Wesley Pereira de Almeida

Aplicação de modelos de Aprendizado de Máquina na estimação da coluna troposférica de NO2 do TROPOMI no estado do Pará

São Paulo, SP

2023

Wesley Pereira de Almeida

Aplicação de modelos de Aprendizado de Máquina na estimação da coluna troposférica de NO2 do TROPOMI no estado do Pará

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Pedro Luiz Pizzigatti Corrêa

Coorientador: Profa. Dra. Luciana Varanda Rizzo

São Paulo, SP

2023

Gerar a ficha catalográfica em https://www.poli.usp.br/bibliotecas/servicos/catalogacao-na-publicacao
Salvar o pdf e incluir na monografia

O ser humano é aquilo que a educação faz dele.

Agradecimentos

Agradeço, primeiramente, à minha família pelo apoio e encorajamento ao longo de minha jornada acadêmica e na conclusão deste projeto.

Gostaria de expressar minha profunda gratidão ao meu dedicado orientador, professor Pedro Luiz Pizzigatti Corrêa, cuja orientação e apoio foram inestimáveis ao longo deste projeto. Também sou grato à professora Luciana Varanda Rizzo por suas valiosas contribuições e orientação.

Estendo minha gratidão aos pesquisadores Jeaneth Machicao e Tony Dias por suas contribuições a este estudo, que enriqueceram significativamente sua qualidade e escopo.

Resumo

No contexto atual, o uso de dados provenientes do sensoriamento remoto desempenha um papel de extrema relevância na compreensão e monitoramento das transformações ambientais em escala global. A motivação para esse estudo é impulsionada pela crescente necessidade de monitorar a qualidade do ar e seus efeitos nas regiões ambientalmente sensíveis, como a Amazônia. Diante desse cenário, este projeto tem como propósito a aplicação de técnicas de Aprendizado de Máquina com o intuito de estimar a concentração de NO2 na coluna troposférica do sensor TROPOMI, com foco voltado para a região do estado do Pará, no Brasil. O desenvolvimento deste trabalho envolve a seleção aleatória de 50 pontos na área do estado do Pará, a aquisição e pré-processamento de dados provenientes de sensoriamento remoto, a construção de um dataset composto por séries temporais de variáveis independentes ao longo de um período de um ano, e, por fim, a aplicação de modelos de regressão. Os resultados obtidos demonstram a viabilidade da abordagem, com métricas de avaliação, como o Coeficiente de Determinação (R^2) atingindo o melhor resultado para o modelo PCA + *XGBoost* que obteve um R^2 de 0.47. Este projeto oferece contribuições significativas para a compreensão e monitoramento ambiental, delineando possíveis direções para futuros refinamentos e aprimoramentos na abordagem.

Palavras-chave: Sensoriamento remoto, Aprendizado de Máquina, NO2, TROPOMI, Regressão, Séries Temporais e Amazônia

Abstract

In the current context, the use of data from remote sensing plays an extremely relevant role in understanding and monitoring environmental changes on a global scale. The motivation for this study is driven by the growing need to monitor air quality and its effects in environmentally sensitive regions, such as the Amazon. In this scenario, this project aims to apply Machine Learning techniques to estimate the concentration of NO2 in the tropospheric column from the TROPOMI sensor, focusing on the state of Pará, Brazil. The development of this work involves the random selection of 50 points within the state of Pará, the acquisition and pre-processing of remote sensing data, the construction of a dataset composed of time series of independent variables over a period of two years, and, finally, the application of regression models. The results obtained demonstrate the feasibility of the approach, with evaluation metrics, such as the Coefficient of Determination (R^2), reaching the best result for the PCA + *XGBoost* model, which achieved an R^2 of 0.47. This project provides significant contributions to environmental understanding and monitoring, outlining possible directions for future refinements and improvements in the approach.

Keywords: Remote Sensing, Machine Learning, NO2, TROPOMI, Regression, Time Series, and Amazon.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
CH4	Metano
CO	Monóxido de Carbono
CO2	Dióxido de Carbono
CRS	Sistema de referência espacial
ESA	Agência Espacial Europeia
GEE	*Google Earth Engine*
GRU	Unidades Recorrentes Gated
LSTM	*Long Short-Term Memory*
MAE	Erro Médio Absoluto
ML	*Machine Learning*
MSE	Erro Quadrático Médio
NO2	Dióxido de Nitrogênio
PCA	Análise dos componentes principais
R^2	Coeficiente de Determinação
RNN	Rede neural recorrente
SO2	Dióxido de enxofre
TROPOMI	TROPOspheric Monitoring Instrument

Lista de símbolos

β	Letra grega minúscula Beta
λ	Letra grega minúscula Lambda
σ	Letra grega minúscula Sigma
$\odot$	Multiplicação elemento a elemento
Σ	Somatório
$\cdot$	Produto interno

Sumário

1 Introdução

No cenário atual, o uso de dados de sensoriamento remoto desempenha um papel fundamental na compreensão e monitoramento das mudanças ambientais globais. Essa tecnologia permite uma observação detalhada e consistente da Terra a partir de satélites em órbita, fornecendo informações para abordar questões críticas relacionadas ao meio ambiente e a qualidade de vida da população. No estado do Pará, localizado na região norte do Brasil, onde a rica biodiversidade da Amazônia se entrelaça com a crescente atividade humana, a utilização de dados de sensoriamento remoto torna-se ainda mais crucial.

A vasta extensão territorial e a complexidade da Amazônia tornam desafiador o monitoramento e a compreensão das mudanças ambientais nessa região. Os dados de sensoriamento remoto, como os provenientes do satélite Sentinel-5P e seu instrumento TROPOspheric Monitoring Instrument (TROPOMI), desempenham um papel central na coleta de informações sobre a qualidade do ar, incluindo a concentração de dióxido de nitrogênio (NO2) na coluna troposférica. Essas informações são essenciais para avaliar os impactos das atividades humanas, como desmatamento e queimadas, na atmosfera da Amazônia, contribuindo assim para uma compreensão mais abrangente da interação entre a biodiversidade e o meio ambiente regional, bem como para os esforços de conservação. Assim, a obtenção de dados de qualidade torna-se uma necessidade incontestável neste contexto.

1.1 Motivação

A motivação para este estudo surge da crescente importância de monitorar a qualidade do ar e seus impactos nas regiões sensíveis, como a Amazônia, especialmente o estado do Pará. Este ecossistema desempenha um papel fundamental no equilíbrio ambiental global e na regulação do clima (ARAGÃO et al., 2018) (MALHI et al., 2008). No entanto, ações humanas, como o desmatamento, têm aumentado a presença de poluentes atmosféricos, incluindo o dióxido de nitrogênio (NO2), na atmosfera amazônica.

A importância de avaliar a concentração de NO2 na coluna troposférica da Amazônia ganha destaque no contexto do estado do Pará, uma região crítica em termos de atividade econômica e impactos ambientais. Dados recentes do satélite Sentinel-5P e seu instrumento TROPOMI proporcionam uma oportunidade única para estimar a concentração e tendência de NO2 (WANG; FALOONA; HOULTON, 2023). Nesse contexto, este estudo visa estimatimar a concentração de NO2 justamente para os pontos onde a presença de nuvens impediu a aquisição de dados diretamente a partir de satélites, como o Sentinel-

5P com seu instrumento TROPOMI. Assim, a aplicação de técnicas de Aprendizado de Máquina (ML) ganha destaque, permitindo a estimativa precisa da concentração de NO2 em locais afetados pela presença de nuvens, contribuindo significativamente para uma compreensão mais abrangente dos impactos ambientais e fornecendo informações valiosas para a conservação da Amazônia

1.2 Objetivos

1.2.1 Objetivos Gerais

O objetivo geral do projeto é contribuir significativamente para os estudos de mudanças climáticas e o balanço de carbono, especificamente no contexto da Amazônia. A região amazônica, sendo um ecossistema vital para a regulação do clima global e um importante reservatório de carbono, enfrenta desafios únicos devido ao desmatamento e às mudanças climáticas. O desenvolvimento de uma metodologia para concentração de NO2 na coluna troposférica proporciona dados essenciais para entender melhor o impacto das atividades humanas e os processos naturais na dinâmica climática e no ciclo do carbono na Amazônia.

1.2.2 Objetivos Específicos

O objetivo específico deste trabalho é desenvolver e avaliar um conjunto de modelos de estimativa da concentração de dióxido de nitrogênio (NO2) na coluna troposférica utilizando dados do sensor TROPOMI do satélite Sentinel-5P, empregando técnicas de Aprendizado de Máquina com o auxílio de diversas variáveis de sensoriamento remoto. Esse estudo visa superar o desafio de obtenção de dados precisos em regiões tropicais, como a Amazônia no estado do Pará, devido à presença frequente de nuvens, contribuindo assim para a monitorização da qualidade do ar e a compreensão dos impactos das atividades humanas nessa região crítica para o equilíbrio ambiental global.

1.3 Justificativa

Este projeto de formatura justifica-se devido a uma série de fatores. Primeiramente, a Amazônia desempenha um papel importante no sistema climático global, influenciando feedbacks climáticos cruciais, como os impactos do desmatamento e da intervenção humana no ciclo de carbono (ARAGÃO et al., 2018); (MALHI et al., 2008). Além disso, o NO2, como um poluente atmosférico crítico, contribui para a concentração de ozônio na troposfera e regula a capacidade oxidante (CRUTZEN, 1979), impactando diretamente a qualidade do ar e o clima global. Portanto, a estimativa precisa da concentração de NO2 na coluna troposférica da Amazônia, especialmente no estado do Pará, é fundamental para

o monitoramento ambiental e a compreensão dos impactos das atividades humanas nessa região.

Além disso, o estudo se torna ainda mais relevante devido à presença recorrente de nuvens nas florestas tropicais amazônicas, o que representa um desafio substancial para a aquisição de dados precisos por meio de sensoriamento remoto. A escassez de observações diretas em pontos afetados pela presença de nuvens limita a capacidade de monitorar e compreender as variações da concentração de NO2 na região. Assim, o uso de técnicas de Aprendizado de Máquina e o sensor TROPOMI do satélite Sentinel-5P apresentam uma abordagem inovadora para estimar a concentração de NO2, preenchendo as lacunas de dados causadas por nuvens. Isso é importante para melhorar o entendimento dos impactos ambientais e para embasar a tomada de decisões relacionadas à conservação da Amazônia.

Por fim, este projeto alinha-se com diversos estudos relevantes que empregam técnicas de Aprendizado de Máquina (ML) em dados de sensoriamento remoto para monitorar a qualidade do ar e estimar a concentração de poluentes atmosféricos. Por exemplo, pesquisas recentes aplicaram ML para estimar a concentração de CO2 em áreas oceânicas (ZENG; MATSUNAGA; SHIRAI, 2022) e estimar a concentração de NO2 a nível de superfície utilizando dados de sensoriamento remoto (GHAHREMANLOO et al., 2021). Além disso, a importância da Amazônia como um bioma crítico e as complexidades associadas à sua monitorização também foram abordadas em estudos que utilizam dados de satélite para analisar a mudança do uso da terra e a resposta climática na região (DAVIDSON et al., 2012); (NOBRE et al., 2016). Portanto, a relevância deste projeto está ancorada na necessidade de aprimorar a capacidade de monitorar a qualidade do ar e compreender os impactos ambientais na Amazônia, alinhando-se com pesquisas anteriores que têm abordado questões semelhantes.

1.4 Organização do Trabalho

A estrutura deste trabalho consiste em seis capítulos interligados que abordam o problema proposto de estimação da concentração de NO2.

No Capítulo 1, a introdução oferece uma visão geral do contexto do projeto, destacando sua relevância e motivação. Também são delineados os principais objetivos e a estrutura geral do trabalho.

No Capítulo 2, intitulado “Aspectos Conceituais”, exploram-se os conceitos fundamentais que servirão como base para a pesquisa. Isso inclui a compreensão de sensoriamento remoto, séries temporais e modelos de machine learning e deep learning.

No Capítulo 3, descreve-se o método de trabalho adotado para abordar o problema. Descrevemos nele a abordagem adotada para resolver o problema proposto de estimação da

concentração de NO2. A metodologia segue um ciclo de experimento de dados, a partir de um estudo de caso no estado do Pará no Brasil. Além disso, apresenta-se nesse capítulo os requisitos funcionais e não-funcionais. Aqui, estabelecem-se as funcionalidades específicas que o modelo deve oferecer, bem como os requisitos de desempenho, escalabilidade e precisão.

No Capítulo 4, o foco está na descrição detalhada do processo de coleta de dados, treinamento e teste dos modelos. São apresentados os resultados obtidos por meio da aplicação de modelos de machine learning e deep learning na estimação da concentração de NO2.

Finalmente, no Capítulo 5, realiza-se uma avaliação do trabalho realizado. São discutidos os resultados alcançados, metas não alcançadas e possíveis perspectivas futuras. Além disso, destaca-se a contribuição do trabalho para a comunidade científica.

2 Aspectos Conceituais

Nesta seção, abordaremos os elementos fundamentais para a estruturação e embasamento deste trabalho. Inicialmente, exploraremos aspectos relevantes do sensoriamento remoto, essenciais para a coleta de dados sobre a concentração de NO2 na coluna troposférica. Em seguida, discutiremos as técnicas de aprendizado de máquina, destacando sua importância na estimativa de NO2 em áreas com cobertura de nuvens, enfocando algoritmos, treinamento, validação e avaliação dos modelos.

