance predictive performance. The primary goal is to create an efficient and accurate binary classification model, providing valuable insights for clinical decision-making in emergency department settings.

INTRODUCTION

Context

In a healthcare system, important decisions are taken every time, critical ones as discharge in the realm of emergency medicine are determinant for patients lives. The integration of machine learning into clinical workflows must be treated as a support and it offers a powerful tool for healthcare professionals in making critical decisions. The MIMIC-IV-ED dataset (v2.2) [Johnson *et al.* 2020], which includes detailed triage assessments, ED stay records, and vital signs data, provides a rich characterization of a real world problem. By leveraging machine learning to interpret complex patterns within the data, we aim to provide explainable insights that empower healthcare professionals, helping them make more informed decisions without undermining the complexity and significance of their role in patient care. This approach underscores the importance of machine learning as a complementary asset, reinforcing the critical human element in healthcare rather than attempting to automate it.

The MIMIC-IV-ED dataset is a comprehensive source of de-identified emergency department (ED) data, derived from the larger MIMIC-IV database. It includes detailed records from the triage, ED stays, and vital signs tables. The triage table documents initial assessments and patient complaints at the time of triage, providing crucial insights into patient acuity upon arrival. The ED stays table was created to track patient admission and discharge from the ED for a single patient stay, including admission, discharge, and transfer details. The vital signs table records key physiological measurements throughout the ED stay, offering valuable data for real-time clinical decision-making and outcome prediction [Johnson *et al.* 2020].

Research Problem

Humans already coexist with Machine Learning (ML) solutions without being noticed. Researchers need to pay attention to the characteristics of machine learning models so that they can be used to make transparent decisions that are coherent with the domains to which they will be applied. At an Emergency Department, this has extra value, because we are dealing with people's

lives. So implementations of robust, interpretable, and clinically relevant models that can assist in the decision-making process for patient admission/transfer or discharge become more contextualized with medical knowledge. Risks to assuming any "metric champion" model as good decision support are lack of generalizability across different patient populations and limited integration with clinical workflows. Moreover, the interpretability of ML models is crucial in a clinical setting, where transparency and trust are essential for clinician adoption.

The research problem, therefore, is to develop and validate an ML-based decision support system that can accurately predict whether a patient presenting with chest pain—a common but complex symptom—should be admitted, transferred, or discharged from the ED. The model must address the challenges of data imbalance, ensure generalizability across diverse patient populations, and provide interpretable insights that can be seamlessly integrated into clinical decision-making processes.

This research will contribute to developing advanced decision support systems that can enhance the quality of care in emergency departments. By providing clinicians with reliable, data-driven insights, the proposed system aims to improve patient outcomes, reduce unnecessary admissions, and optimize the use of ED resources. The findings will have broader implications for integrating ML in healthcare, particularly in settings where rapid and accurate decision-making is critical.

GENERAL OBJECTIVES

Develop, implement, and interpret a robust machine learning pipeline that integrates data preprocessing, feature engineering, model selection, and resampling techniques to accurately predict whether a patient presenting with chest pain in the emergency department (ED) should be admitted, transferred, or sent home.

Specific Research Objectives

The specific research objectives are:

- Address data imbalance within the pipeline by incorporating techniques such as SMOTE (Synthetic Minority Over-sampling Technique - [Chawla et al. 2002]) and Random Under Sampling;

- Optimize the pipeline through hyperparameter tuning with cross-validation technique (Random Search CV);
- Validate the best pipeline in a test set to ensure generalizability and reliability.
- Integrate model interpretability techniques (e.g., SHAP values, LIME, or feature importance analysis) into the machine learning pipeline to provide clear, actionable insights into the factors influencing the model's predictions.

LITERATURE REVIEW

In healthcare decision support systems, interpretability is crucial for clinicians to trust and effectively use machine learning models. The SHAP framework ([Lundberg and Lee 2017]) provides a consistent method for explaining individual predictions by attributing the contribution of each feature, which is vital for transparency in clinical settings, such as predicting patient outcomes in emergency departments. Similarly, the LIME framework ([Ribeiro *et al.* 2016]) offers localized, interpretative explanations for individual predictions, helping clinicians understand how specific features influence a model's recommendations. Both frameworks enhance transparency, enabling healthcare professionals to make more informed and confident decisions based on the model's insights.

Chest pain in the emergency department is challenging as clinicians must differentiate between serious conditions like acute coronary syndrome (ACS) and less severe issues. While clinical judgment is crucial, advancements in diagnostic tools—such as cardiac injury markers, risk scores, stress tests, and noninvasive imaging—have improved risk stratification. Basic tools like history, physical examination, and electrocardiography help identify low-risk patients, enabling efficient management and transition to outpatient care. Advanced imaging techniques, including myocardial perfusion imaging and computed tomographic coronary angiography, further aid in reducing unnecessary admissions. Integrating intelligent systems, such as machine learning, can enhance these diagnostic processes by providing more accurate risk assessments and personalized recommendations, ultimately supporting clinicians in making informed healthcare decisions and improving patient outcomes [Kontos *et al.* 2010].

Machine learning models applied to the MIMIC-IV-ED dataset have demonstrated improved accuracy in identifying high-risk patients and optimizing triage processes. These models can analyze complex interactions between clinical variables, providing insights that surpass traditional assessment tools [Lin *et al.* 2024]. Additionally, incorporating interpretability techniques like SHAP and LIME has enhanced the transparency of these models, facilitating their integration into clinical workflows and aiding clinicians in understanding model predictions [Lundberg and Lee 2017].

This background highlights the potential of advanced machine learning approaches, supported by the MIMIC-IV-ED dataset, to transform chest pain management in the ED. By developing robust predictive pipelines that integrate data preprocessing, feature engineering, and interpretability, this study aims to improve decision-making and patient outcomes in emergency care.

METHODOLOGY

In this section, we detail the procedures of feature engineering, data preprocessing, and balance data for algorithm training and modeling to achieve the objective of our project. In particular, we focus on feature engineering with clinical meaning, random under-sampler to avoid noise from over-sampler methods, and define some popular binary classification models.

Feature Engineering Step

MIMIC-IV-ED is a large, freely available database of emergency department (ED) admissions at the Beth Israel Deaconess Medical Center between 2011 and 2019. The database contains 425,000 ED stays [Johnson *et al.* 2020], but for this project, we focused on ED stays that had 'chest pain' at the patient's reported reason for presenting to the ED, we remain with 14,231 data points. The following features were constructed using tables from MIMIC-IV-ED v2.2:

Triage table

The triage table records vital signs and initial assessments at the point of patient triage in the ED, offering essential data for evaluating the patient's immediate condition. Features considered:

- 'temperature': The patient's body temperature was recorded at the time of triage.
- 'respirate': The patient's respiratory rate, indicating the number of breaths per minute, recorded at triage.
- 'o2sat': The patient's oxygen saturation level, representing the percentage of oxygenated hemoglobin in the blood, measured during triage.
- 'sbp': The patient's systolic blood pressure, the pressure in arteries when the heart beats, recorded at triage.
- 'dbp': The patient's diastolic blood pressure, the pressure in arteries between heartbeats, measured at triage.
- 'heartrate': The patient's heart rate, indicating the number of heartbeats per minute, recorded during triage.

Vital Sign features

The vitalsign table contains aperiodic records of vital signs documented during a patient's stay in the emergency department, capturing key metrics such as temperature, heart rate, respiratory rate, oxygen saturation, blood pressure, and heart rhythm. Features considered:

- 'cv temperature': The coefficient of variation of the patient's body temperature, indicating the stability of this metric during the stay.
- 'cv heartrate': The coefficient of variation of the patient's heart rate, reflecting the stability of heartbeats per minute during the stay.
- 'cv respirate': The coefficient of variation of the patient's respiratory rate, representing the stability of breaths per minute during the stay.
- 'cv o2sat': The coefficient of variation of the patient's oxygen saturation, indicating the stability of blood oxygen levels during the stay.

- 'cv sbp': The coefficient of variation of the patient's systolic blood pressure, reflecting the stability of arterial pressure during heartbeats.
- 'cv dbp': The coefficient of variation of the patient's diastolic blood pressure, representing the stability of arterial pressure between heartbeats.

Pyxis MedStation features

The Pyxis MedStation is an automated system used in the ED for dispensing medications, and tracking each dispensation with detailed timestamps and medication information. Considered features:

- 'med n times': The total number of times medications were dispensed to the patient by the Pyxis MedStation during their stay.

ED stays features

The ED stays table tracks detailed information about each patient's visit to the emergency department, including admission and discharge times, demographics, and outcomes. Considered features:

- 'gender': gender of the patient.
- 'arrival transport': transportation method by which the patient arrived at the ED (AMBULANCE, WALK IN, or UNKNOWN).
- 'acuity': The severity level assigned to the patient during triage, with a lower number indicating higher severity.

Target

- 'target': used disposition column which means the patient's discharge location from the ED, indicating their outcome (HOME, ADMITTED, TRANSFER); target construction: 1 if HOME; 0 otherwise.

Preprocessing Step

In this preprocessing phase, the dataset is first examined to identify categorical and numerical features. Categorical features, which consist of non-numeric

As shown in the figure below, any feature is highly linear correlated (positive and negative). This approach leverages features with complementary information by treating numerical and categorical data separately and applying appropriate preprocessing techniques to each.

Correlation Matrix Heatmap

	med_n_times	cv_temperature	cv_heart_rate	cv_resprate	cv_azblat	cv_dlat	cv_dflap	temperature	heart_rate	resprate	o2blat	slap	dflap	acuity	gender_M	arrival_transport_WALK_N
med_n_times	1	0.017	0.14	0.13	0.054	0.14	0.075	0.0074	0.12	0.062	-0.015	0.054	0.013	0.0013	-0.063	0.021
cv_temperature	-0.017	1	-0.0015	0.023	0.0024	0.13	0.011	-0.11	0.0067	0.047	0.00027	-0.015	-0.002	0.0057	-0.011	-0.017
cv_heart_rate	0.14	-0.0015	1	0.11	0.023	0.18	0.009	-0.0061	0.13	0.0044	0.017	0.00069	0.029	0.0034	0.018	0.095
cv_resprate	0.13	0.023	0.11	1	0.053	0.089	0.066	-0.0074	0.066	0.91	-0.0074	0.0042	-0.035	-0.075	0.0099	-0.038
cv_azblat	0.054	0.0024	0.023	0.053	1	0.039	0.023	0.00021	0.028	0.0079	-0.017	-0.012	-0.0074	-0.029	0.0021	-0.022
cv_dlat	0.14	0.013	0.18	0.089	0.039	1	0.38	0.0075	0.014	-7.7e-05	1.9e-05	0.15	0.018	-0.027	-0.047	0.0085
cv_dflap	0.075	0.011	0.099	0.066	0.025	0.38	1	0.0047	-0.006	0.0045	-0.0035	0.031	0.036	-0.017	-0.084	-0.018
temperature	-0.0074	-0.11	-0.0061	-0.0074	0.00021	0.0075	0.0047	1	0.017	0.0021	-0.004	-0.0074	0.0015	0.013	0.01	0.013
heart_rate	0.12	0.0067	0.13	0.066	0.028	0.014	-0.006	0.017	1	0.035	-0.0088	-0.036	-0.027	-0.054	-0.035	0.07
resprate	0.062	0.0047	0.0044	0.91	0.0079	-7.7e-05	0.0045	0.0021	0.035	1	-0.0067	0.0044	0.0027	-0.0017	-0.012	-0.011
o2blat	-0.015	0.00027	0.017	-0.0074	-0.017	1.9e-05	-0.0035	-0.004	-0.0088	-0.0067	1	0.016	0.003	0.014	0.016	0.029
slap	0.054	-0.00069	0.0042	-0.032	0.15	0.031	-0.0074	-0.036	0.0044	0.016	0.016	1	0.0042	0.024	-0.016	0.11
dflap	-0.013	-0.002	0.029	-0.0035	-0.0074	0.018	0.056	0.0015	0.027	0.0027	0.003	0.0042	1	0.017	0.017	-0.0032
acuity	-0.0013	0.0057	0.0034	-0.075	-0.029	-0.027	-0.017	0.013	-0.054	-0.0037	0.014	0.024	0.017	1	-0.063	0.085
gender_M	-0.063	-0.011	0.018	0.0098	0.0022	-0.047	0.084	0.01	-0.035	-0.012	-0.016	-0.016	-0.017	-0.063	1	0.0088
arrival_transport_WALK_N	-0.021	-0.017	0.095	-0.038	-0.022	0.0085	-0.018	0.013	0.07	-0.011	0.029	0.11	-0.0032	0.085	0.0088	1

Class Label	Count	Percentage
HOME	9280	65.2%
ADMITTED/TRANSFERRED	4951	34.8%

Imbalanced Target Data

Considering the target created as explained above. The distribution is imbalanced, which means Admitted/Transferred patients are more common than sent-home patients after the feature engineering step. The distribution is

At our pipeline we decided to use re balanced class phase we experimented Random Under Sampler and SMOTE ([Chawla *et al.* 2002]) technique, the primary goal is to address class imbalance in the target data by reducing the number of instances from the majority class. This technique involves randomly selecting a subset of samples from the over-represented class to match the number of instances in the minority class, thereby creating a more balanced dataset.

Balancing the data is crucial for improving model performance, as it prevents bias towards the majority class and enhances learning from both classes. This leads to more accurate predictions and better generalization. Additionally, a balanced dataset provides more meaningful evaluation metrics, such as precision, recall, and F1 score, which offer a clearer assessment of model performance. Ensuring fair representation of all classes helps mitigate model bias and supports reliable predictions, especially in scenarios where the minority class is of particular importance.

Modeling Step

In the modeling step, different machine learning models are selected to evaluate their performance. These models include logistic regression, random forest, support vector classification, and gradient boosting methods such as XGBoost and LightGBM. Each model offers unique strengths and is suited for different types of data and problem settings.

To optimize the performance of these models, parameter tuning is performed using randomized search. This involves defining a range of values for various hyperparameters associated with each model. For example, logistic regression parameters include regularization strength and type, while random forest parameters include the number of trees and tree depth. Support vector classification parameters involve the regularization strength and kernel type. Gradient boosting models like XGBoost and LightGBM are tuned based on the number of boosting

rounds, tree depth, learning rate, and additional parameters specific to the boosting algorithm, such as the number of leaves in LightGBM.

Evaluate Step

The randomized search process explores different combinations of these parameters to identify the optimal settings, thereby improving the model's performance and generalizability. We optimized considering the metric Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve to be robust to imbalanced classes.

A confusion matrix applied to a test set supports researchers in evaluating Machine Learning outcomes. This quantifies the evaluation of model performance by showing counts of true positives, true negatives, false positives, and false negatives. So the risk of assuming classifications of the model is based on those metrics extracted from the confusion matrix. Key metrics derived from it include accuracy, precision, recall, F1 score, sensitivity, and specificity. For predicting whether a patient should be sent home or admitted/transferred we check the sensitivity (recall) measure because it quantifies the model's ability to identify patients requiring admission, while specificity assesses its ability to correctly identify those who should be sent home. Both of those metrics were considered in our model selection step.

The Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve measures a model's ability to distinguish between classes. It provides a single value representing the overall performance across all classification thresholds. An AUC close to 1 indicates excellent model performance, as the model effectively distinguishes between positive and negative cases, while an AUC near 0.5 suggests poor performance, akin to random guessing. For predicting patient outcomes, a high AUC signifies that the model reliably identifies patients who need admission versus those who do not, offering a robust measure of its discriminate power.

RESULTS

From the methods explained above, our study's outcome consists of a promising pipeline. The results will be discussed detailing the confusion matrix and AUC metric.

The best performances for each model are presented in Table 2. Paring LightGBM with Random Under Sampler were the best-paired algorithms, so its confusion matrix is presented in Figure 2:

Figura 2 - Confusion Matrix best model: LightGBM.

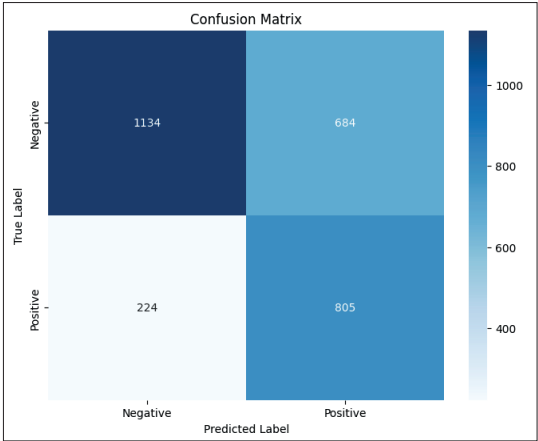


Tabela 2 - Performance by each model (after hyperparameters-tuning).

Model	Sampling Method	AUC	Sensitivity	Specificity
LogReg	Under Sampling	0.68	63%	63%
LogReg	SMOTE	0.68	63%	63%
Random Forest	Under Sampling	0.75	72%	64%
Random Forest	SMOTE	0.74	54%	78%
Support Vector Classifier	Under Sampling	0.75	72%	64%
Support Vector Classifier	SMOTE	0.75	68%	66%
XGBoost	Under Sampling	0.77	72%	66%
XGBoost	SMOTE	0.76	54%	80%
LightGBM	Under Sampling	0.77	72%	68%
LightGBM	SMOTE	0.76	54%	79%

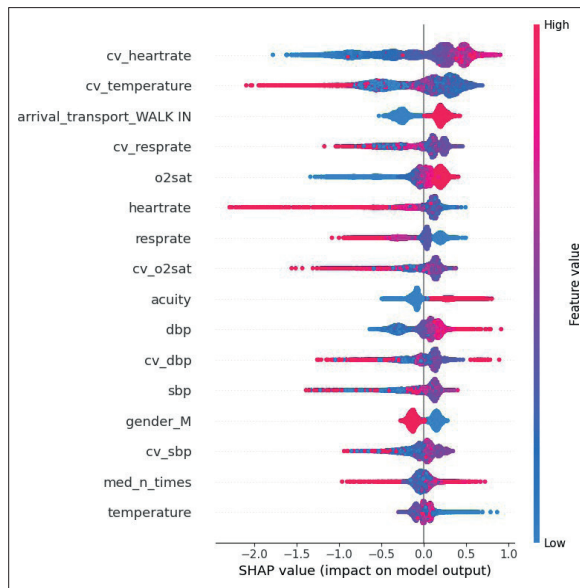
Explainability

To delve deeper into the concept of explainability we interpret SHAP values from the best model selected above (LightGBM). For local interpretability, SHAP values elucidate the impact of individual features on a single prediction, thereby offering insights into why a specific decision was made by the model. This is particularly valuable in clinical settings where understanding the rationale

behind a prediction is critical. For example, if a patient's elevated blood pressure contributes positively to the prediction of a particular diagnosis, this is clearly indicated by a positive SHAP value.