2.1 Sensoriamento Remoto

O sensoriamento remoto é uma disciplina multidisciplinar que desempenha um papel fundamental na coleta de informações sobre a Terra e seus fenômenos por meio de sensores e plataformas espaciais (JENSEN, 2009). Esta área surgiu no início do século XX com o desenvolvimento da aviação e, posteriormente, da exploração espacial, permitindo a aquisição de dados sobre a superfície terrestre a partir de uma perspectiva remota (LILLESAND; KIEFER; CHIPMAN, 2015). Os sensores remotos, montados em aeronaves, satélites e outras plataformas, capturam informações eletromagnéticas em diversas faixas do espectro eletromagnético, incluindo luz visível, infravermelho e micro-ondas, para gerar imagens e dados valiosos sobre a Terra.

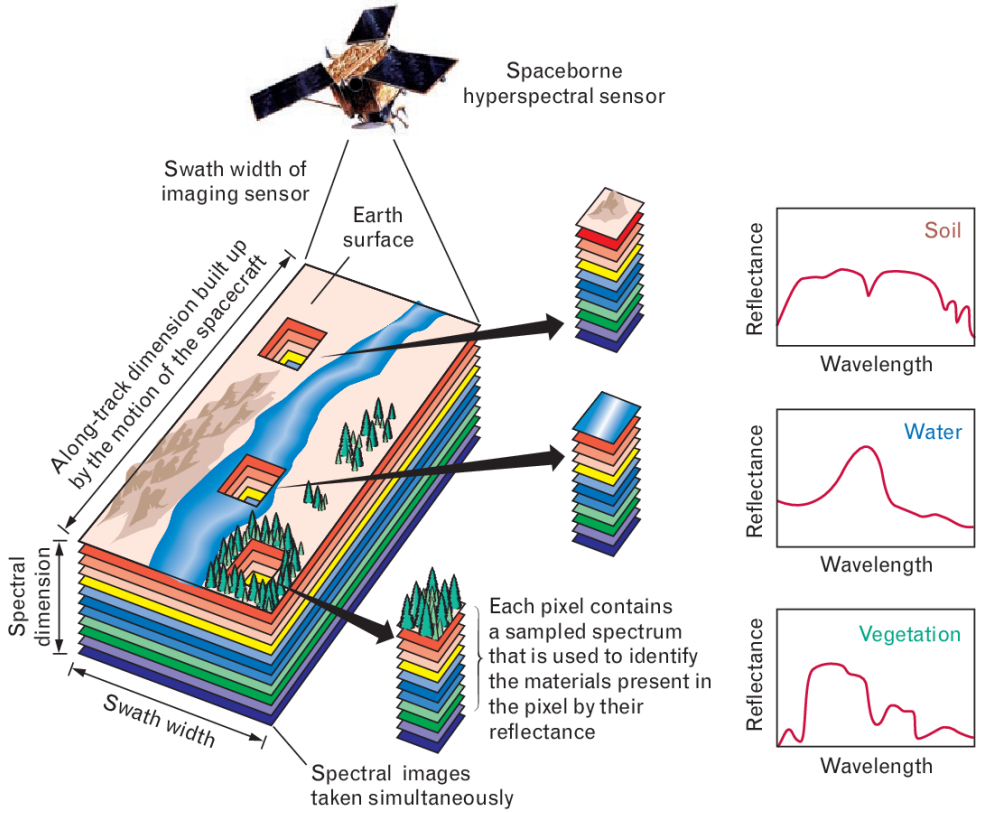
Atualmente, o sensoriamento remoto desempenha um papel indispensável em diversas áreas, incluindo monitoramento ambiental, gestão de recursos naturais, previsão de desastres, agricultura de precisão e estudos climáticos. Ele fornece informações cruciais para compreender os processos ambientais, detectar mudanças na superfície terrestre e tomar decisões informadas para o desenvolvimento sustentável e a preservação do meio ambiente. Portanto, o sensoriamento remoto continua a evoluir e a desempenhar um papel vital no avanço do conhecimento científico e na resolução de desafios globais.

2.1.1 Captura de Imagens por Sensores de Satélites

A captura de imagens por sensores de satélites é uma técnica essencial no campo do sensoriamento remoto. Esses sensores funcionam detectando a radiação eletromagnética refletida ou emitida pela superfície terrestre. As imagens são capturadas em diferentes bandas espectrais, cada uma sensível a diferentes comprimentos de onda. Por exemplo, a banda visível é usada para capturar imagens que se assemelham ao que vemos com nossos olhos, enquanto a banda infravermelha é útil para detectar informações sobre temperatura e vegetação (LILLESAND; KIEFER; CHIPMAN, 2015).

A Figura 1 apresenta o conceito de espectroscopia de imagem. Um sensor de imagem aéreo ou espacial amostra simultaneamente múltiplas faixas espectrais sobre uma grande área em uma cena terrestre. Após o processamento adequado, cada pixel na imagem resultante contém uma medição espectral amostrada de reflectância, que pode ser interpretada para identificar os materiais presentes na cena. Os gráficos na figura ilustram a variação espectral na reflectância para solo, água e vegetação. Uma representação visual da cena em diferentes comprimentos de onda pode ser construída a partir dessas informações espectrais.

Figura 1 – Ilustração do funcionamento de um sensor hiperespectral a bordo de um satélite, mostrando como ele capta imagens em diferentes comprimentos de onda simultaneamente da superfície terrestre.



Fonte: (SHAW; BURKE, 2003)

Os sensores registram a radiação capturada e a convertem em sinais elétricos, que são então transmitidos de volta à Terra e processados para criar imagens. A espectroscopia permite que os cientistas analisem a composição da superfície terrestre com base na maneira como a radiação é absorvida, refletida ou emitida pelos objetos. Cada banda espectral fornece informações únicas sobre a superfície da Terra, permitindo a análise detalhada de vários aspectos geoespaciais (CHANG, 2001). A combinação de várias bandas espectrais é fundamental para aplicações como monitoramento ambiental, agricultura de precisão e

detecção de mudanças na superfície terrestre.

2.1.2 Sensor TROPOMI do Sentinel-5P

O Sensor de Monitoramento Troposférico por Imagens (TROPOMI) é um dos sensores mais avançados embarcados a bordo do satélite Sentinel-5P, lançado em outubro de 2017 como parte do programa Copernicus da Agência Espacial Europeia (ESA). O TROPOMI opera em uma faixa de espectro ultravioleta, visível e infravermelho próximo e é capaz de medir uma ampla variedade de poluentes atmosféricos, incluindo dióxido de nitrogênio (NO2), dióxido de enxofre (SO2), monóxido de carbono (CO), metano (CH4) e partículas em suspensão na atmosfera (ESA. . . , 2023).

Uma das características notáveis do TROPOMI é sua alta resolução espacial, com capacidade de observação de áreas relativamente pequenas da Terra, permitindo uma visão detalhada de regiões críticas, como áreas urbanas e locais de interesse ambiental. Além disso, o sensor oferece uma alta resolução temporal, fornecendo dados diários que são essenciais para a monitorização e compreensão das mudanças na atmosfera da Terra ao longo do tempo (ESA. . . , 2023).

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um campo interdisciplinar que combina conceitos da matemática, estatística, ciência da computação e teoria da informação. Sua história remonta às décadas de 1940 e 1950. Alan Turing, em 1950, foi um dos primeiros a propor uma máquina teórica capaz de aprender, denominada ”Máquina Universal de Turing”. Contudo, o termo ”Aprendizado de Máquina” foi formalizado por Arthur Samuel, em 1959, quando desenvolveu um programa de xadrez que aprimorava seu desempenho à medida que jogava, antecipando os princípios do aprendizado supervisionado.

O AM funciona com base em algoritmos que permitem às máquinas aprender padrões e informações a partir de dados (MURPHY, 2021). Pode-se categorizar em três tipos principais: supervisionado, onde o modelo é treinado em dados rotulados; não supervisionado, em que o modelo é treinado em dados não rotulados buscando padrões por conta própria; e por reforço, onde o modelo interage com um ambiente dinâmico recebendo feedbacks na forma de recompensas ou penalidades. Além desses, existem outras abordagens importantes como o aprendizado semi-supervisionado e auto-supervisionado, ampliando o escopo de possibilidades e estratégias para aprimoramento dos modelos.

Atualmente, o estado da arte do AM é marcado pelo avanço em algoritmos complexos, especialmente no aprendizado profundo (*deep learning*), que utiliza redes neurais com várias camadas para extrair características e padrões dos dados. Isso impulsionou progressos significativos em áreas como processamento de linguagem natural, visão computacional,

reconhecimento de padrões e muito mais. Algumas destas técnicas foram abordadas nesse projeto e serão descritas nas subseções a seguir.

2.2.1 Lasso

O Lasso (Least Absolute Shrinkage and Selection Operator) é um algoritmo amplamente utilizado no contexto do Aprendizado de Máquina, especificamente na regressão linear. É uma técnica que visa selecionar um subconjunto de preditores relevantes para o modelo, eliminando os menos importantes, enquanto regulariza os coeficientes das variáveis de entrada (TIBSHIRANI, 1996), evitando a sobreajuste (*overfitting*). Sua fórmula é dada pela seguinte fórmula:

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.1}$$

onde β_j são os coeficientes das variáveis, x_{ij} são as observações, y_i é o valor alvo, n é o número de observações e p é o número de de preditores. O parâmetro λ controla o grau da regularização.

2.2.2 Florestas Aleatórias (*Random Forest*)

O algoritmo de Árvores Aleatórias, ou simplesmente *Random Forest*, é uma poderosa extensão do conceito de árvores de decisão na aprendizagem de máquina. Ele é conhecido por sua eficiência tanto em tarefas de classificação quanto de regressão. O *Random Forest* opera construindo uma "floresta" de árvores de decisão durante a fase de treinamento. O termo "floresta" surge do processo de combinar várias árvores independentes para criar um modelo robusto.

Cada árvore na floresta é treinada em um subconjunto aleatório dos dados de treinamento, selecionado por meio de um processo chamado *Bootstrap Aggregating*, ou simplesmente "*Bagging*" (BREIMAN, 2001). O *Bagging* é uma técnica que envolve a geração de múltiplas amostras de dados de treinamento (com reposição) e treinamento de cada árvore em uma dessas amostras. Dessa forma, cada árvore possui uma perspectiva única do conjunto de dados. Durante a fase de predição, a *Random Forest* combina as saídas de todas as árvores (média para regressão ou voto majoritário para classificação) para produzir uma resposta final.

Essa técnica mitiga o sobreajuste (*overfitting*) e aumenta a precisão do modelo. A diversidade introduzida pela aleatoriedade na escolha das amostras e características (variáveis preditoras) em cada árvore contribui para um modelo mais robusto e geralmente com melhor desempenho.

2.2.3 Modelos baseados em *Boosting*

O *boosting* é uma técnica de aprendizado de máquina que visa melhorar o desempenho preditivo combinando vários modelos mais fracos (SCHAPIRE; FREUND, 2014). Essa abordagem se destaca por construir modelos sequencialmente, atribuindo pesos diferenciados às instâncias de dados com base no desempenho dos modelos anteriores.

O algoritmo de *boosting* opera em etapas, onde cada modelo subsequente concentra-se nas instâncias classificadas erroneamente pelos modelos anteriores. Essas instâncias recebem maior peso, permitindo que o modelo subsequente se especialize em corrigir as predições incorretas do conjunto anterior.

Dois algoritmos populares de *boosting* são LightGBM (KE et al., 2017) e XGBoost (CHEN; GUESTRIN, 2016). O LightGBM é uma implementação eficiente de *boosting* baseado em árvores, projetado para treinamento rápido e eficaz, sendo especialmente eficiente em grandes conjuntos de dados. Já o XGBoost, uma abreviação de Extreme Gradient Boosting, é um algoritmo versátil que também utiliza árvores de decisão, incorporando técnicas avançadas de regularização e otimização para alcançar alto desempenho.

No contexto deste trabalho, esses algoritmos de *boosting*, LightGBM e XGBoost, foram escolhidos e testados para avaliar sua eficácia na na predição da concentração de NO2.

2.2.4 Convolução em 1 dimensão (Conv1D)

A convolução, uma operação matemática utilizada em processamento de sinais e redes neurais, desempenha um papel na extração de padrões e características (NUSSBAUMER, 2013). Em termos simples, essa operação combina duas funções para produzir uma terceira, revelando como uma influencia a forma da outra.

A fórmula geral da convolução discreta é expressa por:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n-m] \tag{2.2}$$

onde f e g são funções discretas, e $*$ denota a operação de convolução.

Quando aplicada em uma dimensão, como em séries temporais, f geralmente representa os dados de entrada, enquanto g é denominado "filtro" ou "kernel". A convolução 1D envolve deslizar o filtro ao longo da dimensão da série temporal, multiplicando e somando valores correspondentes em cada posição. O resultado é uma nova série temporal, destacando padrões específicos para os quais o filtro foi projetado.

Amplamente utilizada em redes neurais convolucionais (CNNs), a convolução 1D é utilizada para processar dados sequenciais, como séries temporais ou texto. Essa abordagem

permite a extração de características, proporcionando uma maneira eficaz de processar informações sequenciais em diversos domínios.

2.2.5 Unidades Recorrentes Gated (GRU)

As Unidades Recorrentes Gated (GRU) representam uma arquitetura de rede neural recorrente (RNN) aprimorada, desenvolvida por (CHO et al., 2014), com o propósito de superar os desafios comuns enfrentados pelas RNNs, como o gradiente que desaparece ou explode. A GRU introduz unidades de memória com mecanismos de controle de portão, que regulam o fluxo de informações, permitindo capturar dependências de longo prazo em sequências complexas.

A fórmula básica para uma unidade GRU é definida por equações que calculam o estado de ativação atual, incorporando o estado anterior, a entrada atual e os mecanismos de controle de portão. Esses portões determinam quais informações devem ser mantidas ou esquecidas, otimizando o fluxo eficiente de dados pela rede. Embora a GRU tenha uma arquitetura menos complexa que as LSTMs (Long Short-Term Memory), ela se mostra eficaz na modelagem de sequências temporais. As equações da GRU são dadas por:

Portão de Atualização (Update Gate):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2.3}$$

Portão de Reset (Reset Gate):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2.4}$$

Atualização de Nova Memória (New Memory Update):

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \tag{2.5}$$

Atualização do Estado Oculto (Hidden State Update):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{2.6}$$

2.2.6 Long Short-Therm Memory (LSTM)

O LSTM é uma arquitetura especial de rede neural recorrente que foi projetada para superar o problema do desvanecimento do gradiente em sequências temporais longas. Ele possui uma estrutura mais complexa que permite manter informações por longos períodos de tempo e decidir quais informações reter ou descartar.

A LSTM possui três portões essenciais: o portão de esquecimento (Forget Gate), o portão de entrada (Input Gate) e o portão de saída (Output Gate). Esses portões regulam o fluxo de informações na célula de memória. Suas equações são dadas por:

Portão de Esquecimento (Forget Gate):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]) \tag{2.7}$$

Portão de Entrada (Input Gate):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) \tag{2.8}$$

Atualização da Célula de Memória (New Cell Update):

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t]) \tag{2.9}$$

Célula de Memória Atualizada (Cell State Update):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.10}$$

Portão de Saída (Output Gate):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) \tag{2.11}$$

Estado Oculto Atualizado (Hidden State Update):

$$h_t = o_t \odot \tanh(C_t) \tag{2.12}$$

Essa arquitetura permite que a LSTM aprenda a manter e esquecer informações ao longo do tempo, sendo especialmente útil para tarefas que exigem a captura de dependências temporais de longo prazo.

2.2.7 Convolutional Long Short-Term Memory (ConvLSTM)

A ConvLSTM (Convolutional Long Short-Term Memory) (SHI et al., 2015) é uma arquitetura de redes neurais que combina as capacidades de convolução e memória de longo prazo (LSTM). Desenvolvida para lidar com dados sequenciais bidimensionais, como imagens em sequências temporais, a ConvLSTM preserva a estrutura espacial e temporal simultaneamente.

Em sua essência, a ConvLSTM integra as operações de convolução e LSTM, oferecendo uma abordagem única para processar dados sequenciais. A fórmula que define a operação ConvLSTM é dada por:

$$\begin{bmatrix} i \\ f \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (W * x_t + U * h_{t-1} + b) \tag{2.13}$$

onde i, f, o e g representam os gates de entrada, esquecimento, saída e atualização de células, respectivamente. W e U denotam pesos associados aos dados de entrada x_t e à saída anterior h_{t-1}, enquanto b é o vetor de viés. O símbolo $*$ refere-se à convolução, e σ à função sigmoide.

Na prática, a ConvLSTM encontra aplicação em diversas tarefas, especialmente na análise de imagens sequenciais e séries temporais. Sua capacidade de capturar padrões espaciais e temporais complexos torna-a valiosa em campos como reconhecimento de vídeo, previsão meteorológica e processamento de séries temporais.

2.3 Métricas de avaliação dos modelos

A seleção apropriada das métricas de avaliação é de fundamental importância em qualquer projeto de modelagem e aprendizado de máquina. Essas métricas fornecem uma maneira de quantificar o desempenho dos modelos gerados e, assim, permitem que os cientistas de dados e os pesquisadores compreendam quão bem seus modelos estão se ajustando aos dados e fazendo previsões. A variedade de métricas disponíveis é vasta, cada uma oferecendo uma perspectiva única sobre a eficácia do modelo. Portanto, é crucial entender a natureza e as implicações de várias métricas para selecionar as mais relevantes para o contexto específico do problema. Ao escolher as métricas de avaliação apropriadas, podemos ajustar nossos modelos de forma otimizada e fazer decisões informadas sobre sua utilidade prática.

2.3.1 Coficiente de Determinação

O Coeficiente de Determinação (R^2) é uma métrica essencial para avaliar a qualidade de um modelo de regressão. Ele representa a proporção da variância na variável dependente que pode ser explicada pelas variáveis independentes presentes no modelo. O valor do (R^2) varia de 0 a 1, onde 1 indica que o modelo se ajusta perfeitamente aos dados, explicando toda a variabilidade, e 0 indica que o modelo não é capaz de explicar nada da variabilidade observada. Essa métrica é fundamental para entender a eficácia do modelo na captura das variações nos dados de resposta. A fórmula para o (R^2) é dada por:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.14}$$

O R^2 é uma métrica bem estabelecida e é amplamente utilizada na avaliação de modelos de regressão.

2.3.2 Coeficiente de Correlação de Pearson (`r`)

O Coeficiente de Correlação de Pearson (Pearson, 1895) é uma métrica estatística utilizada para avaliar a relação linear entre duas variáveis contínuas. Essa métrica fornece um valor que indica a direção (positiva ou negativa) e a força da associação entre as variáveis. O coeficiente varia de -1 a 1, onde -1 representa uma correlação negativa perfeita, 0 indica ausência de correlação, e 1 representa uma correlação positiva perfeita.

A fórmula para o Coeficiente de Correlação de Pearson (`r`) entre duas variáveis X e Y é dada pela covariância entre X e Y dividida pelo produto do desvio padrão de X e Y:

$$r = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{2.15}$$

onde $cov(X, Y)$ é a covariância entre X e Y, σ_X é o desvio padrão de X, e σ_Y é o desvio padrão de Y. Essa métrica é amplamente utilizada na análise estatística para medir a intensidade e a direção de uma relação linear entre duas variáveis.

2.3.3 Erro quadrático médio (MSE)

O Erro Quadrático Médio (MSE) é uma métrica para avaliar a qualidade de um modelo em termos de suas previsões. Ele mede a média dos quadrados dos erros, ou seja, a diferença entre as previsões do modelo e os valores reais ao quadrado. Essa métrica é particularmente útil porque penaliza mais fortemente os grandes erros, proporcionando uma visão clara da precisão do modelo, sendo essencial para problemas onde erros grandes são indesejáveis. A fórmula do MSE é dada por:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.16}$$

onde n é o número de observações, y_i são os valores reais, e $\hat{y}_i$ são as previsões do modelo para essas observações.

2.3.4 Erro Médio Absoluto (MAE)

O Erro Médio Absoluto (MAE) é uma métrica para avaliar a qualidade de um modelo ao medir a magnitude média dos erros entre as previsões do modelo e os valores reais. Ao contrário do MSE, o MAE não penaliza de forma quadrática os erros, o que significa que é menos sensível a grandes erros. É uma métrica valiosa quando queremos ter uma compreensão direta da magnitude dos erros de previsão sem amplificar os efeitos dos outliers. A fórmula do MAE é dada por:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.17}$$

onde n é o número de observações, y_i são os valores reais, e $\hat{y}_i$ são as previsões do modelo para essas observações.

2.4 Revisão da literatura

A imputação de dados é uma prática essencial na área de ciência de dados, visando lidar com lacunas e valores ausentes em conjuntos de dados. Ao longo do tempo, diversas abordagens foram desenvolvidas para enfrentar esse desafio complexo, especialmente em contextos de sensoriamento remoto, onde a presença de lacunas é comum devido a variados motivos, como erros instrumentais, perda de dados durante transmissão e cobertura de nuvens.

(GOSWAMI; SANGEETA, 2015) destaca a importância da imputação espacial baseada em algoritmos de aprendizado de máquina para preencher pixels desconhecidos em imagens de satélite, abordando especificamente anomalias como quedas de varredura e sincronização inadequada do obturador para o satélite LANDSAT. O uso de abordagens, como o KNN, evidencia a necessidade de considerar a correlação entre os valores existentes no conjunto de dados para uma estimativa mais precisa.

(CHEN et al., 2022) introduz um modelo baseado em séries temporais, a In-BiLSTM, para imputação de dados em séries temporais incompletas do Sentinel-2A, enfrentando desafios específicos de nuvens durante a estação de crescimento das culturas. Ao avaliar a importância das características de entrada e visualizar unidades de estado oculto, o modelo proporciona melhorias significativas na classificação de culturas, destacando a sinergia entre imputação e classificação.

Uma abordagem diferente é feita (WANG et al., 2023) que desenvolve o STA-GAN, uma Rede Generativa Adversarial (GAN) com Atenção Espaço-Temporal, destinada à imputação de valores ausentes em dados de satélite. A atenção espaço-temporal baseada em Graph Attention Network (GAT) permite uma melhor captura de dependências temporais curtas e dinâmicas no dado, resultando em desempenho superior, especialmente em dados com grandes taxas de ausência.

Já (YANG; ZHAO; VATSAVAI, 2022) destaca como a presença de nuvens prejudica a identificação de objetos. A técnica proposta, chamada de Extended Contextual Attention (ECA), utiliza redes neurais profundas para inferir valores espectrais sob nuvens, demonstrando melhorias significativas em comparação com métodos existentes.

Por fim, (LOPS et al., 2023) propõe um modelo de rede neural convolucional parcial (PCNN) com camadas de convolução em profundidade para a estimação espaço-temporal da coluna de NO2 do TROPOMI. O modelo, treinado com dados simulados do modelo de qualidade do ar, supera métodos convencionais e outros modelos PCNN, demonstrando

consistência na reconstrução de imagens de NO2.

2.4.1 Considerações finais

A revisão da literatura revelou uma diversidade de abordagens e métodos empregados na imputação de dados em contextos de sensoriamento remoto. Os estudos analisados enfatizam a importância de técnicas sofisticadas, muitas vezes baseadas em aprendizado de máquina, para lidar com lacunas em conjuntos de dados de imagens de satélite.

A maioria dos métodos revisados utiliza informações contextuais presentes na própria imagem para realizar a imputação de valores ausentes. Contudo, essa estratégia enfrenta desafios consideráveis, especialmente em regiões como a Amazônia, onde a presença predominante de valores ausentes na imagem torna a abordagem baseada na própria imagem impraticável.

Diante dessa complexidade, a abordagem proposta neste trabalho destaca-se por sua orientação pixel-wise, alinhando-se com abordagens anteriores, como mencionado em (LOPS et al., 2023). No entanto, a inovação reside na utilização de séries temporais multivariadas de outras variáveis de sensoriamento remoto para a imputação de valores ausentes. Essa estratégia permite uma consideração mais abrangente das mudanças temporais nos dados e em regiões predominantemente influenciadas pela presença de nuvens. Assim, essa abordagem contribui para uma imputação mais precisa e robusta em ambientes desafiadores, como a região amazônica.

3 Método do trabalho

Nesta seção, descreveremos o processo de desenvolvimento adotado para este trabalho, delineando suas diferentes fases e etapas cruciais. O avanço do projeto foi guiado por uma metodologia que envolveu a especificação clara dos requisitos, o desenho da solução, a implementação efetiva e os testes necessários para garantir a funcionalidade e eficácia do sistema proposto. Cada fase foi moldada em colaboração com o orientador, garantindo a aderência aos objetivos do trabalho e uma abordagem robusta para alcançá-los. Os resultados derivados desse processo são minuciosamente detalhados nos capítulos subsequentes, apresentando uma visão holística do desenvolvimento do projeto, apoiados por referências bibliográficas pertinentes.

3.1 Estudo de Caso

O desenvolvimento deste trabalho foi fundamentado em um estudo de caso realizado no estado do Pará, Brasil. Essa abordagem foi adotada devido à complexidade computacional e à extensão territorial da Amazônia, tornando computacionalmente custoso o desenvolvimento de um modelo para toda a região. O estado do Pará foi selecionado como uma representação significativa da vasta zona de expansão agrícola no Brasil e das questões ambientais associadas.

O estudo de caso permitiu uma análise aprofundada das características ambientais e dos padrões de concentração de dióxido de nitrogênio (NO2) nessa região específica. O foco nas condições do Pará possibilitou uma compreensão mais detalhada das influências climáticas, atividades agrícolas e outras variáveis que afetam a concentração de NO2. Essa análise contextual contribuiu para a adequação do modelo às particularidades locais.

Para embasar essa metodologia, diversos artigos de referência também adotaram a abordagem de estudo de caso para o desenvolvimento de modelos em contextos ambientais e geográficos específicos. Autores como (CHAN et al., 2021), (LONG et al., 2022) e (WANG et al., 2022) utilizaram essa metodologia para analisar padrões de poluição atmosférica e contribuíram com insights valiosos para a aplicação desta abordagem no presente trabalho.