On a global level, SHAP values aggregate feature contributions across numerous predictions to reveal overall feature importance and interactions. This aggregated view helps in identifying which features are most influential in the model's decision-making process and ensures that the model's behavior is aligned with domain knowledge and expectations. The image below focuses on global impact ordered from the most to the least important to the model.

Figura 3 - Mean absolute SHAP values.



In a SHAP summary plot, the red and blue dots represent the feature values and their corresponding SHAP values for a given feature. Here's what each color typically signifies:

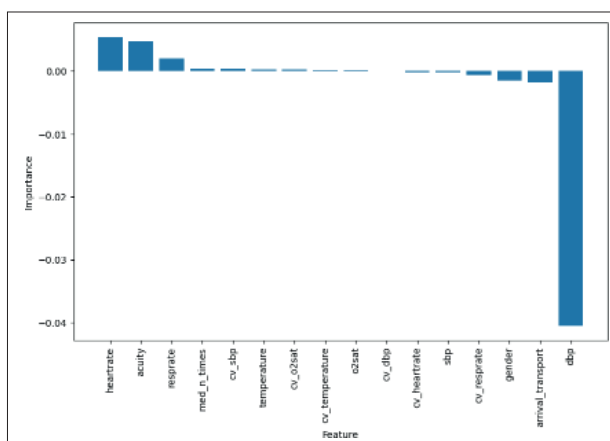
- **Red Dots:** These represent higher feature values. For instance, if a feature is "temperature", a red dot would indicate a higher temperature value. In the context of the SHAP summary plot, red dots show how higher values of the feature affect the model's prediction.

- **Blue Dots:** These represent lower feature values. Continuing with the “temperature” example, blue dots would indicate lower temperature values. Blue dots show how lower values of the feature influence the model’s prediction.

Intuitively, the stability of vital signs, particularly heart rate, represented by the coefficient of variation (CV), is on top of the SHAP values graph above. Often reflect that stable vital signs, indicated by a low CV, contribute positively to the model’s prediction that a patient can be sent home from the Emergency Department (ED). This suggests that patients with stable vital signs are generally at a lower risk of acute medical issues and, therefore, are more likely to be discharged rather than admitted or transferred. The coefficient of variation helps quantify this stability, guiding the model’s decision-making process.

At LIME [Ribeiro *et al.* 2016] interpretation, dbp, and heart rate were top feature importance as we can see in Figure 4. Despite the heart rate feature makes sense to be important, this methodology shows us different insights that need to be discussed with experienced physicians to determine which interpretation tool is more reliable based on the domain of the problem. It is worth remembering that LIME is designed for local interpretations, in other words, this method aims to identify for each data point which feature when modified also modifies the model outcome. So the approach is local, but for a global interpretation we assumed the aggregation to compute the “global impact for each feature”.

Figure 4 - Global Feature Importance using LIME.



CONCLUSIONS

In this study, we developed and evaluated robust machine-learning pipelines for predicting patient discharge from the Emergency Department. With an AUC of 0.77, the results are promising given the features outlined in the Methods section. This pipeline shows potential for supporting, though not replacing, decision-making processes at the Beth Israel Deaconess Medical Center in Boston, MA. However, it is essential to assess the pipeline's performance carefully before applying it to other healthcare settings to ensure consistent results.

Finally, at the explainability step, we compared the interpretation of two popular methods (LIME and SHAP). Surely the results enrich any discussion between the machine learning researchers with the domain experts, connecting the solution with the problem and supporting how to define the next steps based on how much the physician can trust our best pipeline, driving the research for an iterative model improvement. Considering global interpretation SHAP demonstrates more consistency associating clearly heart rate stability with discharge in chest pain, so we conclude this had a better agreement with clinical context.

For future studies, we recommend collaborative work with physicians and emergency care experts to better understand the characteristics that influence medical decision-making for patient discharge predictions. Exploring additional features, such as physical examination results, could potentially improve model performance. Investigating the relevance of these new features through explainability methods such as SHAP may provide deeper insights into their impact on the model's predictions and overall interpretability.

REFERÊNCIAS

- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.
- Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. (2020). Mimic-iv. *PhysioNet*. Available online at: <https://physionet.org/content/mimiciv/1.0/> (accessed August 23, 2021), pages 49–55.
- Kontos, M. C., Diercks, D. B., and Kirk, J. D. (2010). Emergency department and office-based evaluation of patients with chest pain. In *Mayo Clinic Proceedings*, volume 85, pages 284–299. Elsevier.

Lin, S., Lu, W., Wang, T., Wang, Y., Leng, X., Chi, L., Jin, P., and Bian, J. (2024). Predictive model of acute kidney injury in critically ill patients with acute pancreatitis: a machine learning approach using the mimic-iv database. *Renal Failure*, 46(1):2303395.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?"explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

PROCESSING OF METEOROLOGICAL IMAGES TO PREDICT FLIGHT HOLDING MANEUVERS

Stéfano Campos de Oliveira

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)
stefanocamposdeoliveira@gmail.com

Filipe Alves Neto Verri

Division of Computer Science – Aeronautics Institute of Technology (ITA)

ABSTRACT

This work proposes the improvement of an existing model that predicts flight holding maneuvers based on meteorological data and satellite images. A holding maneuver is an airborne delay procedure that aircraft follow when they cannot land immediately, typically due to traffic congestion or adverse weather conditions. Predicting these maneuvers is important for optimizing air traffic management, minimizing delays, and ensuring safety. The original model crops a large static region encompassing all of Brazil, while this research introduces a localized approach that crops smaller, more relevant areas around the destination airport. Specifically, this study tests cropping images with areas of 500 km², 1000 km², and 2000 km² to enhance prediction accuracy, ultimately aiding in better flight management and reducing delays caused by adverse weather conditions.

INTRODUCTION

Air traffic management is essential for maintaining the safety and efficiency of flight operations. Predicting flight holding maneuvers due to weather conditions is essential to minimize delays and ensure smooth operations. This research focuses on refining an existing model by utilizing satellite imagery with a localized approach that focuses on smaller regions around the destination airport. This approach is expected to improve the prediction of flight holding maneuvers, thereby enhancing air traffic management.

BACKGROUND AND RELATED WORKS

Predicting flight holding maneuvers has been a critical area of research for improving air traffic management. Weather conditions significantly influence the need for holding patterns, which can lead to delays and increased fuel consumption. Previous research has utilized various approaches, including machine learning and satellite imagery, to predict such maneuvers.

One notable contribution in this field is the model developed during the *Data Science Challenge at EEF 2024* [Verri 2024]. This challenge, organized by Filipe Verri, inspired the creation of the baseline model used in this research. The challenge focused on leveraging machine learning techniques to improve predictions related to flight operations, providing a foundation for further exploration in this study. The original code for this model is available in the repository [Ryukinix 2024], which provides a foundation for this research.

[Kalyani *et al.* 2020] proposed a machine learning model to predict flight delays based on meteorological data. The study demonstrated the importance of weather variables in predicting flight operations.

[Sharan *et al.* 2021] explored state-of-the-art machine learning techniques for predicting flight delays, emphasizing the need to incorporate detailed meteorological data for accurate predictions.

[Addu *et al.* 2022] developed a model that used satellite imagery to predict flight disruptions due to weather conditions. The study highlighted the importance of regional focus in improving prediction accuracy.

These studies underscore the value of meteorological data in predicting flight maneuvers and suggest the need for more localized data analysis to enhance prediction models.

Research Problem

The research problem addressed in this work revolves around the need for more precise predictions of flight holding maneuvers due to weather conditions. Current models, such as the one being compared, utilize a broad region encompassing all of Brazil for weather data, which may not fully capture localized weather patterns affecting specific airports. This research seeks to answer the question: **“How can predictive models that use localized cropped satellite images around the destination airport, improve the accuracy of flight holding maneuver predictions compared to static regional models, thereby optimizing air traffic management?”**

General Objective

Enhance the accuracy of flight holding maneuver predictions by developing and comparing a model that utilizes localized cropped satellite imagery focusing on areas of 500 km², 1000 km², and 2000 km² around the destination airport to an existing model that uses a static crop encompassing all of Brazil.

Specific Objectives

- Implement a localized cropping strategy that extracts areas of 500 km², 1000 km², and 2000 km² around the destination airport from satellite images: This will ensure the model uses localized meteorological data for more accurate predictions.
- Integrate the localized cropped image embeddings into the existing model's training data: This integration aims to enhance the model's predictive capabilities by providing more relevant meteorological information.

- Compare the performance of the localized cropping approach with the original static model: The impact on model accuracy will be assessed using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to determine if the localized approach offers improvements.

Contributions

This work introduces a localized cropping strategy for satellite images, focusing on areas of 500 km², 1000 km², and 2000 km² around destination airports. These localized cropped images were integrated into the model's training data, aiming to improve the use of localized meteorological information. The study compared this approach to the original static model, revealing insights into the strengths and limitations of localized cropping. Although the localized cropping did not significantly boost accuracy, the research identified key areas for future improvement, such as refining image embedding techniques and incorporating additional temporal and operational data.

DATASETS FOR METEOROLOGICAL IMAGES AND FLIGHT OPERATIONS

The development and comparison of models for predicting flight holding maneuvers rely on several datasets, each providing crucial information. The primary data sources include METAR and TAF reports, the ANAC dataset, and satellite imagery from INPE.

The METAR (Meteorological Aerodrome Report) and TAF (Terminal Aerodrome Forecast) datasets offer detailed weather information crucial for understanding the conditions under which flights operate. These datasets include variables such as wind conditions, visibility, cloud cover, temperature, and pressure, which are essential for capturing the environmental factors that influence flight operations.

Additionally, the ANAC dataset contributes valuable details about airports, including their OACI codes, altitude, operational conditions (both day and night), runway dimensions, and surface types. This information is vital for understanding the physical and operational context in which flights occur.

For satellite imagery, the study utilizes data from the INPE's GOES-16 satellite. These images provide a visual representation of weather conditions

around airports, with the imagery being processed and cropped to focus on specific areas of interest. By integrating these diverse datasets, the study constructs a comprehensive picture of the factors that might influence flight holding maneuvers, allowing for more accurate predictions.

EXPECTED RESULTS

This project will provide a detailed analysis comparing the relationship between localized weather conditions and flight holding maneuvers using localized versus static cropping strategies. The refined model, using a localized approach with different cropping areas, is expected to achieve higher accuracy in predicting flight maneuvers, thereby assisting air traffic controllers and airlines in managing flights more efficiently.

The results will be evaluated using various performance metrics, and the findings will offer insights into how localized meteorological data can enhance the predictability of flight operations. The comparison will determine whether the localized approach offers significant improvements over the original static model.

METHODOLOGY

This section outlines the methodology used to predict flight holding maneuvers by integrating meteorological satellite images with flight data and comparing a localized cropping approach to a static one. The approach includes data handling, image processing, feature extraction, model training, and evaluation.

Data Handling

The process begins with data handling, which involves several key steps to prepare the dataset for model training. These steps include dataset preparation, cleaning and feature engineering, data imputation, and data splitting.

The primary dataset, sourced from public f2.csv, includes flight schedules, meteorological conditions, and satellite image URLs. The full dataset, comprising 302,399 records, was used to ensure comprehensive representativeness. The dataset is loaded using the get train dataset function from datasets.py.

Columns with only null values are removed to reduce noise. We engineer time delta features to capture temporal patterns and drop non-predictive columns, such as those related to datetime and satellite image URLs, ensuring the model focuses on relevant features.

Categorical features with missing values are filled with an "UNKNOWN" category, while numerical features are imputed using CatBoost's default strategies. This process ensures the dataset is complete and ready for training.

The dataset is split into training, validation, and test sets to assess model performance on unseen data. For real-world testing, the test set is replaced with unlabeled samples to simulate practical conditions.

Feature Overview

The training dataset comprises 65 features across 21,123 entries, encapsulating a broad spectrum of data that influences flight operations. These features can be categorized into three main groups: flight-specific attributes, detailed meteorological data, and derived features from satellite imagery.

- **Flight-Specific Features:** These include the departure (origem) and destination (destino) airports, runway change indicators (prev troca cabeceira, troca cabeceira hora anterior), and detailed characteristics of the destination airport such as altitude, operational conditions (day/night operations), runway dimensions, and surface type. These attributes provide a foundational understanding of the physical and operational context of each flight.
- **Meteorological Data:** Detailed weather information is sourced from METAR (Meteorological Aerodrome Report) and METAF (Forecast Aerodrome Report) datasets. This includes wind conditions, visibility, cloud cover, temperature, dew point, and altimeter settings. These weather-related features are critical for capturing the environmental conditions that could impact flight operations, such as visibility on the runway, wind direction and speed, and overall atmospheric pressure.

- **Derived Features:** Among the derived features, one of the most significant is the vector feature, which stores image embeddings extracted from satellite images. These embeddings encapsulate visual weather information around the destination airport, adding a unique and rich layer of data to the model. Another derived feature, *hora classe*, categorizes flights based on the time of day, which can influence flight operations due to varying traffic levels and weather conditions.

This comprehensive dataset *allows* the model to analyze a wide range of factors affecting flight holding maneuvers. The inclusion of both numerical and categorical data, as well as the integration of image-derived vectors, ensures that the model is well-equipped to predict flight outcomes under various meteorological conditions.

Image Processing and Feature Extraction

After providing an overview of the features used in this study, the next step involves processing the satellite images and extracting relevant features from them. This process enhances the dataset by integrating visual meteorological data.

Satellite images corresponding to each flight are downloaded and processed in three ways: using the original static crop encompassing all of Brazil, and localized crops that capture areas of 500 km², 1000 km², and 2000 km² centered around the destination airport. The original static crop, which covers the entire Brazilian region, is illustrated in Figure 1. In contrast, the localized cropping approach, as shown in Figure 2, focuses on a smaller, more relevant area around the destination airport, such as the 500 km² crop. This localized cropping is performed using the `download_image_vectors.py` script and the `crop_airport_area` function in `cropper.py`.

Figure 1 - Original static crop (Brazil).

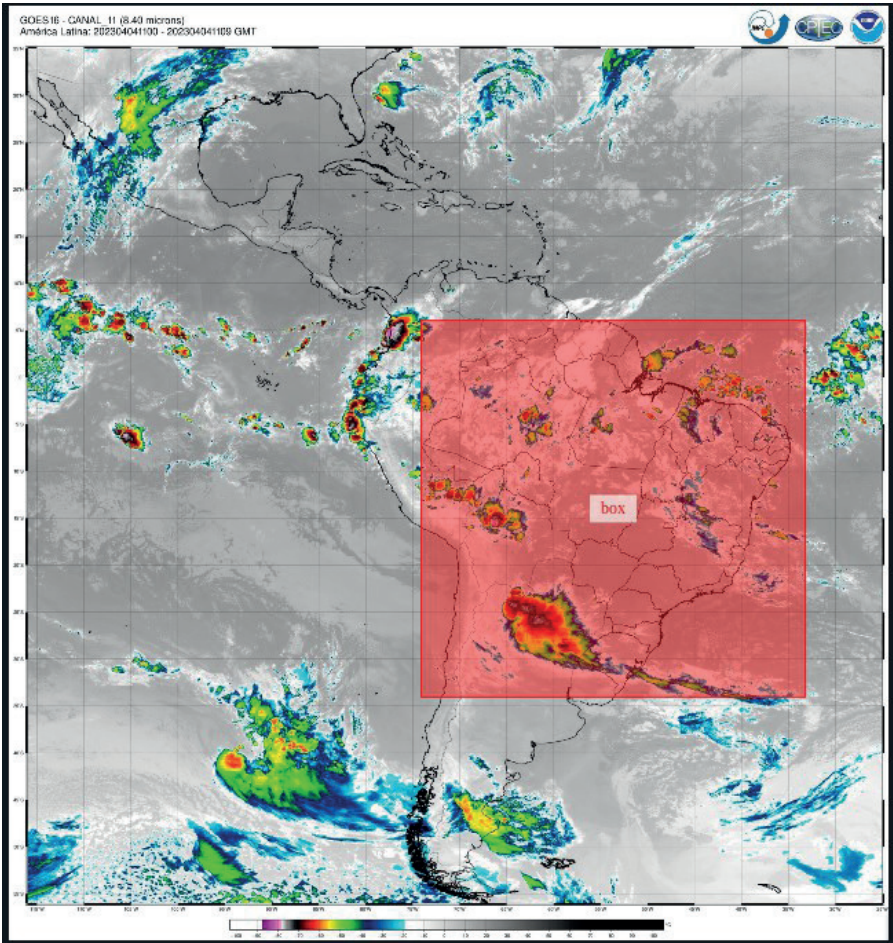
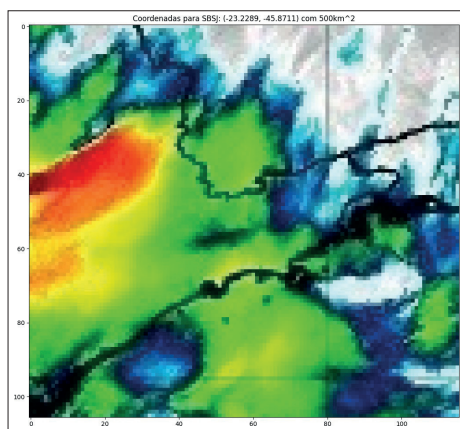


Figure 2 - Localized crop of 500km squared around SBSJ airport.