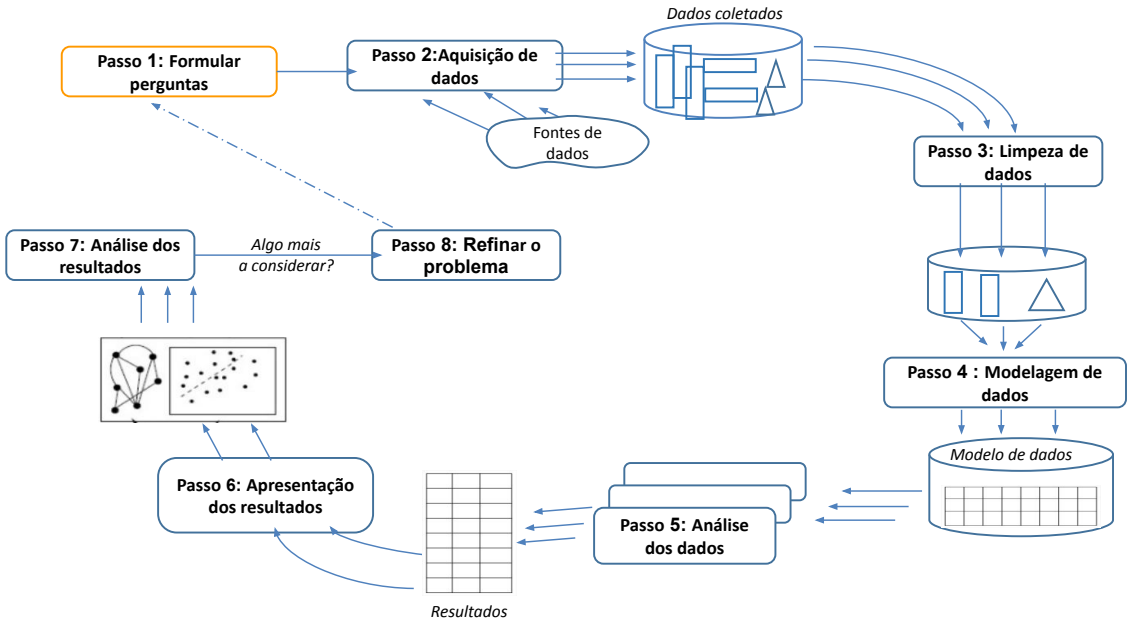
3.2 Ciclo de um experimento de dados

A realização de um experimento em ciência de dados ou aprendizado de máquina demanda um cuidadoso e estruturado ciclo de trabalho. Esse ciclo abrange desde a fase inicial de aquisição dos dados até as etapas posteriores de análise dos mesmos (DEKHTYAR,

2016). É essencial compreender a natureza dos dados, a qualidade da informação que eles contêm e como esses dados podem ser explorados para obter *insights* significativos.

Além da coleta e análise de dados, a apresentação das descobertas de maneira clara e compreensível é crucial. A disseminação eficaz dos resultados obtidos é fundamental para a compreensão das implicações e aplicabilidades das conclusões do experimento. Neste contexto, exploraremos de forma detalhada o ciclo de desenvolvimento de um experimento em ciência de dados e aprendizado de máquina, delineando as principais etapas que o constituem e fornecendo uma visão abrangente sobre como cada fase contribui para a geração de novos conhecimentos. Assim, a Figura 2 apresenta o ciclo de desenvolvimento utilizado para realização do presente trabalho.

Figura 2 – Diagrama do ciclo de experimento um experimento de dados em oito passos



Fonte: (CORRêA, 2022)

3.2.1 Formular perguntas

Nesta etapa, denominada “Formular as Perguntas”, busca-se definir de maneira clara e precisa quais são as questões e objetivos que o experimento visa responder ou atingir. Isso implica na identificação das metas e na formulação de hipóteses que possam ser validadas ou refutadas ao longo do processo. É fundamental estabelecer os parâmetros e métricas de sucesso que serão utilizados para avaliar o desempenho do experimento e alcançar conclusões significativas. Este estágio é importante para guiar todo o ciclo de trabalho e garantir que a análise seja direcionada e focada nos aspectos relevantes para a obtenção de insights valiosos.

3.2.2 Aquisição de Dados

Na etapa de "Aquisição de Dados", o foco está na obtenção dos dados necessários para responder às perguntas formuladas na etapa anterior. Isso envolveu a coleta de dados brutos de fontes diversas, a partir da API do Google Earth Engine em Python. A qualidade e a relevância dos dados são essenciais nessa etapa, pois influenciam diretamente a validade e a confiabilidade das análises posteriores.

Ao final, o objetivo é ter um conjunto de dados pronto e preparado para a limpeza, análise e processamento nas próximas etapas do ciclo.

3.2.3 Limpeza de Dados

O foco desta etapa é garantir que o conjunto de dados obtido esteja em um estado adequado para análise. Isso envolve a identificação, correção e remoção de quaisquer problemas, erros ou inconsistências nos dados que possam afetar a qualidade ou a interpretação dos resultados. Alguns procedimentos comuns incluem lidar com valores ausentes, corrigir erros de digitação, remover duplicatas e normalizar os dados. Além disso, para o caso específico de imagens de satélite, essa etapa pode incluir redimensionamento dos dados e reprojeção do sistema de coordenadas.

A finalidade desta etapa é ter um conjunto de dados limpo, consistente e pronto para ser utilizado nas análises subsequentes.

3.2.4 Modelagem de Dados

A "Modelagem de Dados" consiste em criar estruturas e representações que permitam extrair padrões e informações úteis dos dados. Isso envolve a escolha de algoritmos e técnicas de modelagem adequadas para o problema em questão, bem como a configuração dos parâmetros desses modelos.

Os passos típicos nesta etapa incluem a seleção e aplicação de algoritmos de aprendizado de máquina, definição das *features* relevantes para o modelo, treinamento e avaliação dos modelos escolhidos no conjunto de validação. Ao final desta etapa, espera-se ter modelos bem ajustados e prontos para serem usados na análise de dados.

3.2.5 Análise de Dados

O foco da "Análise de Dados" está na interpretação e exploração dos resultados obtidos a partir dos modelos e das informações contidas nos dados. Esta etapa é fundamental para gerar *insights* e tomar decisões informadas com base nos resultados, além de garantir a consistência e acuracidade dos resultados obtidos na etapa anterior.

As principais atividades desenvolvidas envolvem a visualização dos dados para identificar tendências, padrões e relações, a interpretação dos resultados dos modelos de aprendizado de máquina, a aplicação de técnicas estatísticas para validar hipóteses e a avaliação crítica dos resultados alcançados. Assim, o objetivo final da etapa de Análise de Dados é traduzir os resultados obtidos em informações valiosas e compreensíveis, fornecendo uma base sólida para tomadas de decisão e resolução das questões levantadas etapa 1.

3.2.6 Apresentação de Resultados

Após a análise dos dados, é importante realizar a "Apresentação dos Resultados". O objetivo dessa etapa é comunicar de forma clara e eficaz as descobertas e conclusões alcançadas durante o processo. Essa comunicação pode ocorrer por meio de relatórios, visualizações gráficas, apresentações orais, ou outras formas adequadas de transmissão de informação.

Portanto, o propósito é garantir que as descobertas e conclusões sejam comunicadas de maneira clara, compreensível e impactante.

3.2.7 Análise de resultados

A "Análise de Resultados" consiste em examinar e interpretar os dados processados e os modelos desenvolvidos. Esta etapa é essencial para extrair insights valiosos e avaliar o desempenho dos modelos em relação aos objetivos do projeto.

As etapas desenvolvidas nesta fase envolvem a análise estatística e quantitativa dos resultados, a comparação com métricas de desempenho previamente definidas, a identificação de padrões ou tendências relevantes e a avaliação de quão bem os modelos atendem aos critérios estabelecidos. Além disso, esta etapa diferencia-se da análise de dados devido ao envolvimento dos *stakeholders* para análise e comparação após a apresentação dos resultados.

Como resultados finais, espera-se obter uma compreensão aprofundada do comportamento dos modelos e dos padrões nos dados, a fim de validar a eficácia das abordagens adotadas, identificar possíveis melhorias e fornecer *insights* para a tomada de decisões informadas e futuras iterações do projeto tanto do ponto de vista técnico quanto de negócio.

3.2.8 Refinar o problema

Por fim, na fase de "Refinar o Problema", a atenção volta-se para uma avaliação dos resultados e do processo geral do projeto. Essa fase visa aprimorar a compreensão do problema inicialmente definido e otimizar as estratégias adotadas.

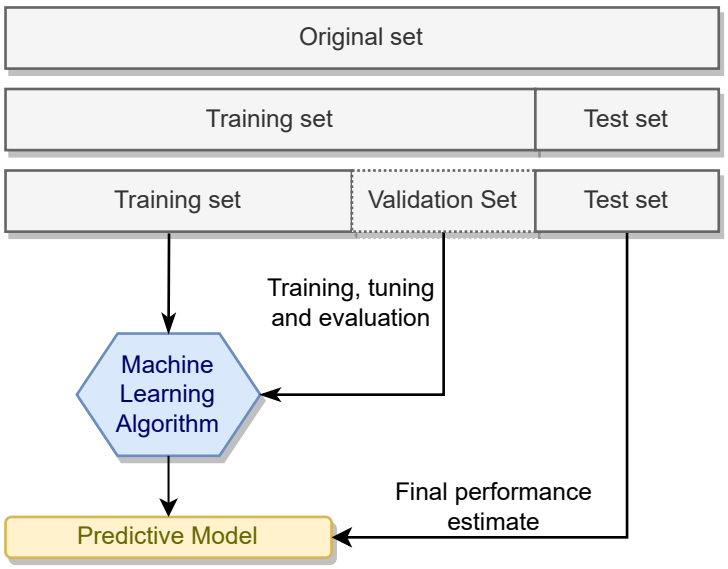
As etapas dessa fase envolvem uma revisão dos resultados obtidos, uma análise das limitações e desafios enfrentados durante o experimento, bem como uma avaliação do impacto das conclusões no contexto original do problema. Além disso, busca-se identificar possíveis melhorias, ajustes nos modelos ou nas abordagens adotadas, levando em consideração o aprendizado adquirido ao longo do projeto.

Dessa forma, a meta da etapa é garantir que o experimento tenha contribuído efetivamente para a compreensão do problema em questão, resultando em *insights* relevantes e úteis para aplicações futuras ou novas iterações do projeto. Isso inclui aperfeiçoar as estratégias de modelagem, a coleta de dados, as técnicas de análise e a interpretação dos resultados para enfrentar de forma mais precisa e eficaz o desafio inicial proposto.

3.3 Validação por *Holdout*

Para realizar a validação dos modelos, foi adotado o método *Holdout*, uma abordagem comum para avaliar o desempenho de modelos de aprendizado de máquina. Neste método, o conjunto de dados foi dividido em dois subconjuntos mutuamente exclusivos: um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento foi utilizado para treinar os modelos, realizar a busca de hiperparâmetros e otimizar o modelo, enquanto o conjunto de teste foi reservado para avaliar o desempenho dos modelos após o treinamento. A figura 3 exemplica o método de validação *holdout*.

Figura 3 – Fluxograma do processo de desenvolvimento de um modelo utilizando o método de validação por *Holdout*



Fonte: Do autor

Vale ressaltar que a divisão dos dados entre treinamento e teste foi realizada de

forma estratificada, garantindo que as proporções das classes do conjunto de dados original fossem mantidas em ambos os subconjuntos. Essa estratificação é importante para evitar viés na avaliação do modelo.

Os dados de treinamento foram utilizados para ajustar os parâmetros dos modelos, enquanto os dados de teste foram empregados para avaliar a capacidade de generalização dos modelos em relação a dados não vistos. Além disso, a escolha pelo método Holdout em detrimento da validação cruzada (Cross-Validation) se baseou nas características do conjunto de dados, no tamanho da amostra disponível e na eficiência computacional. O método Holdout, com sua simplicidade e menor demanda computacional, se mostrou apropriado para a análise dos modelos desenvolvidos neste estudo.

3.4 Especificação de Requisitos

Nesta seção, apresentaremos os requisitos do sistema para o desenvolvimento bem-sucedido do projeto. Os requisitos do sistema constituem os critérios necessários que o software ou solução deve atender para garantir sua eficiência, desempenho e funcionalidade adequados. Eles abrangem aspectos como funcionalidades específicas, interfaces, desempenho, segurança, usabilidade e qualquer outra característica essencial para atender aos objetivos do projeto.

3.4.1 Requisitos Funcionais

Os requisitos funcionais delineiam as funcionalidades específicas que o sistema deve oferecer. Para este projeto, os requisitos funcionais são:

Aquisição e Pré-Processamento dos Dados: Um dos principais requisitos funcionais deste projeto é a aquisição e pré-processamento dos dados provenientes do sensor TROPOMI e de outras fontes de sensoriamento remoto disponíveis por meio do Google Earth Engine. Essa etapa é crucial para obter dados de entrada de qualidade para a aplicação dos modelos de aprendizado de máquina.

Implementação de Modelos de Aprendizado de Máquina: Este requisito funcional envolve a implementação de modelos de aprendizado de máquina com o propósito de estimar a concentração de NO2 com base nos dados adquiridos. Os modelos desempenham um papel central no projeto, fornecendo as previsões desejadas.

Disponibilização de dataset pré-processado: Um dos objetivos finais do projeto é disponibilizar um dataset dos dados de NO2 para o estado do Pará, com os dados faltantes e com baixa qualidade imputados pelo modelo treinado.

3.4.2 Requisitos Não-Funcionais

Os requisitos não-funcionais especificam características de desempenho, usabilidade, segurança e outras qualidades do sistema. Para este projeto, os requisitos não-funcionais são:

Garantia de Correlação Mínima Aceitável: Este requisito não funcional estabelece que o modelo deve alcançar um coeficiente de correlação de Pearson (`r`) de pelo menos 0.5 na estimativa da concentração de NO2. Isso assegura que os modelos implementados sejam capazes de fornecer previsões minimamente precisas e confiáveis.

Eficiência do Tempo de Predição: Requisito não funcional que define que o modelo deve ser capaz de fazer previsões em um tempo adequado. Isso é fundamental para permitir a criação de um grid abrangendo todo o estado do Pará de forma eficiente.

Os requisitos do sistema guiarão o desenvolvimento deste projeto. Eles estabelecem as metas e critérios que devem ser alcançados para garantir um modelo eficiente, confiável e funcional. A correta definição e compreensão desses requisitos são essenciais para o sucesso do desenvolvimento e para a entrega de uma solução que atenda amplamente às necessidades e expectativas estabelecidas.

4 Desenvolvimento do Trabalho

Neste capítulo, apresentaremos o processo de desenvolvimento do trabalho, dividido nas seguintes seções: Tecnologias Utilizadas, onde discutiremos as ferramentas e linguagens escolhidas para o projeto; Projeto e Implementação, abordando a concepção e criação efetiva do modelo e pipeline de tratamento dos dados; e, por fim, Testes e Avaliação, onde verificaremos a eficácia e desempenho do modelo desenvolvido.

4.1 Tecnologias Utilizadas

Nessa seção são discutidos aspectos tecnológicos empregados no presente projeto, como o uso de bibliotecas, *frameworks* e linguagens de programação.

4.1.1 Linguagem de programação Python

O Python (FOUNDATION, 2023) é uma linguagem de programação de alto nível, interpretada e de propósito geral, amplamente adotada na comunidade de ciência de dados e aprendizado de máquina. A linguagem foi adotada para o desenvolvimento desse projeto, pois o Python oferece uma rica gama de bibliotecas especializadas, como TensorFlow, Keras, PyTorch e scikit-learn, que simplificam o desenvolvimento, treinamento e avaliação de modelos. Sua sintaxe clara e concisa facilita o desenvolvimento rápido e a experimentação, acelerando a iteração no processo de construção e refinamento de modelos. Além disso, a vasta comunidade de desenvolvedores contribui para uma extensa documentação e uma ampla gama de recursos, fóruns e tutoriais.

4.1.2 Google Earth Engine API

A API do Google Earth Engine (GOOGLE, 2023) é uma ferramenta que permite aos usuários acessar e analisar uma vasta quantidade de dados geoespaciais, incluindo imagens de satélite, de forma eficiente e escalável. Essa ferramente foi escolhida devido a capacidade da API de processar e analisar grandes conjuntos de dados espaciais diretamente na nuvem, evitando a necessidade de downloads locais e facilitando a manipulação de dados de alta resolução. Além disso, a Earth Engine oferece acesso a um amplo espectro de dados de sensoriamento remoto, possibilitando análises complexas e avançadas em diversas áreas.

4.1.3 Bibliotecas Principais

Para o desenvolvimento do código, foram utilizadas diversas bibliotecas fundamentais que oferecem uma ampla gama de funcionalidades para manipulação, análise e

visualização de dados, bem como implementação eficaz de algoritmos de aprendizado de máquina.

- **Matplotlib**: biblioteca focada em visualização de dados em Python. Com ela, é possível criar gráficos e visualizações de alta complexidade, essenciais para entender os padrões e tendências nos dados.
- **MLflow**: biblioteca Python que ajuda a gerenciar o ciclo de vida de projetos de Machine Learning, facilitando o rastreamento de experimentos, a gestão de modelos e a automação do deployment.
- **Numpy**: é uma biblioteca essencial para computação científica em Python. Ela fornece uma variedade de funções e estruturas de dados para operações numéricas eficientes, sendo fundamental para manipulação de arrays multidimensionais, operações matemáticas e álgebra linear.
- **Pandas**: utilizada para manipulação e análise de dados. Ela oferece estruturas de dados como DataFrames, ideais para trabalhar com conjuntos de dados tabulares, permitindo operações eficientes, limpeza e preparação de dados.
- **PyTorch**: é uma biblioteca amplamente utilizada para implementação eficiente de modelos de aprendizado profundo. Ela oferece ferramentas poderosas para construir e treinar redes neurais, com suporte para computação em GPU, facilitando o processamento paralelo e acelerando o treinamento dos modelos.
- **Scikit-Learn**: é uma das bibliotecas mais usadas para aprendizado de máquina em Python. Ela oferece implementações eficazes de vários algoritmos de aprendizado supervisionado e não supervisionado, além de ferramentas para avaliação de modelos e pré-processamento de dados.
- **Shapely**: biblioteca fundamental para trabalhar com geometria e geoespacialidade. Ela oferece estruturas e operações para manipulação e análise de objetos geométricos, sendo especialmente útil em aplicações que envolvem dados geoespaciais.

4.2 Aquisição dos dados e pré-processamento

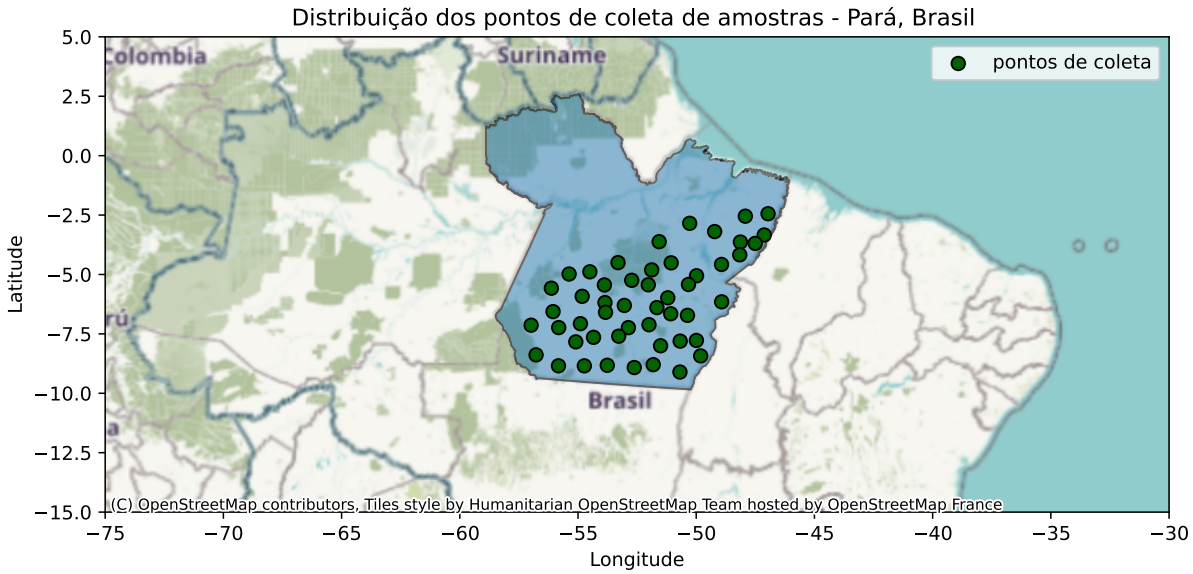
No decorrer da implementação do projeto, alinhado ao ciclo de um experimento de dados, após a formulação das questões, a etapa subsequente é a aquisição dos dados. Nesse contexto, focalizado no objetivo de estimar a concentração de NO2 na coluna troposférica por meio do sensor TROPOMI, foram identificadas variáveis potencialmente correlacionadas com a presença desse gás, conforme será discutido a seguir.

4.2.1 Seleção dos pontos do Estudo de Caso

Para a etapa de treinamento do modelo, optou-se por selecionar aleatoriamente 50 pontos dentro da área de interesse no estado do Pará, Brasil. Essa quantidade foi definida para garantir um conjunto mínimo de amostras viável para a aplicação de modelos de Machine Learning, levando em consideração também as limitações computacionais disponíveis para o treinamento do modelo.

A Figura 4 ilustra a distribuição desses 50 pontos ao longo do estado do Pará, fornecendo uma visualização da representatividade da amostra na área de estudo.

Figura 4 – Mapa da distribuição geográfica dos pontos de coleta de amostras no estado do Pará, Brasil



Fonte: Produzido pelos autores

4.2.2 Definição das variáveis independentes e do *target*

Para compilar as variáveis de entrada necessárias para o treinamento dos modelos, alguns critérios foram levados em consideração. Inicialmente, o pipeline foi construído com base na API do Google Earth Engine em Python, o que implicou em uma limitação inicial ao conjunto de variáveis disponíveis, restringindo-se às opções oferecidas pelo GEE.

Além disso, a resolução temporal diária dos dados fornecidos pelo TROPOMI para o NO2 foi um fator determinante na seleção das variáveis independentes. Nesse contexto, somente as variáveis que possuem resolução temporal diária, como os dados do NO2, foram utilizadas como variáveis independentes no processo.

Com o escopo limitado às variáveis com resolução espacial diária, o próximo passo foi a identificação daquelas que poderiam ter maior correlação com a concentração de NO2, levando em consideração suas características físicas. Essa análise será detalhada posteriormente, descrevendo para cada variável sua relação com a concentração de NO2.

Assim, de maneira resumida, os 3 critérios para escolha das variáveis independentes foram:

1. Disponibilidade do dado via API do Google Earth Engine;
2. Resolução temporal diária;
3. Possível correlação/influência com a concentração da coluna troposférica de NO2 do ponto de vista físico.

A Tabela 1 resume o conjunto de variáveis que foram identificadas como sendo as mais relevantes para servirem como variáveis independentes no treinamento dos modelos. A análise detalhada de cada variável e sua relação com a concentração de NO2 será apresentada nas próximas subseções.

Tabela 1 – Conjunto de variáveis independentes e dependente do dataset de treinamento. A primeira linha representa o *target*. As linhas subsequentes indicam as variáveis independentes.

Nome da variável	**Código da Coleção**	**Res. espacial (m)**
`tropospheric_NO2_column_number_density`	`COPERNICUS/S5P/OFFL/L3_NO2`	1113.2
`Optical_Depth_047`	`MODIS/061/MCD19A2_GRANULES`	1000
`Column_WV`	`MODIS/061/MCD19A2_GRANULES`	1000
`precipitationCal`	`NASA/GPM_L3/IMERG_V06`	11132
`temperature_2m`	`ECMWF/ERA5_LAND/DAILY_AGGR`	11132
`evaporation_from_bare_soil_sum`	`ECMWF/ERA5_LAND/DAILY_AGGR`	11132
`volumetric_soil_water_layer_1`	`ECMWF/ERA5_LAND/DAILY_AGGR`	11132
`surface_latent_heat_flux_sum`	`ECMWF/ERA5_LAND/DAILY_AGGR`	11132
`sm_surface`	`NASA/SMAP/SPL4SMGP/007`	11000

Fonte: Produzido pelos autores.

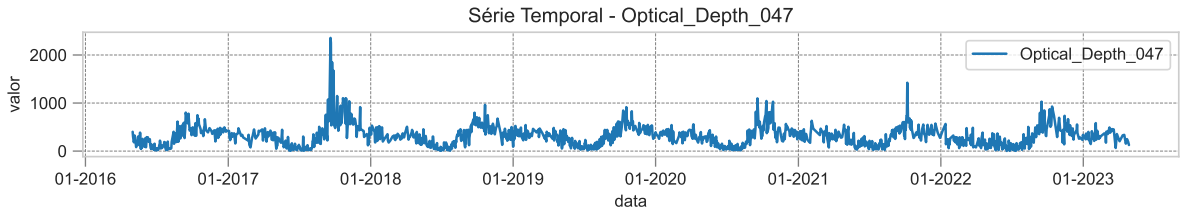
4.2.2.1 `Optical_Depth_047`

A variável `Optical_Depth_047`, que representa a profundidade óptica de aerossóis na banda azul do MODIS (0,47 μm), foi escolhida como variável dependente para estimar a concentração da coluna troposférica de NO2 devido à sua relação potencial com o NO2. Aerossóis, como partículas suspensas na atmosfera, podem interagir e afetar a concentração

de NO2. A profundidade óptica dos aerossóis na banda azul do espectro pode indicar a presença e a concentração dessas partículas, que podem influenciar a disseminação do NO2 na atmosfera. Portanto, essa variável oferece informações úteis para modelar e compreender a concentração de NO2.

Além disso, a descrição da variável menciona que, em altitudes elevadas (superiores a 4,2 km), o AOD não é registrado, a menos que sejam detectadas fumaça ou poeira, caso em que é relatado um valor estático de 0,02 para correção atmosférica. Isso sugere que a variável `Optical_Depth_047` pode ser um indicador sensível de condições atmosféricas especiais, como a presença de fumaça ou poeira, que também podem influenciar a concentração de NO2. Portanto, o uso dessa variável como variável dependente é relevante para capturar essas influências potenciais na estimativa da concentração de NO2. A Figura 5 apresenta a série temporal de uma amostra da variável `Optical_Depth_047`.

Figura 5 – Série temporal de uma amostra da variável Optical_Depth_047



Fonte: Produzido pelos autores

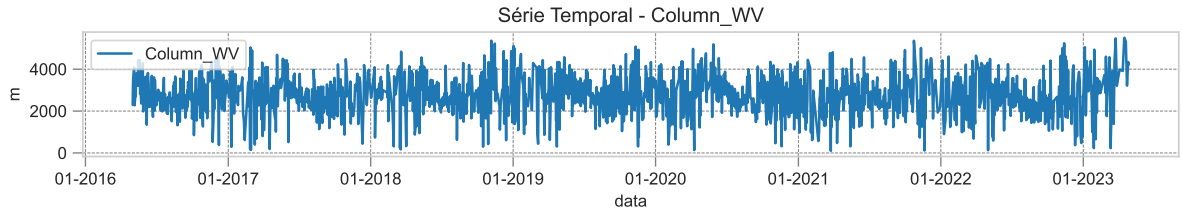
4.2.2.2 `Column_WV`

A escolha da variável `Column_WV`, que representa a quantidade de vapor d'água na coluna sobre áreas terrestres, como variável dependente para prever a concentração da coluna troposférica de NO2, está relacionada à influência do vapor d'água na atmosfera. O vapor d'água é uma das principais variáveis meteorológicas que afeta a concentração de poluentes, incluindo o NO2. A presença de vapor d'água na atmosfera pode afetar a dispersão e a concentração de NO2, uma vez que pode atuar como um "filtro" que modifica a intensidade da luz e afeta a medição do NO2 em sensores remotos. Portanto, a quantidade de vapor d'água na coluna pode fornecer informações importantes sobre as condições atmosféricas que impactam o NO2.