The cropped images are converted into feature vectors using a pre-trained Vision Transformer (ViT) model, facilitated by the feature extraction from image function in `features.py`. These vectors add valuable visual data to the model, providing a comparative basis between the localized and static approaches.

The image vectors are stored in the `images_cropped.pkl.xz` file, alongside their flight IDs, for integration into the main dataset during model training. Separate storage is maintained for vectors derived from the static and localized approaches.

Model Training

With the features extracted from the processed satellite images, the next step is to integrate these features into the dataset and proceed with the model training. This stage is essential for developing a predictive model that effectively leverages the enriched data to forecast flight holding maneuvers.

The stored image vectors from both the static and localized approaches are reloaded and integrated with the main dataset using the `add image vectors` function in `handling.py`, creating comprehensive datasets for comparison during model training.

Graph-based features, such as betweenness centrality and edge connectivity, are generated using the `generate graph features` function in `features.py`, adding a layer of network analysis to the model.

The CatBoostRegressor model is configured with specific parameters, including the number of iterations and early stopping criteria. The model is trained separately on datasets derived from the static and localized cropping approaches using the `catboost-regression.ipynb` notebook.

Model Evaluation

Following the training process, the model's performance must be rigorously evaluated to ensure its effectiveness. The evaluation methodology applied in this work mirrors the original model's evaluation process, using the same metrics and criteria to assess how well the model predicts flight holding maneuvers. This consistent approach allows for a direct comparison of the new model's performance against the original.

An important aspect of the evaluation was the analysis of feature importance. Figure 3 shows the feature importance scores for both the static and localized cropping models. Notably, the "vector" feature—which stores the satellite image embeddings—ranked as the third most important feature, consistent with the findings from the original model. This emphasizes the significance of satellite images in providing critical meteorological context for predicting flight holding maneuvers.

The predictions from both the static and localized approaches are evaluated against the test set using metrics like R^2 and RMSE. A comparison of these metrics will determine if the localized approach offers any improvement over the static model. A confusion matrix is generated for both approaches to assess classification accuracy.

Another important aspect of the model's predictions is the distribution of the predicted values. Figure 4 shows the distribution of predicted holding maneuvers when the "espera" feature exceeds 241 seconds. Interestingly, the distribution resembles a Chi-Square distribution, which may indicate that extreme holding maneuvers, though rare, dominate the tail end of the predictions.

A possible hypothesis is that such long holding times are outliers caused by rare but significant weather events or operational delays. Another hypothesis could be that the model struggles to predict very long holding times accurately due to insufficient training data in these extreme cases.

Prediction Storage

The model’s predictions from both approaches are saved alongside relevant flight details for use in operational decision-making or further analysis. This comparative analysis will help determine the most effective strategy for predicting flight holding maneuvers.

EXPERIMENTAL SETUP

This section details the practical application of the methodology, covering dataset preparation and the steps taken to rigorously evaluate the model’s performance.

Figure 3 - Feature importance plot showing the “vector” as the third most important feature in the model.

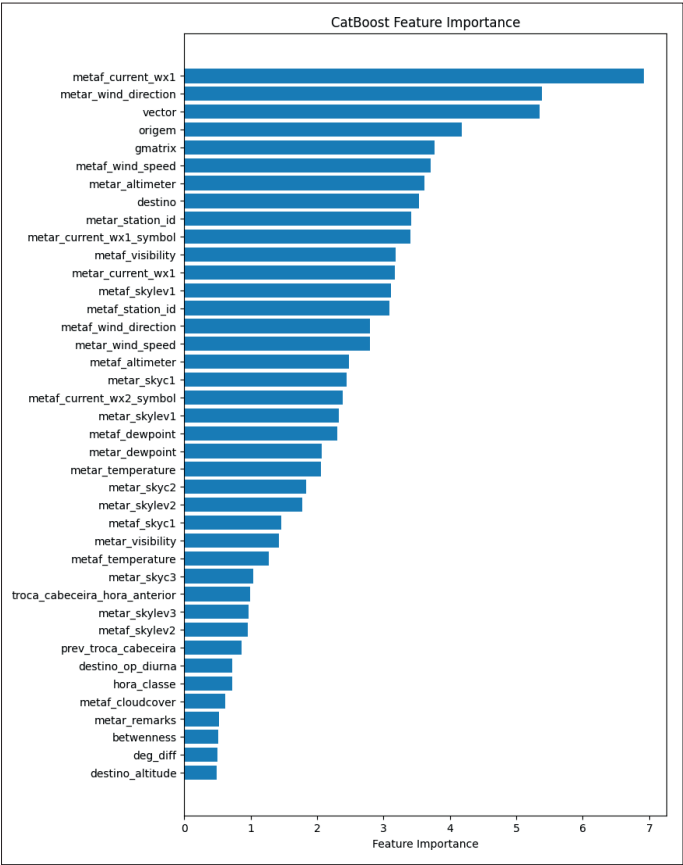
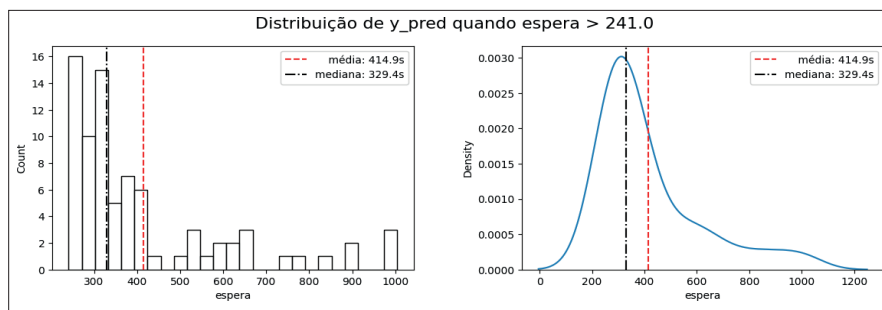


Figure 4 - Distribution of predicted holding maneuvers when “espera” exceeds 241 seconds, resembling a Chi-Square distribution.



Training and Validation Datasets

The datasets used for training, validation, and testing were derived from the `public_f2.csv` file, which contains comprehensive flight and meteorological data. To ensure a robust evaluation of the model, the data was split into training, validation, and test sets using an 80/20 ratio for train and test, respectively.

Initially, the dataset was split into a training set and a test set, with 80% of the data allocated for training and 20% for testing. This test set was kept separate to evaluate the final model's performance on unseen data.

Within the training set, further splitting was performed to create a validation set, which accounted for 12.5% of the original dataset (or 1/8 of the training data). The validation set is used to fine-tune the model's hyperparameters and monitor its performance during training, enabling adjustments to prevent overfitting.

For scenarios requiring real-world testing, the unlabeled samples replaced the standard test set. This substitution simulates practical conditions where the model must make predictions on new, unseen data without prior labels. In these cases, the evaluation of the model focuses on its predictive power and generalization capabilities in a live environment.

To maintain data integrity throughout this process, care was taken to ensure that all splits were random but consistent, using a fixed random seed for reproducibility. Additionally, the categorical and numerical features were appropriately imputed before splitting to avoid data leakage and to ensure that the training, validation, and test sets were all complete and ready for use in the model.

The localized and static approaches were applied separately to these datasets, ensuring that both models were trained, validated, and tested under the same conditions, providing a fair and unbiased comparison of their performance.

RESULTS AND ANALYSIS

This section presents the results obtained from evaluating the original model and various versions of the model using different cropping strategies. The models were assessed from two perspectives: regression and classification. The regression analysis, which predicts the duration of holding maneuvers in seconds, was evaluated using the R^2 score and RMSE. On the other hand, the classification task, which determines whether a holding maneuver will occur or not, was compared using the F1-score and precision. This dual approach allows for a comprehensive evaluation of the model's effectiveness in predicting both the likelihood and the duration of flight holding maneuvers.

Performance Comparison

Table 1 summarizes the performance metrics for the original model and the models using different cropping strategies:

The original model, which uses a static crop encompassing all of Brazil, achieved an R^2 score of 0.092, an RMSE of 131.44 seconds, and an F1-score of 0.115. In comparison, the localized cropping strategies with 500 km², 1000 km², and 2000 km² showed slightly lower R^2 scores and higher RMSE values. Specifically, the localized crop of 500 km² resulted in an R^2 score of 0.085, an RMSE of 131.97 seconds, and an F1-score of 0.093. The 1000 km² localized crop had an R^2 score of 0.087, an RMSE of 131.77 seconds, and an F1-score of 0.109. Finally, the 2000 km² localized crop yielded an R^2 score of 0.087, an RMSE of 131.81 seconds, and an F1-score of 0.095. These results indicate that while the localized cropping strategies did not significantly outperform the original model, they provide a more detailed perspective, with slight variations in performance metrics across different crop sizes.

Table 1 - Performance comparison of the original model and models with different cropping strategies. For F1-score and precision

Model	R^2 Score	RMSE (s)	F1-score	Precision
Original Crop	0.092	131.44	0.115	0.49
Localized Crop (500 km ²)	0.085	131.97	0.093	0.47
Localized Crop (1000 km ²)	0.087	131.77	0.109	0.48
Localized Crop (2000 km ²)	0.087	131.81	0.095	0.47

HYPOTHESES AND ANALYSIS

Hypothesis 1: Embedding Ineffectiveness and Potential Overfitting

One possible reason for the underperformance of the localized cropped models compared to the original model is that the vector embedding of the satellite images might not have captured significant differences between the smaller cropped regions. The embedding process may not have added substantial predictive power, possibly leading to a situation where the model overfits to irrelevant features rather than focusing on meaningful meteorological patterns. This lack of improvement suggests that the embedding strategy might need refinement, or that the selected features do not contribute as expected to the prediction of flight holding maneuvers.

Hypothesis 2: Need for Temporal and Operational Context

Another hypothesis is that the model's current approach may benefit from incorporating additional temporal and operational context. For example, integrating historical weather data from a time window preceding the flight, rather than a single static image, could provide a more comprehensive view of the evolving weather conditions. Additionally, incorporating operational data such as airport capacity and the demand for aircraft landings per hour might better account for the logistical factors influencing holding maneuvers. These enhancements could help the model better understand the full context of each flight scenario, potentially leading to more accurate predictions.

These hypotheses suggest directions for further research, including refining the image embedding process, testing alternative machine learning models,

and expanding the dataset to include more diverse features that capture both meteorological and operational factors.

CONCLUSION

The main goal of this research was to enhance the prediction of flight holding maneuvers by refining a model that uses meteorological data and satellite images. A new approach was introduced that dynamically crops satellite images around the destination airport, with tests conducted on crop sizes of 500 km², 1000 km², and 2000 km². The methodology covered data handling, image processing, feature extraction, model training, and evaluation, with a comparison of the localized cropping approach against the original static model.

Although the localized cropping approach did not show significant improvement over the original model, the study identified areas for potential enhancement. These include refining image embedding techniques and incorporating additional context, such as historical weather data and airport capacity, to improve predictions.

Future work could focus on exploring these enhancements in more depth, as well as testing alternative machine learning models. Expanding the dataset to include more diverse features that capture both meteorological and operational factors could also provide valuable insights and lead to more accurate models. Additionally, investigating the impact of different temporal windows for weather data could offer further improvements in predicting flight holding maneuvers.

Acknowledgments

This work was made possible by the support and guidance of Filipe Verri, who organized the *Data Science Challenge at EEF 2024* [Verri 2024], which provided the inspiration and foundation for the models used in this research. I would also like to acknowledge the contributions of Manoel and Jorge, whose work on the baseline model during the challenge was instrumental in shaping this study. Their efforts have significantly contributed to advancing the field of flight operations prediction. The original code for this work can be found at [Ryukinix 2024].

REFERENCES

Addu, S., Ambati, P. R., Kondakalla, S. R., Kunchakuri, H., and Thottempudi, M. (2022). Predicting delay in flights using machine learning. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 374–379.

Kalyani, N. L., Jeshmitha, G., U., B. S. S., Samanvitha, M., Mahesh, J., and Kiranmayee, B. V. (2020). Machine learning model - based prediction of flight delay. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 577–581.

Ryukinix (2024). Data science challenge 2024 repository. <https://github.com/ryukinix/dsc2024>.

Sharan, S., Sriniketh, M., Vardhan, H., and Jayanth, D. (2021). State-of-art machine learning techniques to predict airlines delay. In *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, pages 1–6.

Verri, F. (2024). Data science challenge at eef 2024.

ENHANCING REAL-TIME OPTIMIZATION WITH MACHINE LEARNING PREDICTIONS OF LOWER CALORIFIC VALUES

Filipe Andersonn Teixeira da Silveira

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)
filipeandersonn@gmail.com

Elcio Hideiti Shiguemori

Specialization Course in Data Science - Aeronautics Institute of Technology (ITA)

ABSTRACT

This paper presents an enhancement to a Real-Time Optimization (RTO) strategy for industrial furnaces by integrating machine learning techniques. The developed methodology utilizes historical process data to build predictive models that estimate the LCV, enabling more effective optimization of the fuel gas mixture, composed of different fuel gases. Implementing this strategy resulted in a significant reduction in the production cost without compromising the quality of the heating process. The results demonstrate the potential of integrating machine learning models into Real-Time Optimization systems, contributing to cost reduction and increased sustainability in the industry.

INTRODUCTION

Continuous combustion industrial furnaces play a critical role in various industrial operations where precise control of the heating process is essential to achieve the desired product properties. Achieving this requires precise control of the product temperature, which is managed by regulating the furnace conditions. Since the 1970s, the challenges associated with the modeling and control of continuous industrial furnaces have garnered significant interest [Yang and Lu 1988, Qiu *et al.* 2024]. Two primary factors drive this focus: the first being the intricate nature of temperature control, due to the complexity inherent in the process [Yang and Lu 1986, Qiu *et al.* 2024]; the second being the substantial energy consumption associated with these furnaces in large-scale production [McBrien *et al.* 2016, Caiza *et al.* 2017].

The process of heating the material relies on energy absorption from the furnace atmosphere, which introduces several control challenges, including managing temperature, airflows, and fuel distribution across different furnace zones. Moreover, the process complexity is exacerbated by the variability in the materials being processed in some industries, where each material type demands a specific temperature profile throughout the heating process, necessitating precise control of temperature across various furnace zones [Tudon-Martinez *et al.* 2022].

As a result, developing effective control systems and implementing advanced control strategies, such as optimal control for industrial furnaces, presents a significant challenge. This is due to the inherent complexity of the process, which includes nonlinear behaviors, considerable time delays, high time constants, and the presence of disturbances and uncertainties [Zhang *et al.* 2002].

Furthermore, industrial furnaces represent one of the most energy-intensive components in many manufacturing processes. These furnaces account for a significant portion of the total energy consumption in production facilities [Hua-Li *et al.* 2014, Guangjun 2008]. Therefore, reducing the energy consumption of these furnaces remains a critical issue in control and optimization research. Previous studies on energy efficiency in industrial furnaces have predominantly focused on controlling the combustion process, analyzing the mechanisms of heat transfer, and developing optimization dispatch models [Lu *et al.* 2017].

Considering the strategy applied by [Silveira and Raffo 2017], where a significant reduction in the consumption of fuel gas was achieved, this paper presents an enhancement obtained in the Real-Time Optimization (RTO) using machine learning techniques to predict the starting point of the lower calorific value (LCV) to increase energy efficiency. The application of machine learning allows for more accurate and adaptive predictions by analyzing complex patterns and relationships in the historical furnace data, which are not easily captured by traditional optimization methods. This leads to more precise control adjustments, minimizing fuel consumption and reducing energy waste. To achieve that, historical time series furnace data is used to build a model that can predict and forward the LCV to the optimal control strategy as a starting point for optimization.

Contextualization

Considering the continuous heating process and its characteristics defined in [Silveira and Raffo 2017], the furnace has 8 zones distributed across 3 units. These furnaces are used to perform heat treatment on materials. To accomplish this, the energy required for the heat treatment must be delivered through gas burners distributed along the zones and furnaces. The gas used is a mixture of NG (Natural Gas) and BFG (Blast Furnace Gas). A mixing station controls the ratio of these gases according to the LCV (Lower Calorific Value) specified in operational practice. The blend is adjusted by controlling the flow of NG mixed with BFG.

The furnace system includes an LCV adjustment of the mixed gas, composed of blast furnace gas enriched with natural gas, which determines the fuel gas's burning power without considering its condensation. The mixing station optimizes the use of BFG, which is produced in the blast furnace process. Therefore, the objective of this station is to optimize the LCV to consume as little NG as possible, which has a high cost, without harming the quality of the treated material. That is, NG must be used when BFG is scarce and/or as energy complementation depending on the operation rate of the process.

To increase the energy efficiency of this process and consequently produce at the lowest possible cost, [Silveira and Raffo 2017] proposed an optimal control strategy, through a technique of Real-Time Optimization for the fuel gases, defining

an optimization problem as shown in 1 and 2. This strategy aims to maximize the use of BFG, which does not have a significant cost, while minimizing NG consumption.

Minimize:

$$\begin{aligned} \delta e_{\min(q_{mg_i}, lcv_{mg})} = & |(\gamma \cdot q_{\max_i} - q_{mg_i}) \cdot lcv_{mg}| \\ \text{w.r.t. } & lcv_{mg} \end{aligned} \quad (1)$$

$$\text{s.t.: } \begin{cases} \text{Furnace Pressure} \leq p_{\max} \text{ mmCA} \\ \text{BFG flow} \leq bfg_{\max} \text{ Nm}^3/\text{h} \\ lcv_{\min} \leq \text{LCV} \leq lcv_{\max} \text{ kcal/Nm}^3 \end{cases} \quad (2)$$

where $\delta e_{\min}$ is the function representing the minimum increment of heat transfer rate among zones to be minimized, taking into account the established percentage of the maximum zone flow (γ), the maximum flow rate ($q_{\max_i}$) of zone i , the gas flow of the current mixture of zone i (q_{mg_i}), and the lower calorific value of the mixed gas (lcv_{mg}). Additionally, $p_{\max}$ denotes the maximum furnace pressure, $bfg_{\max}$ is the maximum BFG flow, and $lcv_{\min}$ and $lcv_{\max}$ represent the lower and upper limits of the LCV, respectively.

Figure 1 - LCV based on the NG and BFG gas flow.

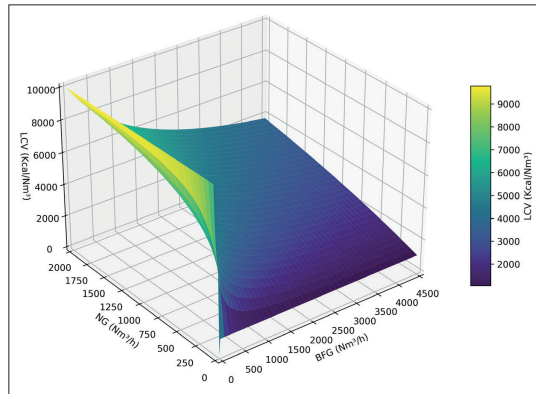


Figure 1 graphically presents the relationship between the LCV of the fuel gas mixture and the NG and BFG flow rates. It can be seen that by minimizing

the LCV of the gas mixture, the NG flow rate is reduced and, consequently, the production cost.

Research Problem

The primary challenge addressed in this research is the optimization of the Lower Calorific Value (LCV) in continuous combustion furnaces. These furnaces, which are critical for maintaining the desired properties of materials during industrial heating processes, pose significant control and optimization challenges due to their inherent complexity and high energy consumption.

In the current operational framework defined in [Silveira and Raffo 2017], the LCV, which is a measure of the energy content of the fuel gas mixture, is adjusted manually based on operator experience and standard practices. This approach, while functional, is suboptimal as it does not fully account for the dynamic nature of the furnace environment, where variations in material type, width, thickness, and production speed continuously influence the required energy input.

Furthermore, while Real-Time Optimization systems have been implemented to enhance energy efficiency by adjusting the fuel mix dynamically, these systems still rely on initial setpoints that may not be optimal. The process of adjusting from these sub-optimal initial conditions to an optimized state can result in delayed energy savings and inconsistent performance.

The research problem, therefore, revolves around the need to predict the optimal initial LCV setpoint using historical furnace data and machine learning techniques. By accurately predicting the LCV starting point that corresponds to optimal fuel efficiency, the transition period during which the RTO system fine-tunes the fuel mix can be minimized. This would lead to more immediate and consistent energy savings, particularly in reducing natural gas (NG) consumption, which is the more expensive component of the fuel mix.

In summary, the research problem is centered on enhancing the Real-Time Optimization of continuous combustion furnaces by developing a predictive model that accurately determines the optimal LCV setpoint. This approach is expected to improve the efficiency of the RTO system, leading to reduced energy costs and enhanced operational performance.

General Target

This research aims to enhance continuous combustion furnaces' energy efficiency and operational performance by integrating machine learning techniques into the RTO framework. This study aims to leverage historical process data to develop predictive models that estimate the optimal Lower Calorific Value (LCV) setpoints. By providing the RTO system with these optimized starting points, the goal is to reduce the consumption of high-cost natural gas (NG), maximize the use of blast furnace gas (BFG), and maintain the quality of the annealing process.

The proposed approach addresses the limitations of the current operational practices, where LCV setpoints are manually determined and may not be optimal. The research introduces a data-driven methodology that allows for dynamic optimization of the fuel mix, ensuring that the furnaces operate at their most energy-efficient state from the outset. By integrating these predictive models into the RTO system, this study seeks to achieve significant cost savings, reduce environmental impact through lower fossil fuel consumption, and improve consistency in product quality.

Specific Target

The specific targets of this research are focused on the development, implementation, and validation of a machine learning-based approach to optimize the operation of continuous combustion furnaces. These targets are outlined as follows:

1. **Data Collection and Preprocessing:** Collect and preprocess historical time series data from the furnace operations, including variables such as material type, temperature setpoints in various furnace zones, material dimensions, LCV settings, production speed, and the state of the RTO system. Ensure that the dataset is clean, comprehensive, and ready for model training.
2. **Feature Engineering:** Identify and engineer relevant features from the collected data that significantly impact the furnace's energy efficiency and operational performance. This includes deriving new features that enhance the predictive power of the machine learning models, such as aggregated temperature profiles or time-lagged variables.

3. **Model Development:** Develop and train machine learning models to predict the optimal LCV setpoints based on the engineered features. Explore various algorithms, including regression models, decision trees, and neural networks, to determine the most accurate and robust model.
4. **Hyperparameter Tuning:** Implement GridSearchCV for the regression models and Keras Tuner for the Neural Network to optimize the hyperparameters of each model, ensuring that the models are fine-tuned for the best performance.
5. **Model Validation and Testing:** Validate the performance of the developed models using appropriate metrics such as Root Mean Squared Error (RMSE) and R-squared. Test the models on unseen data to ensure their generalizability and reliability in real-world scenarios.
6. **Integration with RTO System:** Integrate the selected predictive model into the existing RTO framework of the furnace. This involves developing an interface that allows the model's predictions to be used in real-time to dynamically adjust the LCV setpoints.

Each of these targets is designed to contribute incrementally towards achieving the overarching goal of enhancing the energy efficiency and operational performance of continuous combustion furnaces through the application of machine learning techniques.

METHODOLOGY

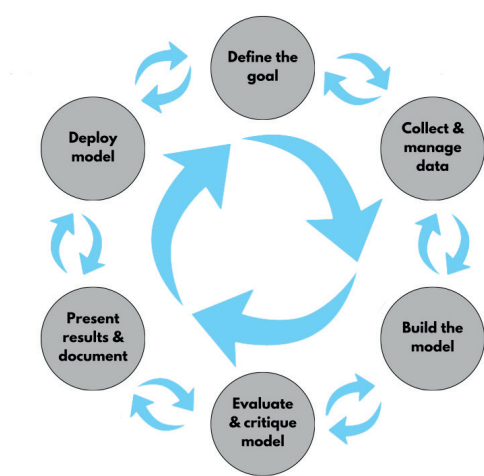
Following the lifecycle of a data science project as proposed by [Mount and Zumel 2019], shown in Figure 2, the development of this work is structured into the following steps: define the goal, collect and manage data, build the model, and experiments and results.

Define the Goal

The goal of this project is to enhance the Real-Time Optimization of continuous combustion furnaces by predicting the Lower Calorific Value (LCV) using machine learning models. This prediction will serve as an input for optimizing the fuel mix, thus reducing the consumption of natural gas and improving energy efficiency.

Since the RTO requires time to adjust from the initial setpoint defined by the operator to the optimal LCV, it was observed that starting with an optimal LCV could enhance the NG reduction, moving from the percentage reduction observed at "RTO Reduction" to the one at "Final Reduction" shown in Table 1. Therefore, this work aims to leverage historical data to train machine learning models capable of predicting this initial setpoint,

Figure 2 - The lifecycle of a data science project [Mount and Zumel 2019]



allowing the system to begin operation closer to the optimal LCV. This approach is expected to further reduce energy consumption, aligning the results more closely with the "Final Reduction" values.

Table 1 - Natural Gas Consumption and Reduction Results.

Start Date	End Date	Material	NG consump. OFF	NG consump. ON	Final NG consumption	RTO Reduction	Final Reduction
10/03/2024 16:00	10/03/2024 18:20	Material A	456 Nm3/h	426 Nm3/h	411 Nm3/h	6,5%	9,8%
10/03/2024 18:50	10/03/2024 19:50	Material B	580 Nm3/h	541 Nm3/h	529 Nm3/h	6,7%	8,8%
11/03/2024 14:00	11/03/2024 17:30	Material C	415 Nm3/h	394 Nm3/h	390 Nm3/h	5,0%	6,0%
23/03/2024 12:45	23/03/2024 15:15	Material D	908 Nm3/h	860 Nm3/h	828 Nm3/h	5,2%	8,8%

Data Collection and Management

The process data for the furnace were collected from a time series data historian, covering the period from 2024-01-24 16:30:00 to 2024-04-23 11:25:00, to build the dataset. The following variables were considered in the dataset:

- **material_type** (categorical): indicates the type of material being processed in the furnace.
- **tic_sp_z1** to **tic_sp_z8** (numeric): temperature setpoint from each of the eight furnace zones, representing the setpoint of each respective temperature control.
- **width** (numeric): the width of the material strip being processed.
- **thickness** (numeric): the thickness of the material strip.
- **LCV_oper** (numeric): the start point of Lower Calorific Value (LCV) set by the operator.
- **speed** (numeric): the speed at which the material strip moves through the furnace.
- **RTO_state** (binary): indicates whether the RTO system is active (1) or inactive (0).
- **LCV** (numeric): the real Lower Calorific Value (LCV) of the fuel mix being used in the furnace.

As each of these variables was collected and stored in the historian with different sampling rates, it was necessary to establish a consistent sampling period of 5 minutes to ensure that all variables had the same sampling rate for the dataset construction. This uniform sampling rate allows for synchronized data points across all variables, facilitating accurate analysis and model development.

Given that the objective is to build a dataset for training a predictive model aimed at determining the starting setpoint of the LCV, considering the performance of the RTO during its active periods to maximize the reduction in natural gas (NG) consumption, data from periods when the RTO was not active were excluded from the dataset. This exclusion was necessary to prevent contamination of the model with data that do not reflect the conditions under which the RTO system was operational.

Furthermore, data from periods when the furnace was either turned off or running without producing "useful" material were also removed from the dataset. This step ensures that the model is trained only on data relevant to the production process, avoiding any potential bias from irrelevant operational states.

Exploratory analysis

An initial descriptive analysis was conducted on the dataset to understand the distribution and characteristics of the collected data. This analysis included both numerical and categorical features.

Table 2 presents a statistical summary of the numerical features in the dataset, with each feature summarized across key statistical metrics such as count, mean, standard deviation, minimum, maximum, and percentiles. This summary provides insight into the distribution and variability of the temperature setpoints across the eight furnace zones (tic_sp_z1 to tic_sp_z8), as well as the physical characteristics of the material being processed (width, thickness), and the operational parameters (LCV_oper , speed , LCV).

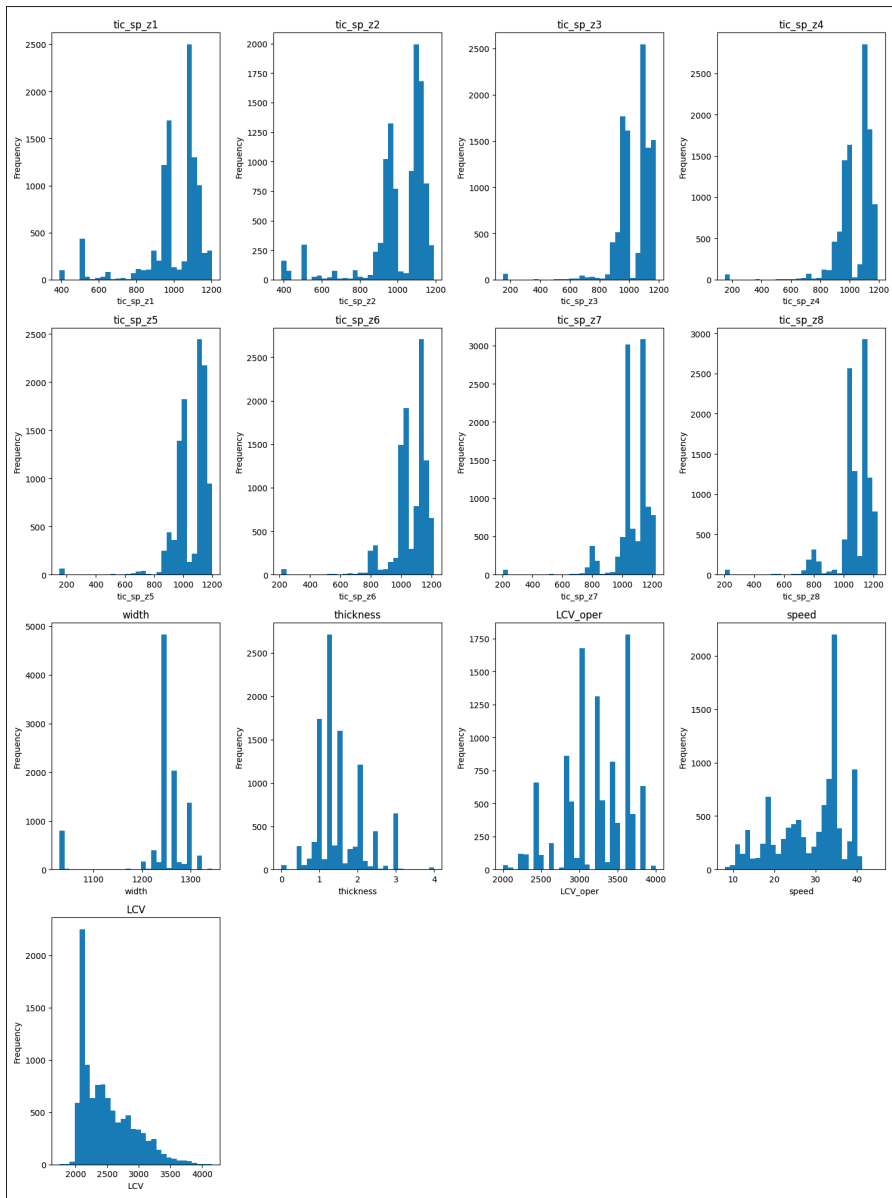
Table 2 - Statistical Summary of Numerical Features.

Statistic	Count	Mean	Std	Min	25%	50%	75%	Max
tic sp z1	10323	1002.90	159.56	390.00	950.00	1070.00	1110.00	1205.00
tic sp z2	10323	1008.25	163.97	386.00	950.00	1080.00	1120.00	1191.00
tic sp z3	10323	1039.75	120.19	150.00	960.00	1090.00	1125.00	1180.00
tic sp z4	10323	1044.02	121.93	150.00	970.00	1100.00	1130.00	1190.00
tic sp z5	10323	1055.88	119.13	150.00	986.50	1108.00	1130.00	1200.00
tic sp z6	10323	1063.94	120.01	211.00	1015.00	1095.00	1135.00	1218.00
tic sp z7	10323	1072.36	121.86	204.00	1035.00	1090.00	1145.00	1225.00
tic sp z8	10323	1077.64	124.36	204.00	1050.00	1085.00	1155.00	1226.00
width	10323	1240.41	63.85	1030.00	1240.00	1240.00	1270.00	1345.00
thickness	10323	1.51	0.61	0.40	1.19	1.20	1.98	4.00
LCV oper	10323	3180.57	415.07	2000.00	2900.00	3200.00	3600.00	4000.00
speed	10323	28.79	8.27	8.00	22.00	32.00	35.00	45.00
LCV	10323	2487.33	405.48	1744.00	2134.00	2384.00	2756.00	4161.00

Figure 3 illustrates the distribution of numerical attributes, further detailing the spread and concentration of values across different features. This visual

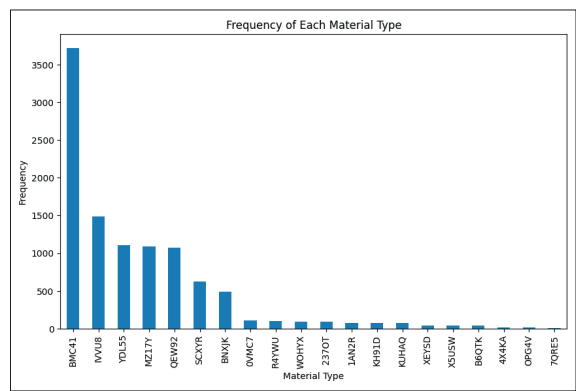
representation complements the statistical summary by highlighting how the data is distributed, allowing for a better understanding of potential outliers or skewness in the dataset.

Figure 3 - Numerical attributes distribution.



In addition to the numerical data, the dataset *also* includes categorical variables. The variable `material_type` captures the different types of materials processed during the data collection period. Figure 4 shows the count of unique values for each material type in the dataset, providing insights into the distribution of different materials processed. It is important to highlight that there is a significant imbalance in the distribution of `material_type`, with certain types, such as BMC41, being much more frequent than others. This imbalance may introduce bias into the predictive modeling process. However, it should be noted that this observed imbalance is a reflection of the production data collected over a 3-month period, during which production was naturally skewed towards certain material types, a common occurrence in industrial operations.

Figure 4 - Distribution of materials processed.