A descrição da variável `Column_WV` também menciona que, quando relatada para pixels nublados, ela representa o vapor d'água acima das nuvens. Isso sugere que essa variável captura não apenas o vapor d'água nas camadas inferiores da atmosfera, mas também o vapor d'água em altitudes elevadas. Essa informação é relevante, pois o vapor d'água em altitudes diferentes pode ter diferentes influências na concentração de NO2,

dependendo das condições meteorológicas. A Figura 6 apresenta a série temporal de uma amostra da variável `Column_WV`.

Figura 6 – Série temporal de uma amostra da variável Column_WV

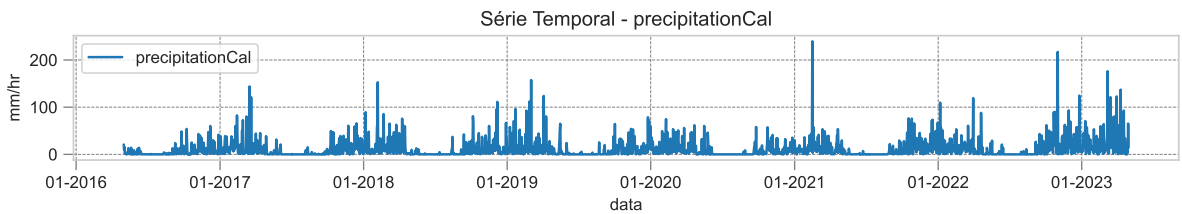


Fonte: Produzido pelos autores

4.2.2.3 precipitationCal

A escolha de utilizar a variável precipitationCal, que representa a precipitação calibrada, como variável dependente na estimativa da concentração da coluna troposférica de NO2, é respaldada pela influência das condições meteorológicas, como a chuva, sobre a concentração de poluentes atmosféricos, em particular o NO2. O NO2 é um gás altamente reativo, suscetível a mudanças nas condições da superfície e da atmosfera. As condições de chuva desempenham um papel crucial na remoção e deposição de poluentes, incluindo o NO2. A precipitação pode agir como um agente de limpeza, removendo partículas e gases poluentes da atmosfera, o que pode resultar em variações na concentração de NO2. Além disso, a chuva também pode afetar a dispersão e diluição de poluentes, influenciando indiretamente a concentração de NO2 em diferentes regiões. Portanto, ao incorporar a variável precipitationCal na modelagem, levamos em consideração essas interações complexas entre a chuva e a concentração de NO2 em escala global. A Figura 7 apresenta a série temporal de uma amostra da variável `precipitationCal`.

Figura 7 – Série temporal de uma amostra da variável precipitationCal



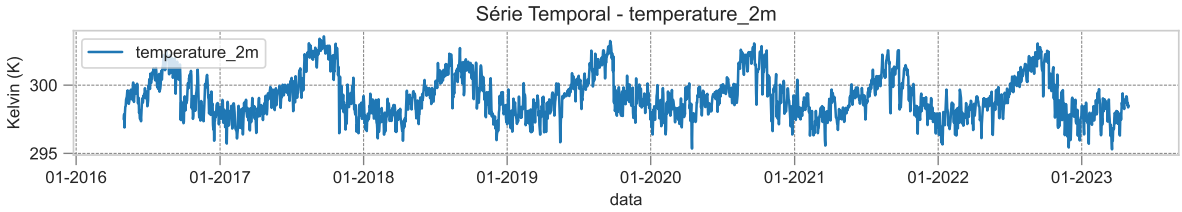
Fonte: Produzido pelos autores

4.2.2.4 `temperature_2m`

A variável `temperature_2m` é importante na previsão de NO2 devido à sua influência nas reações químicas do NO2, com temperaturas mais altas acelerando essas reações. Temperaturas mais baixas podem reduzir a formação de NO2. Assim, monitorar a temperatura ajuda a entender a presença de NO2 na atmosfera e como ela afeta a qualidade do ar. Além disso, variações na temperatura também podem influenciar o transporte e a dispersão de poluentes, o que pode afetar a concentração de NO2 em diferentes regiões.

A temperatura a 2 metros acima da superfície é uma medida direta das condições de temperatura próximas à superfície. Usá-la como variável dependente na modelagem permite avaliar a relação entre a temperatura local e o NO2. Isso é fundamental para a gestão da poluição e para compreender como as condições meteorológicas afetam a concentração de NO2. A Figura 8 apresenta a série temporal de uma amostra da variável `temperature_2m`.

Figura 8 – Série temporal de uma amostra da variável temperature_2m

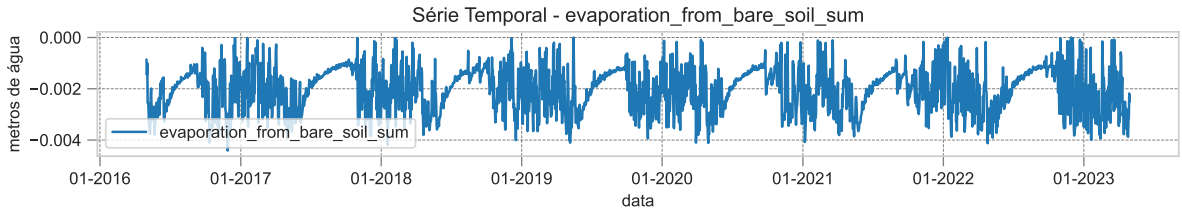


Fonte: Produzido pelos autores

4.2.2.5 `evaporation_from_bare_soil_sum`

A variável `evaporation_from_bare_soil_sum` é relevante para a previsão de NO2 devido à sua ligação com as condições climáticas. A evaporação do solo nu reflete a quantidade de água que evapora da superfície do solo, que pode variar com a temperatura, umidade e radiação solar. A presença de NO2 na atmosfera é sensível a essas condições meteorológicas, pois a temperatura e a umidade afetam as reações químicas que produzem e removem o NO2. Quando a evaporação do solo nu aumenta, pode indicar um clima mais quente e seco, o que, por sua vez, pode afetar a formação e a dispersão de NO2 na atmosfera. Portanto, monitorar a evaporação do solo nu pode fornecer informações importantes sobre as condições que influenciam a concentração de NO2. A Figura 9 apresenta a série temporal de uma amostra da variável `evaporation_from_bare_soil_sum`.

Figura 9 – Série temporal de uma amostra da variável evaporation_from_bare_soil_sum



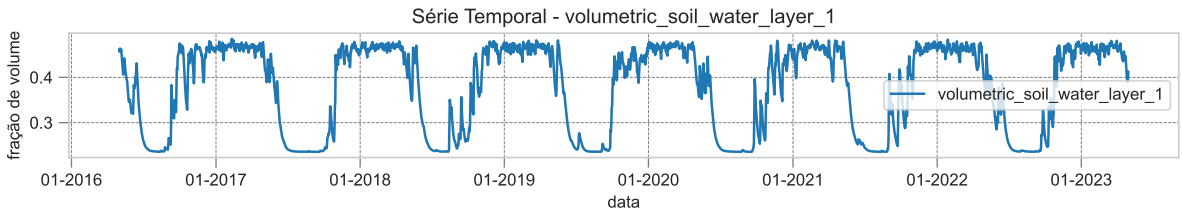
Fonte: Produzido pelos autores

4.2.2.6 `volumetric_soil_water_layer_1`

A variável `volumetric_soil_water_layer_1` é uma escolha relevante como variável dependente para estimar a concentração de NO2 devido à sua conexão com as condições do solo. Essa variável representa o volume de água na camada superior do solo, que possui uma profundidade de 0 a 7 cm. A quantidade de água nessa camada do solo é crucial, pois influencia diretamente a umidade do solo, que, por sua vez, afeta a química atmosférica e a dispersão de poluentes, como o NO2. A água no solo pode atuar como um filtro natural para componentes atmosféricos, modificando sua concentração na coluna troposférica, o que faz dessa variável um preditor relevante.

O volume de água na camada superior do solo pode refletir a capacidade do solo de reter e liberar umidade, impactando as condições do ar e as reações químicas envolvendo o NO2. A relação entre a umidade do solo e a concentração de NO2 é complexa, pois a umidade pode influenciar a deposição seca e úmida do NO2 e sua produção por processos químicos no solo. A Figura 10 apresenta a série temporal de uma amostra da variável `volumetric_soil_water_layer_1`.

Figura 10 – Série temporal de uma amostra da variável volumetric_soil_water_layer_1



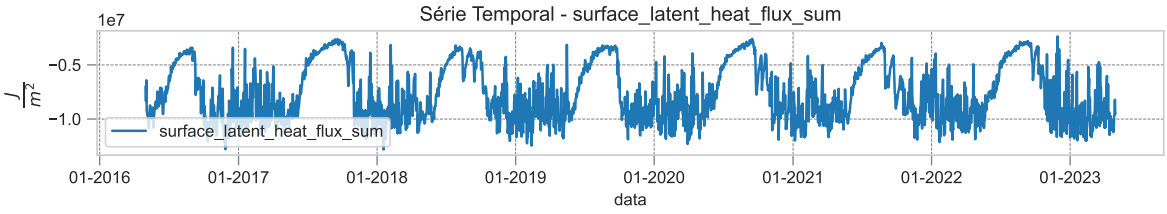
Fonte: Produzido pelos autores

4.2.2.7 `surface_latent_heat_flux_sum`

Essa variável representa a troca de calor latente com a superfície por meio de processos de difusão turbulenta. A troca de calor latente é uma parte fundamental dos processos de evaporação e condensação, que estão diretamente relacionados à umidade do solo e da atmosfera. Esses processos influenciam a dinâmica atmosférica e as condições meteorológicas, afetando a dispersão e a concentração de poluentes atmosféricos, como o NO2.

A relação entre a troca de calor latente e a concentração de NO2 está relacionada à dinâmica atmosférica. As variações na troca de calor latente podem afetar a formação de nuvens, a circulação atmosférica e a dispersão de poluentes. Além disso, a evaporação da água da superfície terrestre para a atmosfera pode influenciar a concentração de NO2, uma vez que pode afetar a umidade relativa e a formação de aerossóis, que são precursores de poluentes. A Figura 11 apresenta a série temporal de uma amostra da variável `surface_latent_heat_flux_sum`.

Figura 11 – Série temporal de uma amostra da variável surface_latent_heat_flux_sum



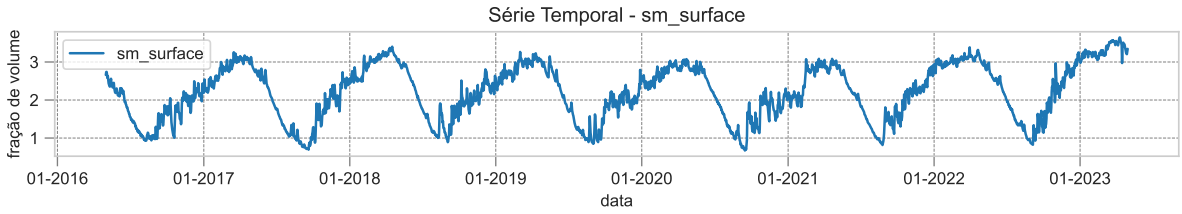
Fonte: Produzido pelos autores

4.2.2.8 `sm_surface`

Essa variável representa a umidade do solo que desempenha um papel crucial na regulação das condições atmosféricas e, portanto, afeta a formação e dispersão de poluentes atmosféricos, incluindo o NO2. A umidade do solo influencia a taxa de evaporação da água da superfície terrestre para a atmosfera, que por sua vez pode afetar a umidade relativa do ar, a formação de nuvens e a circulação atmosférica. Esses fatores têm impacto direto na concentração de NO2, pois afetam a dinâmica atmosférica e a dispersão de poluentes.

Além disso, a umidade do solo na camada superior também está relacionada à disponibilidade de água para as plantas e à sua atividade transpiratória, o que pode influenciar a produção de compostos orgânicos voláteis (COVs) que reagem na atmosfera para formar NO2. A Figura 12 apresenta a série temporal de uma amostra da variável `sm_surface`.

Figura 12 – Série temporal de uma amostra da variável sm_surface

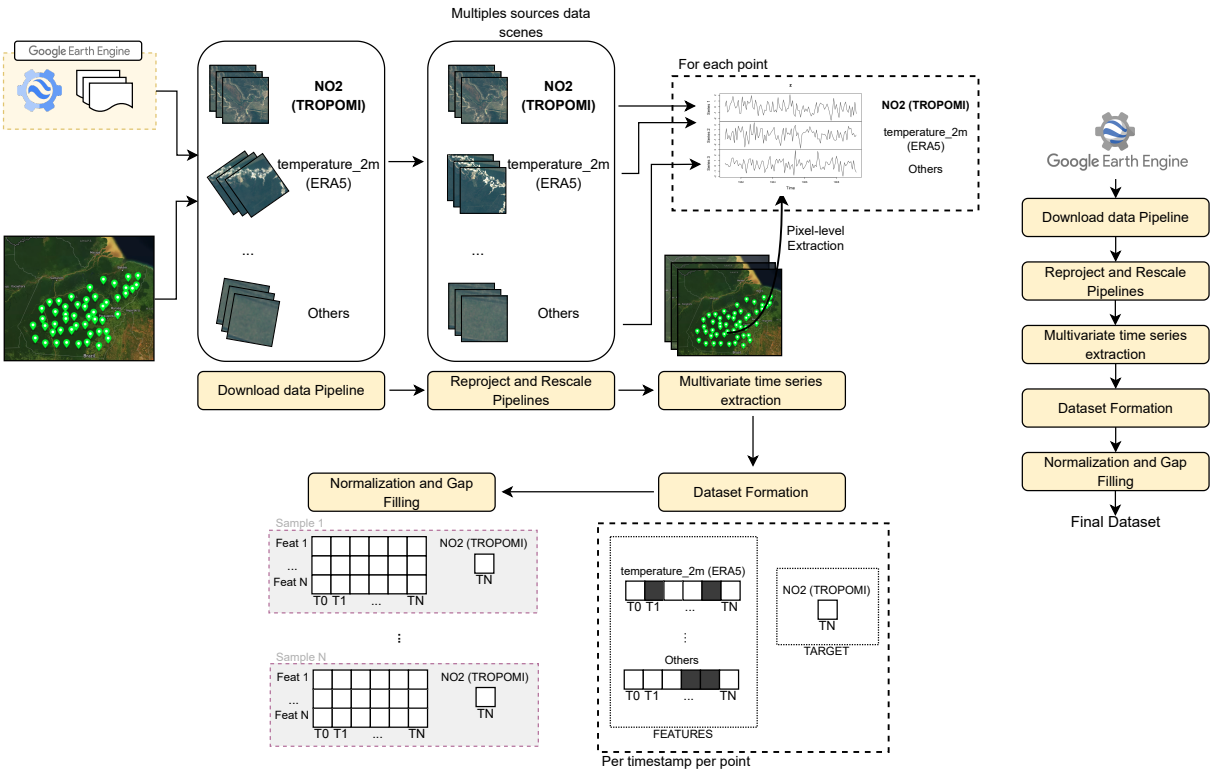


Fonte: Produzido pelos autores

4.2.3 Pré-processamento e criação do *dataset* de treinamento

Foi elaborado um pipeline para efetuar o download dos dados, bem como para realizar a limpeza e pré-processamento necessários. Esse pipeline tem como objetivo formatar o conjunto de dados final que será empregado no treinamento dos modelos. A Figura 13 ilustra o pipeline desenvolvido, o qual será explorado detalhadamente nas seções subsequentes.