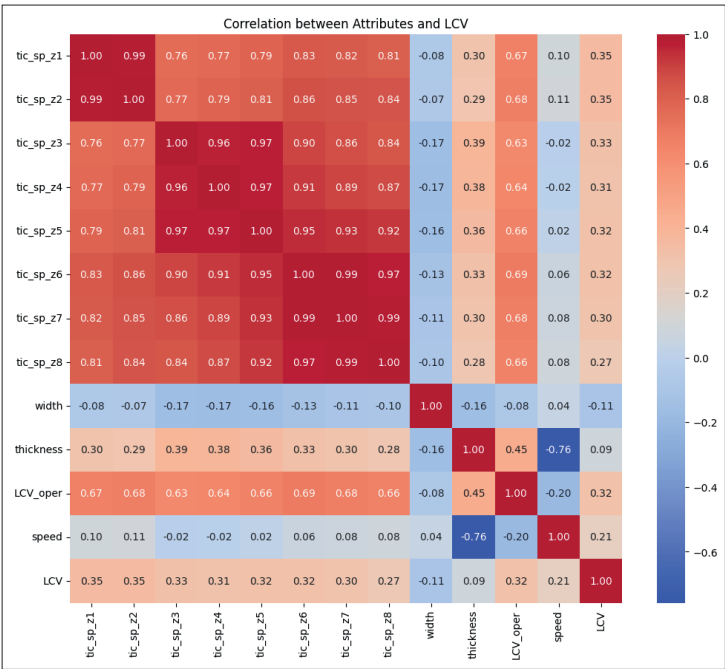
Furthermore, a correlation analysis was performed to explore the relationships between the numerical variables, as shown in Figure 5. Notably, there is a moderate positive correlation between `LCV` and the `tic_sp_z1`, `tic_sp_z2`, and `LCV_oper`, suggesting that these variables have a significant impact on the `LCV` value. Also, the correlation heatmap reveals strong positive correlations between the temperature setpoints across different furnace zones (`tic_sp_z1` to `tic_sp_z8`), indicating that these temperatures tend to vary together across the zones. This high degree of correlation among the temperature setpoints suggests the presence of multicollinearity, which could potentially inflate the variance of regression coefficients and lead to unreliable statistical inferences. Addressing this multicollinearity will be a critical step in the model development process to ensure the accuracy and stability of the predictive models.

The exploratory analysis highlights the inherent variability and complexity of the dataset, which is critical in developing a robust predictive model. The numerical summary provides detailed insights into the distributions and variability of temperature set-points, material characteristics, and operational parameters, offering a foundation for feature selection and model tuning. Additionally, the categorical analysis reveals significant imbalances in material types, possibly introducing bias into the model if not properly addressed. Furthermore, the strong multicollinearity observed among the temperature setpoints across different furnace zones indicates the need for careful feature engineering and potential dimensionality reduction techniques to ensure the stability and accuracy of the predictive models.

Building the Model

The process of building the predictive model involved critical steps, including data preparation, pre-processing, model selection, and evaluation. Below, we detail each of these steps and the methodologies employed.

Figure 5 - Correlation between attributes and LCV.



Data Preparation

The dataset was divided into features (X) and the target variable (y), which is the LCV (Lower Calorific Value). The features included both numerical variables, such as temperature setpoints and material characteristics, and categorical variables, such as the type of material processed. The data was then split into training and testing sets using an 80-20 split to ensure that the model could be effectively trained and evaluated.

Pre-Processing

Given the complexity of the data, a comprehensive pre-processing pipeline was developed. The numerical features (`tic_sp_z1` to `tic_sp_z8`, `width`, `thickness`, `speed`) underwent several transformations, including the exclusion of observations with missing values and standardization to ensure all features were on a comparable scale. Additionally, as a high degree of multicollinearity was identified among the temperature setpoints across different furnace zones, a dimensionality reduction via Principal Component Analysis (PCA) was employed. PCA transforms these correlated variables into a set of uncorrelated principal components, thereby stabilizing model estimates and improving interpretability. The PCA was configured to retain 95% of the variance in the data, significantly reducing the dimensionality while preserving most of the information critical for accurate predictions.

The categorical feature `material_type` was handled separately, with missing values being imputed with a constant placeholder, followed by encoding using one-hot encoding. This approach allowed us to manage categorical variables effectively, ensuring they were properly integrated into the modeling pipeline.

Figure 6 - Correlation Matrix of Principal Components After PCA.

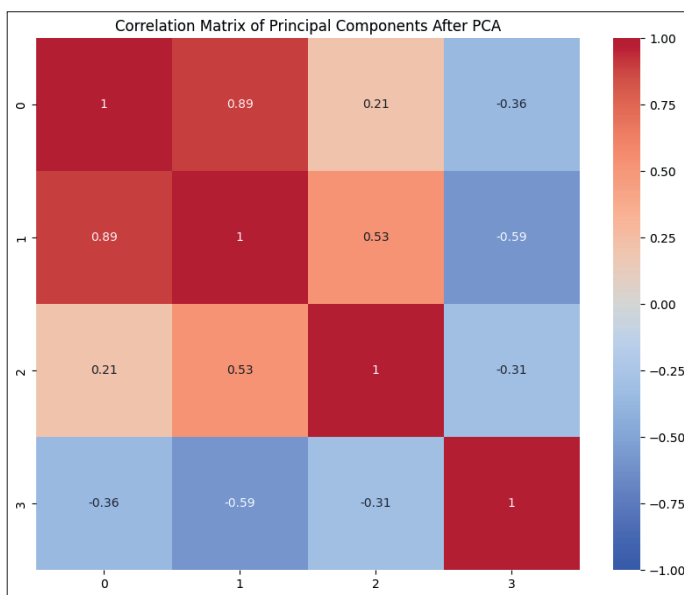


Figure 6 illustrates the correlation matrix of the principal components after applying PCA. During the pre-processing phase, the PCA retained 4 principal components, which collectively explained 95% of the variance in the dataset. This dimensionality reduction was critical for improving the model's efficiency and interpretability, without sacrificing significant predictive power.

Model Selection and Evaluation

To identify the most effective predictive model, we evaluated several regression algorithms, including Linear Regression, Ridge Regression, Polynomial Regression, Random Forest Regressor, Gradient Boosting Regressor, and Neural Network. The selection of these models was driven by a strategy aimed at finding the simplest possible model that could be effectively implemented within the constraints of a limited platform. The models were chosen for the following reasons:

- **Linear Regression:** Included as a baseline model due to its simplicity and ease of interpretation, providing a straightforward comparison against more complex models.

- **Ridge Regression:** Selected to account for potential multicollinearity in the features while maintaining the simplicity of a linear model, with added regularization to improve generalization.
- **Polynomial Regression:** Considered to capture potential nonlinear relationships in the data while keeping the model complexity manageable.
- **Random Forest Regressor:** Chosen for its ability to handle complex, non-linear patterns in the data, offering robustness through ensemble learning.
- **Gradient Boosting Regressor:** Selected for its strong performance in modeling complex relationships by combining the strengths of multiple weak learners.
- **Neural Network:** Evaluated for its flexibility and potential to model intricate relationships, as well as its relative ease of implementation on modern platforms, despite its inherent complexity.

Each model was embedded into a pipeline that included the pre-processing steps, ensuring that the transformations were consistently applied across all models.

For optimizing the hyperparameters of these regression models, we utilized Grid-SearchCV. This allowed us to systematically explore a predefined hyperparameter space for each algorithm. For example, we adjusted the regularization strength for Ridge Regression, the polynomial degree for Polynomial Regression, and the number of estimators and feature selections for Random Forest and Gradient Boosting Regressors. The Grid-SearchCV framework ensured that the optimal set of hyperparameters was selected based on cross-validated performance metrics.

For the Neural Network model, we employed the Keras Tuner to optimize the architecture and hyperparameters. This involved adjusting the number of layers, the number of units per layer, activation functions, dropout rates, and learning rates. The tuner searched across a predefined hyperparameter space to identify the configuration that minimized validation loss.

We employed 5-fold cross-validation to rigorously assess model performance, focusing on the Root Mean Squared Error (RMSE) and R-squared (R^2) metrics. These metrics provided insights into the accuracy and explanatory power of the models, respectively.

EXPERIMENTS AND RESULTS

In this section, we present the results obtained for the various models evaluated, including Linear Regression, Ridge Regression, Random Forest Regressor, Gradient Boosting Regressor, and a Neural Network model. Table 3 summarizes the performance of each model based on the Root Mean Squared Error (RMSE) and R-squared (R^2) metrics on the test data.

Table 3 - Model Comparison.

Model	RMSE (Test)	R^2 (Test)
Random Forest Regressor	158.45	0.85
Gradient Boosting Regressor	173.03	0.82
Neural Network	221.65	0.71
Polynomial Regression	320.89	0.38
Linear Regression	349.93	0.26
Ridge Regression	350.04	0.26

The results show that the **Random Forest Regressor** was the best-performing model, achieving a test RMSE of 158.45 and a test R^2 of 0.85. The Gradient Boosting Regressor also performed well, with a test RMSE of 173.03 and R^2 of 0.82. The Neural Network model, though not the best, still showed reasonable performance with a test RMSE of 221.65 and R^2 of 0.71 after significant hyperparameter optimization.

For the Polynomial Regression model, a degree of 2 was found to be optimal, resulting in a test RMSE of 320.89 and R^2 of 0.38. However, the performance was inferior compared to the more complex models, indicating that while a polynomial relationship exists, it was not sufficiently captured by this model alone.

Linear Regression and Ridge Regression models showed the lowest performance, with both models having a test RMSE of 349.93 and 350.04 respectively, and R^2 of 0.26. These results indicate that more complex models, such as Random Forest and Gradient Boosting, are more effective at capturing the nonlinear relationships present in the data.

Optimal Hyperparameters

During the model training process, optimal hyperparameters were identified for each model to enhance their performance:

- **Ridge Regression:** The best-performing alpha parameter was found to be 1.0. This regularization parameter controls the strength of the regularization; a value of 1.0 indicates moderate regularization, balancing bias and variance.
- **Polynomial Regression:** The optimal polynomial degree was found to be 2, indicating a quadratic relationship between the features and the target variable.
- **Random Forest Regressor:** Optimal parameters included setting max features to 'sqrt' and using 500 estimators. The max features parameter limits the number of features considered at each split, reducing overfitting, while the number of estimators determines the number of trees in the forest, enhancing model robustness.
- **Gradient Boosting Regressor:** The learning rate was optimized to 0.2, and the number of estimators was set to 500. The learning rate controls how quickly the model adapts to the problem, and the number of estimators balances the trade-off between bias and variance, where more estimators reduce bias but increase computational cost.
- **Neural Network:** The optimal configuration involved a model with three layers, starting with 64 units in the first layer (using ReLU activation), 384 units in the second layer (with tanh activation), and 416 units in the third layer (again using ReLU activation). Dropout rates of 0.1, 0.4, and 0.1 were applied respectively to prevent overfitting, and the learning rate was set to 0.001 to ensure steady convergence.

The **Random Forest Regressor** was ultimately selected as the best model for predicting LCV, and its performance on the test data further confirmed its generalization capability, with a test RMSE of 158.45 and an R^2 of 0.85.

Error Analysis and Model Interpretation

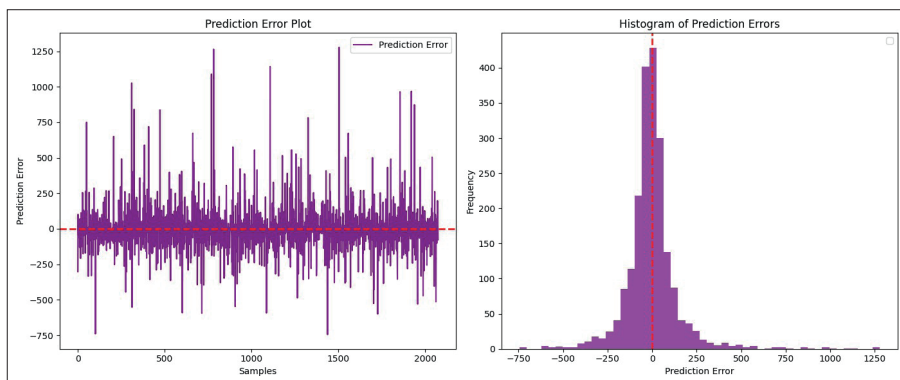
In addition to the evaluation metrics presented earlier, a detailed analysis of the prediction errors for the **Random Forest Regressor**, which was the best-performing model, was conducted. The error analysis included plotting the prediction error across the test samples as well as examining the distribution of these errors.

Figure 7 illustrates two key visualizations: the prediction error plot and the histogram of prediction errors. The prediction error plot, shown on the left, displays the difference between the real and predicted values across all test samples. The majority of errors are centered around zero, with relatively few large deviations, indicating that the model consistently produces accurate predictions. However, some outliers with larger errors suggest areas where the model may struggle with specific patterns in the data.

On the right, the histogram of prediction errors provides further insight into the distribution of these errors. The majority of the errors are concentrated around zero, confirming that the Random Forest model generally performs well. The histogram reveals a roughly symmetric distribution, with a slight skew towards positive errors, indicating a tendency of the model to slightly underestimate the target variable in some cases.

These visualizations support the numerical results presented earlier, reinforcing the conclusion that the Random Forest Regressor is not only accurate in terms of RMSE and R^2 , but also demonstrates a consistent performance with relatively few significant errors. This consistency is crucial for practical applications where reliability and robustness of predictions are essential.

Figure 7 - Prediction Error Plot and Histogram of Prediction Errors for the Random Forest Regressor.



CONCLUSION AND FUTURE WORK

This study has successfully demonstrated the potential of integrating machine learning models into the Real-Time Optimization process for continuous

combustion furnaces. By predicting the Lower Calorific Value using the best-performing machine learning model, we will be able to propose a strategy that optimizes the fuel mix, ultimately leading to significant improvements in energy efficiency.

The results indicate that the *Random Forest Regressor* was the most effective model, with an RMSE of 158.45 and an R^2 of 0.85, outperforming other tested models. This model can be effectively deployed within the current system constraints, offering a practical and efficient solution to optimize the LCV setpoint. Its implementation is expected to enhance furnace efficiency, reduce natural gas consumption, and contribute to substantial cost savings—potentially around 345,000 BRL per year—while promoting more sustainable practices in the industry. This annual cost reduction is projected based on the data observed in the tests presented in Table 1 and a production forecast.

However, this study has some limitations, and for future work, the following directions are recommended:

- Explore the application of deep learning models with more complex architectures to better capture the nonlinear relationships present in the data.
- Expand data collection to include a greater diversity of material types and operational conditions, ensuring greater robustness and applicability of the predictive models across different industrial contexts.
- Implement online learning techniques to adjust the models in real time as new information and operating conditions become available, increasing the system's adaptability.

In conclusion, this work reinforces the importance of integrating machine learning techniques into Real-Time Optimization of industrial processes, demonstrating the potential of these approaches to promote more efficient and sustainable production.

REFERENCES

- Caiza, L., Rosich-Oliva, A., Ocampo-Martínez, C., and Benítez, D. S. (2017). Reducing energy consumption in an industrial process by using model predictive control. In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pages 1–6. IEEE.
- Guangjun, C. (2008). The problem and energy-saving of reheating furnace. *Energy Metall. Ind.*, pages 32–35.
- Hua-Li, K., Bin, D., and Bo, Y. (2014). Research and application of slab heating curve in reheating furnace. *Metall. Ind. Autom.*, pages 50–55.
- Lu, B., Chen, D., Chen, G., and Yu, W. (2017). An energy apportionment model for a reheating furnace in a hot rolling mill - a case study. *Applied Thermal Engineering*, pages 174–183.
- McBrien, M., Serrenho, A. C., and Allwood, J. M. (2016). Potential for energy savings by heat recovery in an integrated steel supply chain. *Applied Thermal Engineering*, pages 592–606.
- Mount, J. and Zumel, N. (2019). *Practical data science with R*. Simon and Schuster.
- Qiu, J.-q., Feng, J.-x., Huang, X.-m., and Huang, Z.-f. (2024). Optimization of reheating furnace rolling delay strategies based on a minimum energy consumption principle. *Journal of Iron and Steel Research International*, pages 1–13.
- Silveira, F. A. and Raffo, G. V. (2017). Optimal control of fuel gases on continuous annealing furnaces of steel coil strip.
- Tudon-Martinez, J. C., Lozoya-Santos, J. d.-J., Cantu-Perez, A., and Cardenas-Romero, A. (2022). Advanced temperature control applied on an industrial box furnace. *Journal of Thermal Science and Engineering Applications*, 14(6):061001.
- Yang, Y. and Lu, Y. (1986). Development of a computer control model for slab reheating furnaces. *Computers in Industry*, 7:145–154.
- Yang, Y. and Lu, Y. (1988). Dynamic model based optimization control for reheating furnaces. *Computers in Industry*, 10:11–20.
- Zhang, B., Cheng, Z., Xu, L., Wang, J., Zhang, J., and SHAO, H. (2002). The modeling and control of a reheating furnace. *Proceedings of the American Control Conference, May, 8-10*, pages 3823–3828.

MODELO PREDITIVO PARA IDENTIFICAÇÃO DE OPORTUNIDADES DE FINANCIAMENTO IMOBILIÁRIO EM INSTITUIÇÕES FINANCEIRAS

Marcelo Kawate

Curso de Especialização em Data Science – Instituto Tecnológico de Aeronáutica (ITA)

Elton Felipe Sbruzzi

Divisão de Ciência da Computação – Instituto Tecnológico (ITA)

ABSTRACT

The availability of reliable granular data presents a significant challenge for studies involving the real estate market. This paper aims to develop a model for identifying potential real estate financing in Brazilian financial institutions. Given the challenging economic landscape, which affects credit availability and the real estate market, the research focuses on creating a model capable of identifying geographic regions with the highest potential for financing. By employing time series, regression analysis, and other quantitative modeling techniques, the study seeks to enhance the allocation of commercial resources and optimize the prospecting of new business opportunities. The model is tested using economic, demographic, and market data collected from public and private sources.

RESUMO

A disponibilidade de dados confiáveis em granularidade é um grande desafio para estudos envolvendo o mercado imobiliário. Este artigo busca encontrar um modelo de identificação de oportunidades para financiamento imobiliário em instituições financeiras brasileiras. Dado o cenário econômico desafiador, que impacta a concessão de crédito e o mercado imobiliário, a pesquisa se propõe a desenvolver um modelo capaz de identificar regiões geográficas com maior potencial para contratação de financiamento. Utilizando séries temporais, análise de regressão e outras técnicas de modelagem quantitativa, o estudo busca melhorar a alocação comercial e otimizar a prospecção de novos negócios, a partir de testes com dados econômicos, demográficos e de mercado coletados de fontes públicas e privadas.

INTRODUÇÃO

No Brasil, a aquisição imobiliária é uma das maiores decisões em valor investido pela população em geral. O contexto macroeconômico influencia desde a captação dos recursos pelas instituições financeiras que fazem as concessões, passando pelo mercado de construção e mão-de-obra, que influencia preços, até chegar no valor pago mensalmente pelos mutuários. Em um cenário de restrições de rentabilidade pelas instituições, a busca pela acuracidade comercial passa a ser fundamental para a busca de clientes que se adequem ao apetite de crédito. Ferramentas comerciais que apontem isso de forma assertiva, ainda que difíceis de serem encontradas, podem conferir ganhos relevantes de receita, produtividade e margens nos resultados.

Contextualização

O objetivo da casa própria é um misto de sonho e desafio que atinge boa parte da sociedade brasileira, além de coexistir uma relação da aquisição da moradia com qualidade de vida e bem-estar de toda uma população [Andreoli; Mariano; Maruyama 2014]. Para uma grande parcela das pessoas, a compra de um imóvel é a maior decisão de investimento das suas vidas, em termos de valor empregado e estima pessoal. Apesar de toda esta relevância, atualmente o déficit habitacional no Brasil é de cerca de 5,8 milhões de moradias [Ministério do Desenvolvimento Regional; Fundação João Pinheiro 2021], incrementando ainda mais a discussão sobre o tema.

O contexto macroeconômico do país é fator fundamental para a disponibilidade de novos empreendimentos e por toda valorização do mercado de imóveis novos e usados, pois existe relação direta com a disponibilidade de crédito para construção e aquisição, além de refletir também no mercado de trabalho da construção civil. O mercado imobiliário e seus preços não seguem uma aleatoriedade, mas sim um parâmetro de comportamento cíclico [João 2017]. Desta forma, pressupõe-se que estudos econométricos podem contribuir na descrição de influências e por consequência, na observação de oportunidades projetadas.

Em virtude da aquisição imobiliária exigir aportes financeiros acima de uma disponibilidade média de liquidez que a maior parte dos brasileiros possui,

o financiamento imobiliário é o principal instrumento para a concessão de crédito para esta finalidade. Atualmente, o Brasil possui dois modelos vinculados ao *funding* das instituições financeiras voltados para a concessão de crédito imobiliário: o Sistema Financeiro da Habitação (SFH), que envolve recursos do Sistema Brasileiro de Poupança e Empréstimo (SPBE) e do Fundo de Garantia por Tempo de Serviço (FGTS); e o Sistema Financeiro Imobiliário (SFI), onde os recursos são originados no mercado de capitais, através da captação via mercado secundário de hipotecas [Marques 2014].

Com a redução constante no saldo de poupança, devido a mudanças no comportamento dos meios de investimentos por grande parte da população brasileira, há uma necessidade das instituições financeiras buscarem alternativas de *funding* que tragam a rentabilidade esperada nas operações pelos acionistas. Em 2021, a fatia do *funding* no mercado de capitais cresceu 34%, enquanto na poupança recuou 40% e no FGTS com menor oscilação, 26%. [ABECIP 2023]. Contudo, diferentemente da poupança que possui um custo de captação baixo para as instituições financeiras, o mercado de capitais tem como contrapartida a oferta de remuneração mais atrativa para os investidores, aumentando seu custo. Desta forma as instituições financeiras estão buscando maior acuracidade na sua concessão de crédito imobiliário, seja através da restrição com aumentos de taxas, seja com direcionamento para clientes e mercados mais vantajosos nos aspectos da visão institucional, onde uma menor rentabilidade do financiamento imobiliário é compensada com outras cestas de produtos e serviços mais vantajosos a serem ofertados. O proposto trabalho busca estudar e formular modelos para identificar oportunidades de melhor concessão de crédito imobiliário para instituições financeiras, avaliando as necessidades de cada mercado regional, relacionada com o apetite de crédito para os respectivos perfis de clientes.

Problema de Pesquisa

O problema a ser abordado relaciona-se com o atual cenário do financiamento imobiliário no Brasil, a partir da acuracidade na captação de clientes que possuem interesse em financiamento imobiliário, minimizando os custos de prospecção e mirando correntistas que possam trazer incrementos de receitas via aumento de relacionamento e *cross-sell* para as instituições financeiras. Desta

forma, o seguinte problema de pesquisa foi enunciado **“Como utilizar dados de relacionamento bancário e mercado imobiliário para identificação de potencial de Crédito Imobiliário e tomada de decisão quanto a alocação de times comerciais para prospecção do produto?”**.

Objetivo Geral

O objetivo da pesquisa é “desenvolver um modelo preditivo para identificar oportunidades de financiamento imobiliário, otimizando a alocação de recursos comerciais e a prospecção de novos negócios em instituições financeiras” visando reduzir custos de prospecção e incrementar receitas com aumento de relacionamento com os respectivos clientes assimilados.

Objetivos Específicos

Os objetivos específicos identificados a partir do objetivo geral são:

- Identificar fatores econômicos e regionais que influenciam a demanda por financiamento imobiliário;
- Validar variáveis econômicas e regionais que impactam a oferta de financiamento imobiliário;
- Mapear regiões com maior potencial de demanda por financiamento imobiliário, considerando as condições de mercado;
- Organizar as variáveis validadas e processar os dados disponíveis, demonstrando efetividade a partir da experimentação de dados via teste de modelos e previsão.

REVISÃO BIBLIOGRÁFICA

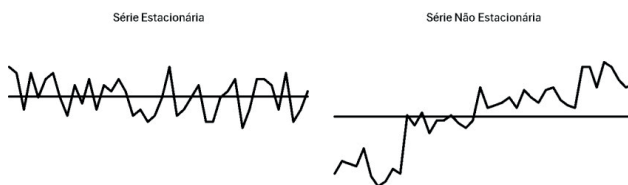
O mercado imobiliário é um dos mais suscetíveis a intempéries da macroeconomia que tem suas variáveis refletindo diretamente no cotidiano das pessoas. Desta forma, os dados econômicos são peças fundamentais para a observação de padrões de comportamento imobiliário. Em adicional, o mercado imobiliário tem por significado a definição de onde a população se concentra para viver e de forma

indireta, trabalhar e consumir. Logo, é prudente estudar os dados demográficos dos municípios, que é onde as pessoas de fato estão, para identificar possíveis relações analogias. Contudo, é notório que o Brasil ainda caminha de forma incipiente, sobretudo no poder público, quanto a divulgação e disponibilização de dados, em granularidades que permitam analisar de forma direta as relações das pessoas e da demanda com o mercado imobiliário.

Dados confiáveis para este mercado é um desafio hercúleo, pois há escassez e precariedade das informações registradas em prefeituras e cartórios. As bases atualmente disponibilizadas em ambientes virtuais têm restrições de tamanhos, históricos e variáveis, o que contribui muito para possíveis *overfits* nas análises [Ribeiro; Helena; Cipparone 2023]. Com esse contexto em mente, a utilização de modelos quantitativos de séries temporais tende a ser o ajuste mais adequado, visando o desafio no cenário apresentado, disponibilidade de dados e requinte na modelagem para previsão de demandas.

O objetivo das séries temporais é investigar e descrever o comportamento gerador dos dados, através de análises gráficas e testes para verificação de médias, tendências, ciclos e sazonalidades declaradas. Para isso é necessário testar os dados para certificação de que há um processo estocástico descrito e adequado para aplicação do modelo de previsão, ou seja, existem variáveis aleatórias que descrevem o comportamento de algum fato observado. Há dois tipos de processos que podem ser observados, os estacionários ou não estacionários. Uma série temporal estacionária é aquela que ao longo do tempo apresenta um comportamento equilibrado em torno da sua média ao longo do tempo. Já nas séries não estacionárias esse equilíbrio não é observado, como ilustra a Figura 1. Testes podem ser utilizados para comprovar a estacionariedade, como o *Argumented Dickey-Fuller* (ADF), com as seguintes hipóteses a serem analisadas: H_0 , tem raiz unitária e não é uma série estacionária; H_1 , não tem raiz unitária e é uma série estacionária [Moro 2017].

Figura 1 - Séries Estacionárias e Não Estacionárias.



Fonte: elaborado pelo autor.

Observando o comportamento dos dados é possível determinar qual melhor modelo de previsão de demanda pode ser treinado, testado e ajustado para verificação de observações futuras. É possível demonstrar em um quadro resumo as características dos principais modelos de previsão de demanda por séries temporais, como mostra a Tabela 1 [Bortoletto *et al.* 2016].

Tabela 1 - Modelos de previsão de demanda.

Método de Previsão	Descrição
Média Móvel Simples (MMS) e Média Móvel Ponderada (MMD)	Calcula a média dos últimos n períodos de uma série temporal. Adequado para conjuntos de dados muito curtos ou irregulares, onde características como tendência e sazonalidade não podem ser significativamente determinadas, e onde a média se mantém constante. Os modelos MMD reagem melhor quando as séries possuem um componente de tendência através de pesos destinados a suavizar esse comportamento.
Suavização Exponencial; Métodos de Holt-Winters	Usa processos de médias móveis, entretanto suavizam os componentes de nível, tendência e sazonalidade de séries temporais através de pesos destinados a suavizar esses componentes, geralmente dando mais peso aos dados mais recentes. Os métodos de Suavização Exponencial Simples são indicados para séries que possuem tendência, mas não possuem sazonalidade. O método de suavização de Holt é indicado quando a série possui somente tendência. Entretanto, o método de suavização Holt-Winters é indicado para séries que possuem tendência e sazonalidade.
Autorregressivos Integrados de Médias Móveis (ARIMA)	Uma classe mais complexa de métodos de médias móveis, capaz de refletir as autocorrelações inerentes aos dados observados. É um modelo mais robusto que os de suavização quando há um período longo de observações. São indicados para dados estacionários, onde através de diferenciações as séries temporais podem se transformar de não estacionárias em estacionárias. Constitui-se na agregação de termos autorregressivos (AR) e termos de médias móveis (MA), integrando as diferenciações da série caso haja necessidade (d). São indicados para séries que possuem tendência e sazonalidade.
Análise de Regressão	Ajustam uma reta aos dados históricos usando variáveis independentes e um termo de erro. São modelos considerados causais e não necessitam de análises profundas na série temporal a ser prevista.

Fonte: Bortoletto et al. 2016.

De forma objetiva, podem ser demonstradas as equações características de cada modelo, conforme Tabela 2.

Tabela 2 - Equações de modelos de previsão de demanda.

Método de Previsão	Equações
Média Móvel Simples (MMS)	$MMS = F_t = (Y_t + Y_{t-1} + \dots + Y_{t-n+1}) / n$
Média Móvel Ponderada (MMD)	$MMD = F_t = (\alpha Y_t + \beta Y_{t-1} + \dots + \gamma Y_{t-n+1}) / (\alpha + \beta + \gamma)$
Suavização Exponencial; Métodos de Holt-Winters (HW)	$L_t = \alpha(Y_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + b_{t-1})$
	$b_t = \beta(L_t - L_{t-1}) + (1 - \beta)b_{t-1}$
	$S_t = \gamma(Y_t - L_t) + (1 - \gamma)S_{t-s}$
	$HW = F_{t+m} = L_t + b_t m + S_{t-s+m}$
Autoregressivos Integrados de Médias Móveis (ARI-MA)	$AR = c + \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p}$
	$MA = c + \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_t$
	$I_1 = Y'_t = Y_t - Y_{t-1}$
	$I_2 = Y''_t = Y'_t - Y'_{t-1} = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})$
	$ARIMA = F_t = AR + I + MA$
Análise de Regressão	$Y_t = a + bx + \varepsilon_t$

Fonte: Bortoletto et al. 2016.

Onde:

- Y_t é o valor observado no período t ;
- F_t são os valores previstos para o período t ;
- α , β e γ são os pesos e/ou coeficientes das expressões;
- L_t é o componente do cálculo relativo ao nível da série;
- b_t é o componente do cálculo relativo à tendência da série;
- S_t é o componente do cálculo relativo à sazonalidade da série;
- s refere-se ao tamanho do ciclo sazonal;
- m é o período à frente a ser previsto;
- c é constante ou ponto de intercepto;
- φ_t é o coeficiente do fator autorregressivo;
- θq é o coeficiente do fator de média móvel;
- Y'_t são os componentes das diferenças tomadas da série;
- ε_t são os erros aleatórios;
- a é o valor do intercepto do eixo Y ;
- b é o coeficiente angular que minimiza os quadrados dos erros.

De forma sucinta, a escolha dos modelos pode levar em conta algumas características específicas do comportamento dos conjuntos de dados analisados, como mostrado na Tabela 3.

Tabela 3 - Escolha dos modelos.

Série Temporal	Nível	Tendência	Sazonalidade	Estacionariedade	Não Estacionariedade
Média Móvel Simples (MMS)	✓			✓	
Média Móvel Ponderada (MMD)	✓	✓		✓	
Suavização Exponencial Simples (SES)	✓			✓	
Suavização Exponencial Dupla (Holt)	✓	✓		✓	
Suavização Exponencial Tripla (Holt-Winters)	✓	✓	✓	✓	
Auto Regressivos e de Médias Móveis (ARMA)	✓	✓	✓	✓	
Auto Regressivos Integrados e de Médias Móveis (ARIMA e SARIMA)	✓	✓	✓	✓	✓

Fonte: Bortoletto et al. 2016.

Métricas complementares são utilizadas para refinar o uso dos modelos, avaliando critérios e parâmetros para os modelos, bem como os erros de previsão, ou seja, os desvios entre a demanda real e previsão do modelo. A tabela 4 mostra um panorama geral de algumas destas métricas.

Tabela 4 - Métricas para avaliação dos modelos.

Métrica	Descrição	Como Avaliar
Log-Likelihood	Mede a probabilidade do modelo dado os dados observados.	Quanto maior, melhor. Modelos com valores mais altos indicam um melhor ajuste aos dados.
AIC (Akaike Information Criterion)	Critério de informação que penaliza pela complexidade do modelo.	Quanto menor, melhor. Indica um bom ajuste com menor complexidade do modelo.
BIC (Bayesian Information Criterion)	Similar ao AIC, mas penaliza mais severamente pela complexidade do modelo.	Quanto menor, melhor. Indica um bom ajuste com maior simplicidade do modelo, preferível quando comparado ao AIC em grandes amostras.
MSE (Mean Squared Error)	Média dos quadrados dos erros de previsão.	Quanto menor, melhor. Indica menor erro de previsão.

Métrica	Descrição	Como Avaliar
Resíduos Normalidade	Avalia se os resíduos do modelo seguem uma distribuição normal.	Testes como Shapiro-Wilk ou Q-Q plot são usados. Idealmente, os resíduos devem seguir uma distribuição normal.
Autocorrelação Residual	Verifica a presença de correlação nos resíduos do modelo.	Utiliza-se o teste de Ljung-Box. A ausência de autocorrelação é preferível (p-valor > 0.05).
Heterocedasticidade	Avalia se a variância dos resíduos é constante ao longo das previsões.	Teste de Breusch-Pagan ou White são usados. A ausência de heterocedasticidade é preferível (p-valor > 0.05).

Fonte: Bortoletto et al. 2016.

Analogamente aos modelos, cada métrica de avaliação pode ser interpretada da seguinte forma:

Log-Likelihood

A função de verossimilhança (*Log-Likelihood*) é uma medida que expressa a probabilidade de os dados observados serem gerados pelo modelo estimado. É calculada através do logaritmo natural da função de verossimilhança, dada por:

$$\text{Log-Likelihood} = \sum_{i=1}^n \ln(f(y_i|\theta))$$

Onde:

- n é o número de observações;
- y_i são os dados observados;
- θ são os parâmetros do modelo;
- $f(y_i | \theta)$ é a função densidade de probabilidade dos dados y_i condicionada aos parâmetros θ .

AIC (Akaike Information Criterion)

O Critério de Informação de Akaike (*AIC*) é utilizado para comparar modelos estatísticos, penalizando modelos que possuam um número excessivo de parâmetros. A fórmula é dada por:

$$AIC = 2k - 2 \ln(\hat{L})$$

Onde:

- k é o número de parâmetros estimados no modelo;
- $\ln(\hat{L})$ é o valor da Log-Likelihood do modelo.

BIC (Bayesian Information Criterion)

O Critério de Informação Bayesiano (*BIC*) é semelhante ao *AIC*, mas aplica uma penalização mais forte para modelos com mais parâmetros. Sua fórmula é dada por:

$$BIC = k \ln(n) - 2 \ln(\hat{L})$$

Onde:

- n é o número de observações;
- k é o número de parâmetros estimados no modelo;
- $\ln(\hat{L})$ é o valor da Log-Likelihood do modelo.

MSE (Mean Squared Error)

O Erro Quadrático Médio (*MSE*) é uma medida de precisão das previsões do modelo, calculada como a média dos quadrados dos resíduos:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Onde:

- n é o número de observações;
- y_i são os valores observados;
- $\hat{y}_i$ são os valores preditos no modelo.

Normalidade de Resíduos

A normalidade dos resíduos é um pressuposto fundamental para a validade de muitos testes estatísticos e é avaliada através de testes como o teste de Shapiro-Wilk ou o teste de Jarque-Bera. A estatística do teste de Jarque-Bera, por exemplo, é dada por:

$$JB = \frac{n}{6} \left(S^2 + \frac{(K-3)^2}{4} \right)$$

Onde:

- n é o número de observações;
- S é a assimetria dos resíduos;
- K é a curtose dos resíduos.

Autocorrelação Residual

A autocorrelação dos resíduos é uma indicação de que há padrões não capturados pelo modelo, e pode ser avaliada pelo teste de Durbin-Watsonb (DW) e Ljung-Box (Q):

$$DW = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}$$

Onde:

- e_i são os resíduos.