Figura 13 – Fluxograma do pipeline de download e pré-processamento de dados para montagem do dataset de treinamento



Fonte: Produzido pelos autores

4.2.3.1 Download dos dados

Nesta etapa inicial, os dados são adquiridos de maneira automatizada através da API do Google Earth Engine. Um arquivo base em formato YAML contendo os metadados necessários é utilizado para orientar o download das características (*features*) e do alvo (*target*) do modelo. Esse arquivo fornece informações relevantes para aquisição das bandas específicas e, quando aplicável, das bandas de controle de qualidade dos dados.

4.2.3.2 Reprojeção e rescala dos dados

Para lidar com a heterogeneidade dos dados provenientes de diferentes fontes de sensoriamento remoto, é fundamental um processo de reprojeção e redimensionamento. Diferentemente de dados RGB convencionais utilizados em modelos de Visão Computacional, os dados de sensoriamento remoto são georreferenciados, ou seja, possuem referências geográficas e temporais para cada pixel da imagem.Essas informações variam de acordo com o satélite e sensor, levando a sistemas de referência espacial (CRS) distintos e resoluções variadas.

Assim, é necessário realizar uma padronização para garantir que todas as variáveis estejam no mesmo grid e tenham a mesma resolução espacial. Esse processo é realizado através de redimensionamento e reprojeção, utilizando o CRS da variável `precipitationCal` da NASA como base. Essa escolha se deve à necessidade de selecionar uma das variáveis com menor resolução para degradar a resolução das demais, permitindo uniformizá-las em um mesmo grid. A etapa de redimensionamento e reprojeção foi executada com o auxílio da API em Python do Google Earth Engine.

Ao final desse processo, todas as variáveis estavam padronizadas, com resolução de 11132 m/pixel e no CRS da variável `precipitationCal`.

4.2.3.3 Extração das Séries Temporais

Após a organização das imagens no mesmo grid, inicia-se a etapa de extração das séries temporais baseadas nos pontos de coleta de dados. Essas séries temporais são obtidas a nível de pixel, utilizando o valor do pixel correspondente ao ponto de coleta de amostras. A extração abrange as nove variáveis listadas na Tabela 13, incluindo variáveis independentes e dependente. As séries temporais das variáveis independentes são extraídas para o período de junho de 2017 (um ano antes do início dos dados provenientes do TROPOMI) a abril de 2023 (momento inicial da coleta dos dados para treinamento do modelo).

4.2.3.4 Formatação do Dataset de Treinamento

Para a criação do dataset de treinamento, são selecionadas as séries temporais das variáveis independentes em um período de 365 dias (1 ano). Essas séries temporais são utilizadas para estimar a concentração de NO2 do TROPOMI para o dia mais recente da série temporal. Por exemplo, as séries temporais das variáveis independentes que abrangem o período de 21/10/2021 a 21/10/2022 são utilizadas para estimar a concentração de NO2 do TROPOMI para o dia 21/10/2022. O período de 365 dias para as variáveis independentes foi escolhido de forma arbitrária, permitindo a possibilidade de experimentos futuros para determinar o melhor período de lag, ou seja, qual o tamanho ideal da série temporal que ainda contribui com informações relevantes para o treinamento do modelo.

Assim, ao final dessa etapa os dados são separados em um conjunto de séries temporais multivariadas e seu respectivo *target*.

4.2.3.5 Normalização e Imputação de Valores Faltantes

Na última etapa do pipeline de pré-processamento, são realizadas duas operações: imputação de valores faltantes e normalização dos dados. Inicialmente, é feita a imputação dos valores ausentes. Para manter a simplicidade e assegurar a completude dos dados, utilizou-se um método de imputação pela média dos valores para cada variável. Os valores faltantes são substituídos pela média dos valores observados daquela variável.

Após a imputação, é aplicada a normalização dos dados. Esta etapa é fundamental para garantir que todas as variáveis tenham uma escala similar. Utilizou-se o `StandardScaler` da biblioteca scikit-learn. A fórmula de normalização é dada por:

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Onde x é o valor da variável, μ é a média e σ é o desvio padrão. A normalização é crucial para evitar que variáveis com magnitudes muito diferentes possam influenciar de maneira desproporcional no treinamento do modelo.

A imputação e normalização dos dados são práticas comuns em aprendizado de máquina, visando ajustar os dados de entrada para os modelos. Estudos como discutido em (GOODFELLOW; BENGIO; COURVILLE, 2016) e (HASTIE; TIBSHIRANI; FRIEDMAN, 2001) ressaltam a importância dessas etapas na preparação dos dados para a construção de modelos de aprendizado de máquina e garantir a qualidade dos dados de entrada.

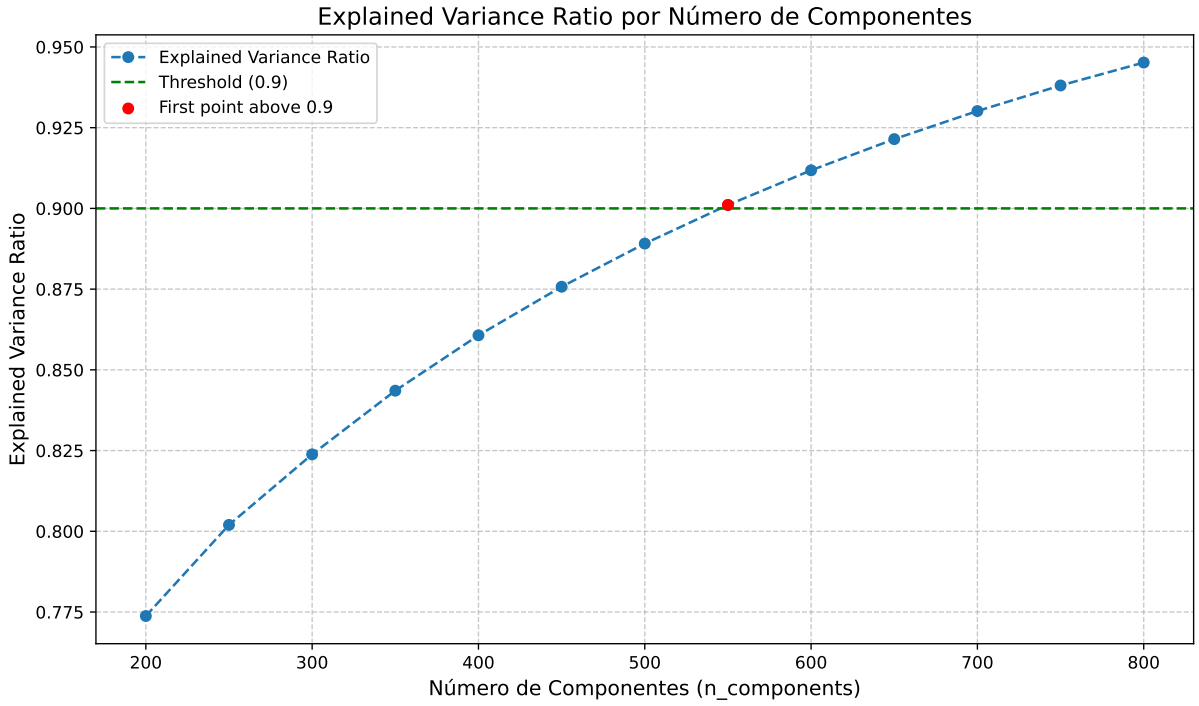
4.2.4 Redução de Dimensionalidade

A redução de dimensionalidade é uma etapa crucial no processo de preparação dos dados para o treinamento dos modelos de aprendizado de máquina. Neste projeto, aplicou-se uma técnica conhecida como Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos dados de entrada.

O PCA é uma técnica estatística que visa reduzir a quantidade de características (ou atributos) em um conjunto de dados, mantendo as informações mais significativas. Funciona encontrando novas dimensões (os componentes principais) que são combinações lineares das características originais. Esses componentes são classificados em ordem decrescente de importância, com o primeiro componente principal explicando a maior variação nos dados.

Para determinar o número de componentes principais a serem mantidos, foi definido um valor mínimo de variância explicada necessária. Neste caso, a seleção incluiu componentes principais do PCA que mantivessem pelo menos 90% (0.9) da variância explicada. Aplicando o PCA e variando o número de componentes de 50 em 50, no intervalo de 200 a 800, identificou-se que a primeira quantidade de componentes que ultrapassou os 90% de variância explicada foi de 550. A Figura 14 apresenta a variância explicada em função do número de componentes principais.

Figura 14 – Gráfico da Razão de Variância Explicada por Número de Componentes.



Fonte: Produzido pelos autores

Portanto, para os modelos de aprendizado de máquina clássicos que não utilizam

deep learning, foram utilizados os componentes extraídos do PCA que apresentaram uma variância explicada de 0.9011. Essa abordagem contribuiu para a redução da dimensionalidade dos dados de entrada, mantendo informações essenciais para a construção dos modelos.

4.3 Divisão do *dataset*

Para efetuar o treinamento dos modelos, o conjunto de dados passou por um processo de divisão, conforme detalhado na seção 3.3, que empregou o método de *holdout*. O *dataset* resultante foi segmentado em três componentes distintos: o conjunto de treinamento, o conjunto de validação e o conjunto de teste.

O conjunto de treinamento, responsável por aproximadamente 70% das amostras, abrangendo um total de 6569 observações, teve como função a formação e calibração dos modelos de machine learning. Esse conjunto serviu como base de treinamento para a construção dos modelos, permitindo o aprendizado com os dados e as relações subjacentes entre as variáveis.

Por outro lado, tanto o conjunto de validação quanto o conjunto de teste representaram cerca de 15% cada um. O conjunto de validação, composto por 1482 amostras, foi utilizado no ajuste dos hiperparâmetros dos modelos de *deep learning* e na avaliação do desempenho dos modelos a cada época de treinamento. Enquanto isso, o conjunto de teste, composto por 1445 amostras, foi estritamente reservado para a validação final e para a comparação entre os modelos desenvolvidos. A figura 15 mostra a divisão do *dataset* nos 3 conjuntos citados.

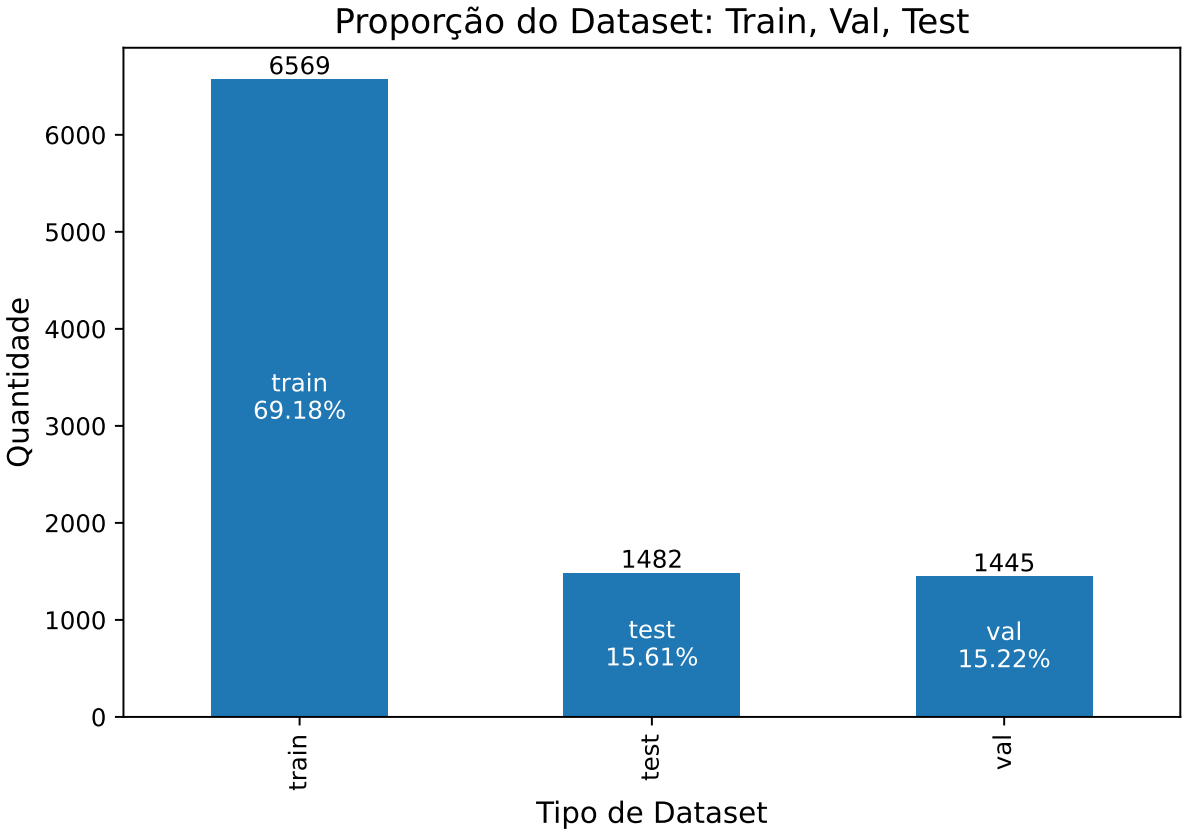
Assim, ao final, a divisão do conjunto de dados permitiu a formação de três conjuntos distintos, cada um com seu propósito específico, resultando em um total de 9496 amostras.