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n-k}$$

Onde:

- n é o número de observações;
- $\hat{\rho}_k$ é o coeficiente de autocorrelação no *lag* k .
- h é o número de *lags* testados.

Heterocedasticidade

Heterocedasticidade ocorre quando a variância dos resíduos não é constante ao longo das observações. O teste de Breusch-Pagan é uma das formas de detectar heterocedasticidade:

$$BP = \frac{n}{\downarrow} \left(\frac{R^2}{1 - R^2} \right)$$

Onde:

- R^2 é o coeficiente de determinação da regressão auxiliar sobre os resíduos ao quadrado.

METODOLOGIA

Todos os algoritmos foram desenvolvidos e executados em linguagem Python, através da plataforma *Google Colaboratory*, em sistema operacional *Windows 11 Home*, em computador com processador *12th Gen Intel® Core™ i5-1235U* 1.30 GHz RAM 8GB, 64-bit.

Conjunto de dados

Com o objetivo de ajustar modelos aptos a projetarem a demanda de financiamento imobiliário no Brasil, os dados abaixo foram coletados a partir de bases abertas e fechadas:

- Crescimento populacional: dados da população por município, disponibilizados pelo IBGE, ao ano, no período de 2017 a 2021;
- População acima 18 Anos: dados da população acima de 18 anos por município, disponibilizados pelo IPEA, por intervalo de dez anos, no período de 2000 a 2022;
- Acesso a esgoto: dados do percentual de residências com acesso a esgoto por município, disponibilizados pelo IPEA, por intervalo de dez anos, no período de 2000 a 2022;

- PIB per capita: dados do PIB per capita por município, disponibilizados pelo IBGE, ao ano, no período de 2010 a 2021;
- Dados de financiamento e crédito: dados de volumes da carteira de financiamentos imobiliários e empréstimos em geral por município e taxa SELIC nominal, disponibilizados pelo Banco Central do Brasil, ao ano, no período de 2014 a 2023;
- Renda per capita: dados de renda per capita por município, disponibilizados pelo IBGE, por intervalo de dez anos, no período de 1991 a 2010;
- Quantidade de domicílios: dados da quantidade de domicílios por capitais do Brasil, disponibilizados pelo IBGE, ao ano, no período de 2016 a 2022;
- Nível de desocupação: dados do nível de desocupação da população acima de 14 anos por capitais do Brasil, disponibilizados pelo IBGE, ao ano, no período de 2012 a 2023;
- Rendimento médio: dados da renda média da população acima de 14 anos por estados do Brasil, disponibilizados pelo IBGE, ao ano, no período de 2012 a 2023;
- Lançamentos imóveis: dados fechados da quantidade de lançamentos imobiliários disponíveis em *bureau* de dados especializado, por município, ao ano, no período de 2019 a 2023. A base contém os dados complementares de valor de venda e quantidade de unidades;
- Financiamentos imobiliários: dados fechados da quantidade de financiamentos imobiliários em uma instituição financeira, por município, ao ano, no período de 2019 a 2023. A base contém cerca de 314.557 registros de contratações, sumarizados em 6.154 linhas. Contempla ainda dados de valor total financiado, valor médio financiado, valor solicitado para financiamento, valor avaliado do imóvel, taxa média de contrato e área do imóvel;

Preparação inicial dos dados

Em função da restrição temporal dos dados de financiamentos imobiliários disponíveis em anos completos a partir de 2019 até 2023, todos os demais dados precisaram ser equiparados na mesma base temporal. Em adicional, no Brasil os dados abertos têm uma grande defasagem temporal de atualização das bases

disponibilizadas. Em função disso, algumas técnicas foram realizadas para que os dados de todas as bases tivessem uma equivalência temporal, conforme demonstrado na Tabela 5.

Tabela 5 - Preparação dos dados.

Tabela	Ajustes
Crescimento Populacional	Aplicação da taxa de crescimento do ano de 2021 para 2022 e 2023
População Acima 18 Anos	Conversão dos dados decenais em anuais por interpolação e aplicação da taxa de crescimento populacional para 2023
Percentual Residências com Acesso à Esgoto	Conversão dos dados decenais em anuais por interpolação e aplicação da taxa de crescimento do ano de 2022 para 2023
PIB per capita	Aplicação da taxa de crescimento do ano de 2021 para 2022 e 2023
Dados Financiamento e Crédito	-
Renda per capita	Conversão dos dados decenais em anuais por interpolação e aplicação da taxa do PIB per capita até 2023
Quantidade de Domicílios	Aplicação da proporção de população por município na extrapolação da quantidade de domicílios e aplicação da taxa de crescimento populacional para 2023
Nível de Desocupação	Réplica do dado da capital nos demais municípios do respectivo estado
Rendimento Médio	Réplica do dado do estado nos demais municípios do respectivo estado
Financiamentos Imobiliários	-
Lançamentos Imobiliários	-

Fonte: elaborado pelo autor.

Além disso, foi realizada uma marcação de região geográfica para dividir os dados, conforme atendimento comercial vigente na instituição financeira provedora dos dados de financiamentos imobiliários, mostrado na Tabela 6.

Tabela 6 - Agrupamento regional comercial.

Região Geográfica	Agrupamento
Norte	Região 1
Nordeste	Região 2
Sudeste	Região 3
São Paulo	Região 6
Sul	Região 7
Centro-Oeste	Região 8

Fonte: elaborado pelo autor.

Processamento dos dados

Após a preparação inicial dos dados para garantir que os espaços temporais sejam os mesmos, já dentro do ambiente *Google Colaboratory* todas as tabelas foram importadas para geração de um *dataframe* único para início das análises, a partir da combinação de Ano, Região, UF e Município. Além disso, foram filtrados os dados a partir de 2019 e tratados a fim de garantir que todas as combinações tivessem 5 linhas, representando os anos de 2019 a 2023, com o preenchimento de eventuais valores ausentes com a média do período. Foi utilizado como premissa os comportamentos regionais para o produto Financiamento Imobiliário, logo os respectivos agrupamentos refletirão a quantidade de análises e de modelos.

Posteriormente, para que cada modelo fique enxuto, porém refinado foram realizadas as análises de correlações das *features* para exclusão daquelas que apresentam o comportamento semelhante dentro da modelagem. A Tabela 7, demonstra as features relevantes em cada análise de modelo por região.

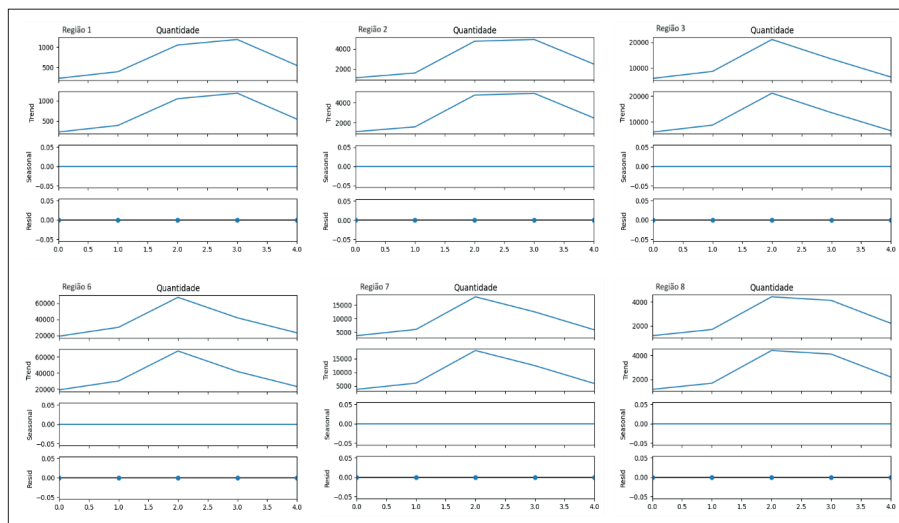
Tabela 7 - Features utilizadas.

Agrupamento	Features Utilizadas
Região 1	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Selic
Região 2	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Valor Avaliado, Taxa, Área, Empreendimentos, Unidades, Selic
Região 3	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Valor Avaliado, Área, Selic
Região 6	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Área, Selic
Região 7	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Valor Avaliado, Área, Selic
Região 8	População, %Esgoto, PIB Capita, Renda Capita, Desocupação, Rendimento, Valor Médio, Área, Empreendimentos, Selic

Fonte: elaborado pelo autor.

Com a seleção das respectivas features, foram realizadas as análises gráficas, conforme Figura 2, com objetivo da observação dos comportamentos das séries e a fim de identificar possíveis cenários de tendência, sazonalidade e estacionariedade.

Figura 2 - Análises gráficas das séries para cada região.



Fonte: elaborado pelo autor.

Ainda com relação a estacionariedade, também foi verificado através do teste ADF, apresentado na Tabela 8.

Tabela 8 - Resultado dos testes ADF.

Agrupamento	Resultado Valor-p Teste ADF
Região 1	0,554379
Região 2	0,568261
Região 3	0,528939
Região 6	0,502860
Região 7	0,513549
Região 8	0,545748

Fonte: elaborado pelo autor.

Avaliando as seis séries temporais observa-se um provável comportamento cíclico, característico do mercado imobiliário, porém em função da restrição do histórico de dados não é possível afirmar. Existe uma tendência de crescimento experimentado entre 2020 e 2022, com uma desaceleração no ano de 2023, retornando a patamares próximos a 2024. Utilizando o conhecimento prévio deste mercado, ainda que a restrição da quantidade de dados na série temporal não revele, os modelos sazonais serão testados para observação de alguma melhora nos ajustes. Com relação a estacionariedade, todos os testes ADF, revelam valores-p que aceitam a hipótese H_0 , não demonstrando estacionariedade. Contudo da mesma forma, não é possível afirmar que este comportamento seja padrão em series maiores.

Diante das análises, como decisão para avanço do estudo, serão utilizados os modelos autorregressivos integrados de médias móveis que tendem a trazer resultados mais bem ajustados a séries não estacionárias.

Métodos

Para testar qual modelagem melhor se ajusta aos dados de cada agrupamento que busca identificar a quantidade de financiamentos em cada município – as variáveis dependentes, foram testados para todos os modelos: Regressão Linear Múltipla, ARIMA, ARIMAX, SARIMA e SARIMAX, sendo os dois últimos avaliando a eventual presença sazonal. Os ajustes dos modelos autorregressivos foram feitos via processo de *Grid Search* para busca da otimização dos hiperparâmetros, eventual sazonalidade através da minimização dos erros quadráticos médios – *MSE* e ausência de autocorrelação de resíduos via Ljung-Box. Como forma de complementar as análises, foi realizada ainda a avaliação das variáveis exógenas

para os modelos ARIMAX e SARIMAX com objetivo de testar se eventuais modelos refinados com menos variáveis independentes possa apresentar erros menores.

Os dados foram divididos em conjuntos de treinamento e teste e através da biblioteca "*statsmodels*", os modelos foram ajustados e testados para gerar previsões. Para avaliação do modelo foi utilizada a biblioteca "*sklearn.metrics*", com o cálculo do *MSE* e Ljung-Box. Como apoio gráficos das previsões foram realizados usando a biblioteca "*matplotlib.pyplot*".

Após cada análise, os modelos treinados com melhores ajustes podem ser usados para gerar novas previsões para os próximos anos.

RESULTADOS E DISCUSSÕES

Os testes e resultados de cada modelo foram avaliados através dos resumos fornecidos via utilização da biblioteca "*statsmodels*". Como observação, nenhum modelo mostrou normalidade nos resíduos, porém os modelos sem as variáveis exógenas demonstraram melhor comportamento quanto a autocorrelação de resíduos e tiveram transformações logarítmicas realizadas como aprimoração através da biblioteca "*numpy*".

Para Região 1, os modelos ARIMA (1, 1, 1) e SARIMA (0,1,1)x(0,1,[1],2), apresentaram os melhores resultados com *MSE* entre 3,3 e 6,2 após transformação logarítmica, como mostra a Tabela 9.

Tabela 9 - Resumo das métricas para Região 1.

Modelo		Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	-	770,4	1.559,0	1.587,0	204,9	Não (valor-p=0,00)	Não (valor-p=0,2838)	-
ARIMA(1, 1, 1)	-	269,3	544,6	553,8	6,2	Não (valor-p=0,00)	Não (valor-p=0,9133)	Não (valor-p=0,4775)
ARIMAX(2, 1, 2)	-	766,8	1.559,5	1.599,5	98,5	Não (valor-p=0,00)	Não (valor-p=0,6649)	Sim (valor-p=0,00)
ARIMAX(2, 1, 2) Refinado	-	766,0	1.548,0	1.572,5	106,5	Não (valor-p=0,00)	Não (valor-p=0,7190)	Sim (valor-p=0,00)
SARIMA(0, 1, 1)x(0, 1, [1], 2)	-	273,2	552,5	561,7	3,3	Não (valor-p=0,00)	Não (valor-p=0,8894)	Não (valor-p=0,2657)
SARIMAX(2, 1, 2)x(0, 1, 0, 2)	-	784,4	1.594,9	1.634,7	81,3	Não (valor-p=0,00)	Não (valor-p=0,0682)	Sim (valor-p=0,00)
SARIMAX(2, 1, 2)x(0, 1, 0, 2) Refinado	-	784,1	1.590,2	1.623,9	91,7	Não (valor-p=0,00)	Não (valor-p=0,1064)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

Na análise da Região 2, os modelos ARIMA (2, 1, 2) e SARIMA (2,1,1)x(1,1,[1],4), não apresentaram autocorrelação residual e heterocedasticidade, conforme Tabela 10. Porém, a transformação logarítmica é efetiva apenas para o ARIMA.

Tabela 10 - Resumo das métricas para Região 2.

Modelo		Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	-	3.154,6	6.337,0	6.398,0	1.049,9	Não (valor-p=0,00)	Sim (valor-p=0,0160)	-
ARIMA(2, 1, 2)	-	960,0	1.929,9	1.951,8	3,1	Não (valor-p=0,00)	Não (valor-p=0,6387)	Não (valor-p=0,0662)
ARIMAX(0, 1, 2)	-	3.156,0	6.344,0	6.414,0	1.020,9	Não (valor-p=0,00)	Não (valor-p=0,1123)	Sim (valor-p=0,00)
ARIMAX(0, 1, 2) Refinado	-	3.155,8	6.323,5	6.349,7	1.088,5	Não (valor-p=0,00)	Não (valor-p=0,1216)	Sim (valor-p=0,00)
SARIMA(2, 1, 1)x(1, 1, 1, 4)	-	3.513,6	7.039,2	7.065,4	3.514,3	Não (valor-p=0,00)	Não (valor-p=0,9305)	Não (valor-p=0,7319)
SARIMAX(1, 1, 0)x(1, 1, 1, 4)	-	3.327,8	6.689,6	6.763,8	274,6	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
SARIMAX(1, 1, 0)x(1, 1, 1, 4) Refinado	-	3.307,3	6.630,6	6.665,5	306,3	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

Na Região 3, os modelos ARIMA (1, 1, 2) e SARIMA (0,2,2)x(1,0,[0],3), com MSE entre 2,7 e 6,7, tiveram os melhores resultados levando em consideração os testes, como indicado na Tabela 11.

Tabela 11 - Resumo das métricas para Região 3.

Modelo		Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	-	8.013,0	16.053,0	16.105,0	16.891,3	Não (valor-p=0,00)	Sim (valor-p=0,00)	-
ARIMA(1, 1, 2)	-	2.146,4	4.300,8	4.321,4	6,7	Não (valor-p=0,00)	Não (valor-p=0,7133)	Não (valor-p=0,6610)
ARIMAX(1, 1, 0)	-	8.269,5	16.563,0	16.624,7	16.887,3	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
ARIMAX(1, 1, 0) Refinado	-	8.271,7	16.553,4	16.579,1	19.058,9	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
SARIMA(0, 2, 2)x(1, 0, 0, 3)	-	2.153,8	4.315,6	4.336,2	2,7	Não (valor-p=0,00)	Não (valor-p=0,4632)	Não (valor-p=0,3407)
SARIMAX(1, 2, 0)x(0, 1, 1, 3)	-	8.814,2	17.654,4	17.721,1	2.184,8	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
SARIMAX(1, 2, 0)x(0, 1, 1, 3) Refinado	-	8.809,9	17.635,8	17.676,9	2.694,0	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

A Tabela 12 retorna as métricas dos modelos testados na Região 6. Os modelos ARIMA (1, 2, 2) e SARIMA (2, 1, 1) x(0, 1, [1], 2), não apresentam heterocedasticidade e autocorrelação residual pós transformações logarítmicas.

Tabela 12 - Resumo das métricas para Região 6.

Modelo	Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	- 9.143,1	18.310,0	18.360,0	215.200,6	Não (valor-p=0,00)	Não (valor-p=0,9983)	-
ARIMA(1, 2, 2)	- 2.413,4	4.834,7	4.855,1	35,5	Não (valor-p=0,00)	Não (valor-p=0,8640)	Não (valor-p=0,3837)
ARIMAX(2, 1, 2)	- 9.192,6	18.413,2	18.484,6	231.362,0	Não (valor-p=0,00)	Não (valor-p=0,0574)	Sim (valor-p=0,00)
SARIMA(2, 1, 1)x(0, 1, [1], 2)	- 2.413,7	4.837,4	4.862,8	57,3	Não (valor-p=0,00)	Não (valor-p=0,8949)	Não (valor-p=0,8485)
SARIMAX(1, 1, 2)x(0, 1, 1, 5)	- 9.337,2	18.702,4	18.773,7	225.729,8	Não (valor-p=0,00)	Não (valor-p=0,5728)	Sim (valor-p=0,00)
SARIMAX(1, 1, 2)x(0, 1, 1, 5) Refinado	- 9.333,7	18.685,5	18.731,3	227.499,3	Não (valor-p=0,00)	Não (valor-p=1,0000)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

Os modelo ARIMA (1, 2, 2) e SARIMA (0, 2, 2) x(0, 1, [1], 3), apresentam ausência de autocorrelação residual e heterocedasticidade. A Tabela 13 mostra que na Região 7, os critérios AIC e BIC são similares entre esses dois modelos.

Tabela 13 - Resumo das métricas para Região 7.

Modelo	Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	- 7.485,3	14.990,0	15.050,0	6.206,2	Não (valor-p=0,00)	Não (valor-p=0,9927)	-
ARIMA(1, 2, 2)	- 2.155,1	4.318,2	4.338,7	6,9	Não (valor-p=0,00)	Não (valor-p=0,1301)	Não (valor-p=0,3147)
ARIMAX(1, 1, 2)	- 7.515,7	15.059,4	15.131,2	6.800,4	Não (valor-p=0,00)	Não (valor-p=0,1107)	Sim (valor-p=0,00)
ARIMAX(1, 1, 2) Refinado	- 7.513,1	15.042,1	15.083,1	6.874,0	Não (valor-p=0,00)	Não (valor-p=0,1462)	Sim (valor-p=0,00)
SARIMA(0, 2, 2) x(0, 1, 1, 3)	- 2.177,1	4.362,2	4.382,6	3,3	Não (valor-p=0,00)	Não (valor-p=0,1287)	Não (valor-p=0,9959)
SARIMAX(2, 2, 0) x(0, 0, 0, 3)	- 8.053,9	16.133,8	16.200,4	1.784,2	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
SARIMAX(2, 2, 0)x(0, 0, 0, 3) Refinado	- 8.039,8	16.095,6	16.136,6	2.014,5	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

Por fim, na Região 8, os modelos ARIMA (1, 1, 2) e SARIMA (2,1,2)x(0,1,[1],3), não apresentaram autocorrelação residual e heterocedasticidade. Porém, a transformação logarítmica tornou-se efetiva apenas para o ARIMA, como mostra a Tabela 14.

Tabela 14 - Resumo das métricas para Região 8.

Modelo	Log-Likelihood	AIC	BIC	MSE	Resíduos Normalidade	Autocorrelação Residual	Heterocedasticidade
OLS	- 2.349,5	4.721,0	4.766,0	2.634,5	Não (valor-p=0,00)	Não (valor-p=0,0576)	-
ARIMA(1, 1, 2)	- 726,1	1.460,2	1.476,5	6,6	Não (valor-p=0,00)	Não (valor-p=0,5743)	Não (valor-p=0,1105)
ARIMAX(0, 1, 2)	- 2.359,6	4.745,2	4.798,1	2.770,3	Não (valor-p=0,00)	Não (valor-p=0,0717)	Sim (valor-p=0,00)
ARIMAX(0, 1, 2) Refinado	- 2.356,6	4.727,2	4.755,8	2.781,4	Não (valor-p=0,00)	Não (valor-p=0,0883)	Sim (valor-p=0,00)
SARIMA(2, 1, 2)x(0, 1, 1, 3)	- 2.734,0	5.480,0	5.504,4	5.523,4	Não (valor-p=0,00)	Não (valor-p=0,0610)	Não (valor-p=0,6749)
SARIMAX(0, 2, 2)x(1, 1, 1, 3)	- 2.597,8	5.225,6	5.286,5	1.150,1	Não (valor-p=0,00)	Sim (valor-p=0,00)	Sim (valor-p=0,00)
SARIMAX(0, 2, 2)x(1, 1, 1, 3) Refinado	- 2.370,1	4.758,2	4.794,8	2.866,7	Não (valor-p=0,00)	Não (valor-p=0,5390)	Sim (valor-p=0,00)

Fonte: elaborado pelo autor.

CONCLUSÕES

As análises dos testes de normalidade de resíduos mostram que nenhum dos modelos apresentados, nas seis regiões agrupadas, alcança resultados satisfatórios. Isso é um forte indicativo que a disponibilidade temporal dos dados apresentados é insuficiente para um desenvolvimento robusto em modelos requintados. A partir da aplicação de transformações logarítmicas, foram encontrados modelos sem autocorrelação residual e heterocedasticidade, o que, para a limitação de dados apresentada, possibilita almejar modelos para identificação de interesse na contratação de financiamento imobiliário.

Em função da complexidade de coleta de dados granulares atualizados no Brasil, além da restrição dos próprios dados coletados, os modelos que apresentaram melhores resultados têm na sua composição apenas a variável endógena de quantidade de financiamentos imobiliários em cada região. A análise gráfica dos dados históricos não permite afirmar a presença de uma sazonalidade declarada, ainda que o conhecimento da indústria bancária afirma que o comportamento do financiamento imobiliário é cíclico. Em função da análise disponível, os modelos SARIMA são preteridos aos ARIMA para geração de previsões de financiamentos em cenários futuros por município, de acordo com cada região geográfica agrupada.

Como sugestões de trabalhos futuros, recomenda-se a coleta de dados complementares com históricos maiores, até de outras instituições financeiras, para possibilitar a avaliação de modelos com variáveis exógenas e buscar mais precisão nas previsões. Novas variáveis relacionadas a demografia e desenvolvimento urbano

e social podem ser agregadas em uma nova modelagem. Em adição, modelos de aprendizado supervisionado como *Random Forests* podem ser testados, buscando a captura eficaz de relações entre várias variáveis econômicas, demográficas e de mercado via árvores de decisão. Além disso, a Regressão Logística pode ser aplicada para classificar diferentes regiões e momentos como mais ou menos propensos a terem alta demanda por crédito imobiliário.

REFERÊNCIAS

ABECIP Associação Brasileira das Entidade de Crédito Imobiliário e Poupança. **Mercado de capitais ganha espaço no funding imobiliário**. São Paulo, SP: ABECIP, 2023. Disponível em: <https://www.abecip.org.br/imprensa/noticias/mercado-de-capitais-ganha-espaco-no-funding-imobiliario-valor-economico>. Acesso em 25 de junho de 2023.

Andreoli, A.C.; Mariano, E.B, Maruyama, L.Y. **Aplicação de métodos para previsão de demanda no setor de crédito imobiliário**. In: XXI Simpósio de Engenharia de Produção, Bauru, SP, 2014.

Bortoletto, W.W.; Petrelli, M.Z.; Ignacio, P.S.A.; Silva, A.L. Modelos de séries temporais para previsão de demanda: estudo de caso em uma indústria eletroeletrônica. In: XXXVI Encontro Nacional de Engenharia de Produção, João Pessoa, PB, 2016.

João, R.J. **Relação econométrica da oferta e demanda no mercado imobiliário brasileiro: uma abordagem macroeconômica no comportamento dos preços, entre 2008 e 2015**. Trabalho de Conclusão de Curso (Graduação) – Ciências Econômicas, Universidade Federal de Ouro Preto, Mariana, MG, 2017.

Marques, F.E.F.A. **O Problema do Funding no Mercado de Crédito para Habitação no Brasil**. Dissertação (Mestrado em Economia Política) – Pontifícia Universidade Católica de São Paulo, São Paulo, SP, 2014.

Ministério do Desenvolvimento Regional; Fundação João Pinheiro. **Déficit Habitacional no Brasil – 2016-2019**. Belo Horizonte, MG: MDR; FJP, 2021.

Moro, M. **Modelo híbrido de séries temporais para previsão de demanda no mercado imobiliário de São Paulo**. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Maria, Santa Maria, RS, 2017.

Ribeiro, C.G.A.; Helena, F.F.; Cipparone, F.A.M. Construção de banco de dados do mercado imobiliário. Um estudo na cidade de São Paulo. In: Simpósio Brasileiro de Banco de Dados - SBBBD, 38, Belo Horizonte, MG, 2023.

SOBRE OS ORGANIZADORES

Elton Felipe Sbruzzi

PhD em Finanças Computacionais pela University of Essex (UK), Mestre e Graduado em Economia pela UFRGS e Unicamp, respectivamente; e Administrador de Carteira e Valores Mobiliários registrado na CVM/Anbima na modalidade Notório Saber. Atualmente, Professor Adjunto com Dedicção Exclusiva na Divisão de Ciência da Computação do Instituto Tecnológico de Aeronáutica. Aprovado em concurso público de provas e títulos na área de Inteligência Artificial e Ciência de Dados. Leciona as seguintes disciplinas: Matemática Computacional, Econometria, Projeto de Aprendizado de Máquina em Finanças e Modelagem de Investimentos e Riscos. Pesquisa aplicação de Ciência de Dados e Inteligência Artificial em Finanças e Investimentos.

Lattes: <http://lattes.cnpq.br/0026358605322965>

Filipe Alves Neto Verri

Formou-se em Ciências de Computação (2014) pela Universidade de São Paulo. Concluiu seu doutorado em Ciências de Computação e Matemática Computacional (2018) na mesma instituição. Em 2016, foi Pesquisador Visitante na School of Electrical, Computer and Energy Engineering da Arizona State University, sob a supervisão do Prof. Ying-Cheng Lai. Atualmente, é Professor Adjunto na Divisão de Ciência da Computação, Departamento de Métodos Computacionais, do Instituto Tecnológico de Aeronáutica (ITA). Seus interesses de pesquisa incluem ciência de dados, aprendizado de máquina, redes complexas e sistemas complexos.

Lattes: <http://lattes.cnpq.br/0145582312635382>

Vitor Venceslau Curtis

Possui doutorado (bolsista CAPES) em Ciência da Computação pelo Instituto Tecnológico de Aeronáutica (ITA), com estágio sanduíche (bolsista PDSE-CAPES) na Viterbi School of Engineering, University of Southern California (USC) em 2017, mestrado (bolsista CNPq) em Ciência da Computação pelo ITA e graduação em Informática pela Faculdade de Tecnologia (FATEC). Foi chefe do Departamento de Sistemas Computacionais (IEC-SC) da Divisão de Ciência da Computação (IEC) em 2023 a 2024. Atualmente é Professor Adjunto do ITA e membro fundador dos grupos de pesquisa: Grupo de Interesse em Finanças Computacionais e Investimentos Sistemáticos (FICO-ITA) e DroneComp Research Group (Dronecomp). Atua na área de otimização, aprendizado de máquina, computação de alto desempenho e finanças computacionais.

Lattes: <http://lattes.cnpq.br/1785341067396776>

ÍNDICE REMISSIVO

A

Aircraft Fuselage: 373, 374, 376, 377, 379, 385, 406, 409

Análise de Sentimento: 102, 104, 105, 106, 107, 108, 109, 110, 116, 117, 358

Análise Preditiva: 52, 55, 66, 184, 185

Anomalous Vehicle Interactions: 68, 69

Aprendizagem de Máquina: 180, 182, 184, 185, 186, 187, 194

Artificial Intelligence: 28, 46, 49, 196, 203, 237, 238, 277, 291, 292, 293, 294, 299, 372, 375, 407, 426, 444

Attention-Based Regression: 274, 275, 278, 279, 286, 287

AutoML: 182, 187, 188, 196

Avaliação de Imóveis: 141, 145, 150, 165

C

Chikungunya: 180, 181, 182, 183, 184, 185, 186, 194, 195, 196

Commodities Agrícolas: 167, 169, 170

Computer Vision: 68, 69, 70, 72, 83, 89, 96, 100, 374, 375, 380, 385, 388, 404, 407, 409, 410

Consultas no Setor Financeiro: 204, 206

Convolutional Neural Networks: 47, 72, 373, 374, 376, 392, 408, 410

Cuidado de Idosos: 197

D

Dados do Twitter: 102, 104, 105

Decision Support Systems: 412, 413, 415, 416

Defect Detection: 388, 394, 407, 408, 409

Demand Prediction: 291, 299

Desempenho: 14, 18, 57, 58, 61, 62, 63, 64, 65, 106, 107, 124, 128, 129, 131, 132, 133, 134, 135, 139, 154, 155, 157, 162, 163, 164, 178, 186, 189, 191, 192, 194, 206, 211, 212, 215, 217, 237, 242, 300, 302, 303, 304, 307, 308, 309, 311, 320, 322, 326, 327, 329, 330, 336, 339, 340, 341, 343, 344, 345, 346, 347, 348, 349, 352, 354, 356, 358, 359, 365, 369, 370, 371

Dólar Futuro: 119, 122

E

Editais de Credenciamento: 48, 50, 52, 57, 64, 66

Energia Solar: 300, 302, 304, 323

F

Fatores Macroeconômicos: 167, 168, 169, 170

Financiamento Imobiliário: 467, 469, 471, 472, 480, 483, 489

G

Gestão Pública: 328, 349

Grande Escala: 354, 356, 357, 364, 370

GraphRAG: 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 236, 237

H

Healthcare: 49, 67, 195, 196, 296, 412, 414, 415, 416, 426

Highways: 68, 70, 91, 100

Holding Maneuvers: 428, 429, 430, 431, 432, 433, 435, 437, 438, 439, 440, 441, 442, 443

I

Identificação de Oportunidades: 467, 469

Image Processing: 375, 433, 435, 443

Indicadores Financeiros: 108, 167, 170, 171

Instance Segmentation: 373, 374, 376, 378, 384, 388, 391, 406, 407, 409

Instituições Financeiras: 122, 145, 237, 242, 467, 469, 470, 471, 472, 489

Inteligência Artificial: 50, 53, 67, 102, 106, 144, 145, 169, 170, 185, 186, 202, 207, 324, 328, 349, 357, 411

Inteligência Artificial: 50, 53, 67, 102, 106, 144, 145, 169, 170, 185, 186, 202, 207, 324, 328, 349, 357, 411

Internação Hospitalar: 48, 50

Interpretability: 243, 261, 265, 287, 412, 415, 416, 417, 423, 426, 459, 460

L

LLMs: 67, 204, 205, 206, 207, 209, 210, 211, 237, 355, 356, 357, 359, 360, 362, 363, 364, 365, 370, 371

Loan Default: 240, 243, 247, 257, 264

Lower Calorific Values: 445

M

Machine Learning: 12, 21, 22, 23, 25, 34, 35, 37, 38, 39, 40, 41, 42, 46, 47, 49, 67, 83, 100, 101, 118, 127, 137, 138, 139, 141, 142, 143, 163, 166, 176, 184,

185, 186, 187, 195, 196, 228, 229, 230, 231, 233, 234, 235, 238, 240, 241, 243, 244, 245, 246, 247, 254, 255, 256, 258, 263, 264, 265, 266, 277, 286, 289, 290, 292, 297, 299, 323, 372, 377, 384, 385, 387, 392, 411, 412, 413, 414, 415, 416, 417, 420, 421, 422, 426, 427, 430, 442, 443, 444, 445, 446, 448, 450, 451, 452, 453, 464, 465

Machine Learning: 12, 21, 22, 23, 25, 34, 35, 37, 38, 39, 40, 41, 42, 46, 47, 49, 67, 83, 100, 101, 118, 127, 137, 138, 139, 141, 142, 143, 163, 166, 176, 184, 185, 186, 187, 195, 196, 228, 229, 230, 231, 233, 234, 235, 238, 240, 241, 243, 244, 245, 246, 247, 254, 255, 256, 258, 263, 264, 265, 266, 277, 286, 289, 290, 292, 297, 299, 323, 372, 377, 384, 385, 387, 392, 411, 412, 413, 414, 415, 416, 417, 420, 421, 422, 426, 427, 430, 442, 443, 444, 445, 446, 448, 450, 451, 452, 453, 464, 465

Machine Learning Models: 21, 22, 23, 34, 35, 37, 38, 39, 40, 42, 240, 241, 244, 245, 246, 254, 256, 258, 263, 265, 277, 289, 299, 392, 412, 414, 416, 417, 421, 442, 443, 446, 451, 452, 453, 464

Maratonas de Programação: 354, 359

Meteorological Images: 428, 432

Missing Data: 250, 300, 301, 302

Modelo Preditivo: 123, 124, 127, 154, 155, 157, 163, 170, 177, 467, 472

Modelos de Linguagem: 206, 207, 209, 211, 354, 356, 357, 358, 364, 370

Models for Financial Trend Prediction: 21

O

OHLCV: 21, 22, 23, 24, 27, 29, 37, 38, 40

P

Paralisações de Obras Públicas: 324, 328

Parques Eólicos: 12, 13, 14, 15

Perdas de Dados: 300, 308, 322

Predição de Cronicidade em Chikungunya: 182

Predição do Preço do Bitcoin: 102

Predictions: 24, 25, 30, 31, 35, 37, 247, 248, 249, 255, 256, 257, 259, 262, 263, 378, 416, 417, 421, 424, 426, 427, 430, 431, 433, 438, 439, 440, 442, 443, 445, 448, 452, 459, 464

Predictive Analysis: 240

Protótipo: 197, 199, 201

R

Random Forest: 22, 24, 25, 30, 31, 32, 37, 38, 39, 40, 41, 42, 47, 58, 61, 62, 66, 142, 143, 155, 181, 182, 184, 190, 191, 192, 193, 194, 241, 242, 243, 245, 246, 256, 258, 260, 261, 262, 263, 272, 273, 284,

285, 286, 294, 298, 299, 325, 326, 339, 340, 341, 342, 344, 347, 349, 413, 421, 423, 460, 461, 462, 463, 464, 465

Real Estate Valuation: 274, 275, 276, 277, 282, 287, 289

Real-Time Optimization: 445, 446, 448, 450, 452, 464, 465

Regressão Logística: 119, 121, 122, 123, 124, 125, 127, 128, 129, 132, 136, 137, 138, 242, 490

Remote Sensing: 70, 291, 292, 293, 294, 297, 299, 408

Risco de Atrasos: 324, 326, 328, 329, 345

S

Saúde Pública: 182, 183, 184, 186, 196

Sistema de Apoio: 197

Sistema de Negociação: 119

Sistemas Automatizados: 141

T

Traffic Cameras: 68, 69



científica digital



VENDA PROIBIDA - ACESSO LIVRE - OPEN ACCESS