4.4 Treinamento dos modelos

Nesta seção, discutiremos os modelos de Machine Learning utilizados para abordar o problema de estimar a concentração da coluna troposférica de NO2. Os modelos selecionados abrangem uma variedade de técnicas, desde abordagens tradicionais até Deep Learning. Esses modelos foram definidos com base em estudos anteriores que lidaram com séries temporais multivariadas, como em (CHE et al., 2018) e (MOSKOLAï et al., 2021) e também em modelos clássicos amplamente utilizados para resolver problemas das mais diversas áreas. Os modelos escolhidos para este estudo incluem:

- Lasso;

Figura 15 – Divisão do conjunto de dados nos conjuntos de treinamento, validação e teste



Fonte: Produzido pelos autores

- Florestas aleatórias (*Random Forest*);
- Algoritmos Baseados em *Boosting*
  - LightGBM;
  - XGBoost.
- Convolução 1D;
- *Gated Recurrent Unit* (GRU);
- *Long Short-Term Memory* (LSTM);
- Convolutional Long Short-Term Memory (ConvLSTM).

Cada um desses modelos será explorado em detalhes, incluindo seus hiperparâmetros, resultados e contribuições para a estimação da concentração de NO2. Esta variedade de modelos permite uma análise abrangente do desempenho e identificação das abordagens mais eficazes para o problema em questão.

4.4.1 Lasso

Para treinar o modelo Lasso, foi utilizada a classe `Lasso` do *scikit-learn*. Durante o processo de treinamento, realizou-se uma busca em grade (*Grid Search*) para otimizar os hiperparâmetros do modelo. Os hiperparâmetros testados estão contidos na Tabela 2:

Tabela 2 – Conjunto de hiperparâmetros testados durante o grid search para o modelo Lasso.

Hiperparâmetro	**Descrição**	**Valores testados**
`alpha`	Constante que multiplica o termo L1, controlando a força da regularização.	`[0.01, 0.05, 0.1, 0.5, 1.0]`
`max_iter`	O número máximo de iterações.	`[500, 1000, 1500, 2000]`

Fonte: Produzido pelos autores.

4.4.2 Florestas aleatórias (*Random Forest*)

Para treinar o modelo, foi utilizada a classe `RandomForestRegressor` do *scikit-learn*. Assim como nos demais casos, conduziu-se um busca por melhores hiperparâmetros que estão contidos na Tabela 3.

Tabela 3 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de Florestas Aleatórias.

Hiperparâmetro	**Descrição**	**Valores testados**
`bootstrap`	Se as amostras bootstrap são usadas na construção de árvores. Se for Falso, todo o conjunto de dados será usado para construir cada árvore.	`[True, False]`
`max_depth`	A profundidade máxima da árvore.	`[20, 40, 60, 80, 100, None]`
`max_features`	O número de recursos a serem considerados ao procurar o melhor split.	`['auto', 'sqrt']`
`n_estimators`	O número de árvores na floresta.	`[100, 200, 300, 400, 500]`

Fonte: Produzido pelos autores.

4.4.3 LightGBM

O treinamento e implementação do modelo foi feita utilizando a biblioteca LightGBM (CORPORATION, 2023), que já possui suporte completo para realizar a busca de hiperparâmetros. Os hiperparâmetros testados na busca estão contidos na tabela 4.

Tabela 4 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de LightGBM.

Hiperparâmetro	Descrição	Valores testados
`boosting_type`	Define o tipo de estimador utilizado para os regressores fracos.	`['rf', 'dart', 'gbdt']`
`learning_rate`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.001, 0.005, 0.01, 0.05, 0.1, 0.5]`
`n_estimators`	Número de árvores a serem ajustadas.	`[100, 300, 500, 700, 800, 1000]`
`num_leaves`	O número de folhas por árvore.	`[31, 50, 70, 100]`

Fonte: Produzido pelos autores.

4.4.4 XGBoost

O treinamento e implementação do modelo foi feita utilizando a biblioteca XGBoost (DEVELOPERS, 2023). Além disso, utilizou-se a interface disponível com a biblioteca scikit-learn para realizar a busca de hiperparâmetros. Os hiperparâmetros testados na busca estão contidos na tabela 5.

Tabela 5 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de XGBoost.

Hiperparâmetro	Descrição	Valores testados
`colsample_bytree`	Razão de subamostragem de colunas ao construir cada árvore.	`[0.8, 0.9, 1.0]`
`eta`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.001, 0.005, 0.01, 0.05, 0.1, 0.5]`
`n_estimators`	Número de árvores a serem ajustadas.	`[200, 400, 600, 800, 1000]`
`subsample`	Razão de subamostragem das instâncias de treino.	`[0.7, 0.9, 1.0]`
`max_depth`	A profundidade máxima da árvore.	`[20, 40, 60, 80, 100, None]`

Fonte: Produzido pelos autores.

4.4.5 Convolução 1D (Conv1D)

Para implementar o modelo Conv1D, utilizou-se a biblioteca PyTorch, com o auxílio da biblioteca PyTorch-Lightning, que simplifica a lógica de treinamento e gerenciamento

do modelo. Dessa forma, o conjunto de hiperparâmetros testados estão contidos na Tabela 6.

Tabela 6 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de Conv1D.

Hiperparâmetro	**Descrição**	**Valores testados**
`learning_rate`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.01, 0.05, 0.1, 0.5]`
`batch_size`	Número de amostras de treinamento usadas em uma iteração de treino.	`[16, 32, 64, 128]`
`kernel_size`	Tamanho do kernel de convolução.	`[7, 11, 15, 19, 23]`
`filters_per_conv`	Descreve o número filtros por camada de convolução	`[(8, 2, 1), (8, 3, 1)]`

Fonte: Produzido pelos autores.

4.4.6 *Gated Recurrent Unit* (GRU)

Para implementar o modelo GRU, utilizou-se a biblioteca PyTorch, com o auxílio da biblioteca PyTorch-Lightning, que simplifica a lógica de treinamento e gerenciamento do modelo. Como um modelo de *Deep Learning*, o GRU possui hiperparâmetros específicos relacionados ao treinamento da rede, além dos relacionados à sua arquitetura. Dessa forma, o conjunto de hiperparâmetros testados estão contidos na Tabela 7.

Tabela 7 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de GRU.

Hiperparâmetro	**Descrição**	**Valores testados**
`learning_rate`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.01, 0.05, 0.1, 0.5]`
`batch_size`	Número de amostras de treinamento usadas em uma iteração de treino.	`[16, 32, 64, 128]`
`hidden_dim`	O número de *features* no estado oculto h.	`[8, 16, 32, 64]`
`n_layers`	Número de camadas recorrentes.	`[1, 2, 3]`

Fonte: Produzido pelos autores.

4.4.7 *Long Short-Term Memory* (LSTM)

Para implementar o modelo de LSTM, utilizou-se a biblioteca PyTorch, com o auxílio da biblioteca PyTorch-Lightning, que simplifica a lógica de treinamento e gerenciamento do modelo. O conjunto de hiperparâmetros testados estão contidos na Tabela 8.

Tabela 8 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de LSTM.

Hiperparâmetro	**Descrição**	**Valores testados**
`learning_rate`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.01, 0.05, 0.1, 0.5]`
`batch_size`	Número de amostras de treinamento usadas em uma iteração de treino.	`[16, 32, 64, 128]`
`hidden_dim`	O número de *features* no estado oculto h.	`[8, 16, 32, 64]`
`n_layers`	Número de camadas recorrentes.	`[1, 2, 3]`

Fonte: Produzido pelos autores.

4.4.8 Convolutional Long Short-Term Memory (ConvLSTM)

Para implementar o modelo de ConvLSTM, utilizou-se a biblioteca PyTorch, com o auxílio da biblioteca PyTorch-Lightning, que simplifica a lógica de treinamento e gerenciamento do modelo. O conjunto de hiperparâmetros testados estão contidos na Tabela 9.

Tabela 9 – Conjunto de hiperparâmetros testados durante o grid search para o modelo de ConvLSTM.

Hiperparâmetro	**Descrição**	**Valores testados**
`learning_rate`	A taxa de aprendizado utilizada para controlar o tamanho dos passos dados em direção ao mínimo da função de perda.	`[0.01, 0.05, 0.1, 0.5]`
`batch_size`	Número de amostras de treinamento usadas em uma iteração de treino.	`[16, 32, 64, 128]`
`hidden_dim`	O número de *features* no estado oculto h.	`[8, 16, 32, 64]`
`n_layers`	Número de camadas recorrentes.	`[1, 2, 3]`

Fonte: Produzido pelos autores.

4.5 Testes e Avaliação

De acordo com 3.3 descrito anteriormente, a validação foi realiza utilizando a técnica por *holdout*, na qual o conjunto de teste é estritamente reservado para a validação final.

4.5.1 Métricas individuais dos modelos

Esta subseção fornecerá uma apresentação dos resultados individuais alcançados por cada modelo, incluindo gráficos para visualização representativa dos dados.

4.5.1.1 Lasso

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 10. As métricas obtidas pelo modelo Lasso mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 11.

Tabela 10 – Hiperparâmetros para o melhor modelo Lasso após o grid search

Hiperparâmetro	**Valor**
`alpha`	0.05
`max_iter`	2000

Fonte: Produzido pelos autores.

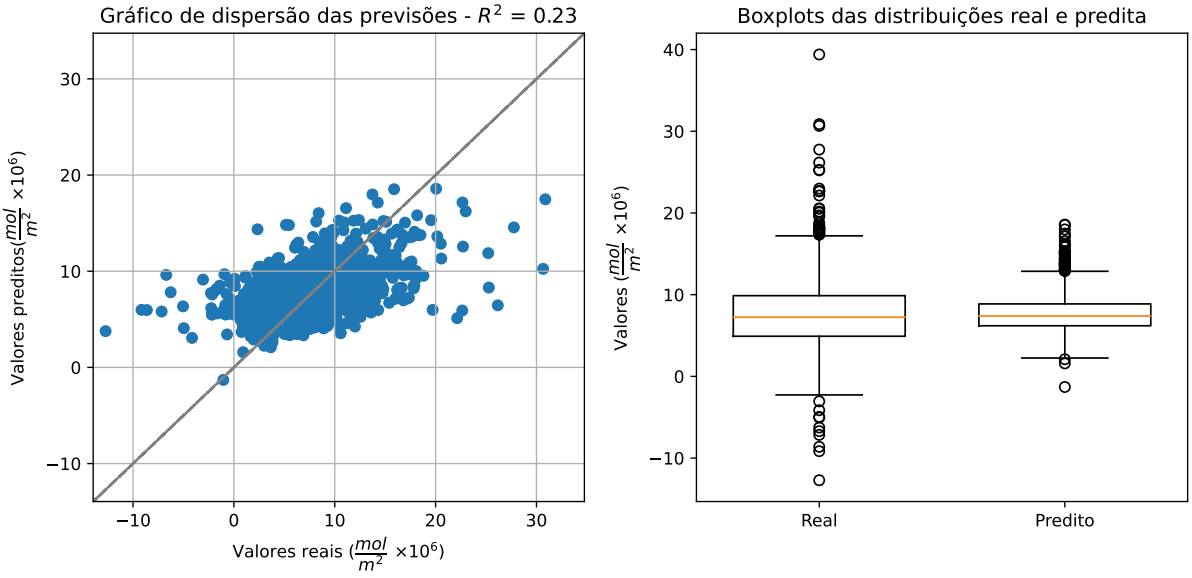
Tabela 11 – Métricas de avaliação de desempenho para o modelo Lasso

Métrica	**Valor**	**Unidade**
R^2	0.23	Adimensional
`r`	0.48	Adimensional
MSE	14.70	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.83	$\frac{mol}{m^2} \times 10^6$
MAE	2.84	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Adicionalmente, a Figura 16 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

Figura 16 – Dispersão entre a distribuição real e os valores preditos pelo modelo Lasso



Fonte: Produzido pelos autores

4.5.1.2 Random Forest

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 12. As métricas obtidas pelo modelo Random Forest mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 13.

Tabela 12 – Hiperparâmetros para o melhor modelo Random Forest após o grid search

Hiperparâmetro	**Valor**
`bootstrap`	`True`
`max_depth`	80
`max_features`	`'auto'`
`n_estimators`	200

Fonte: Produzido pelos autores.

Adicionalmente, a Figura 17 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

4.5.1.3 XGBoost

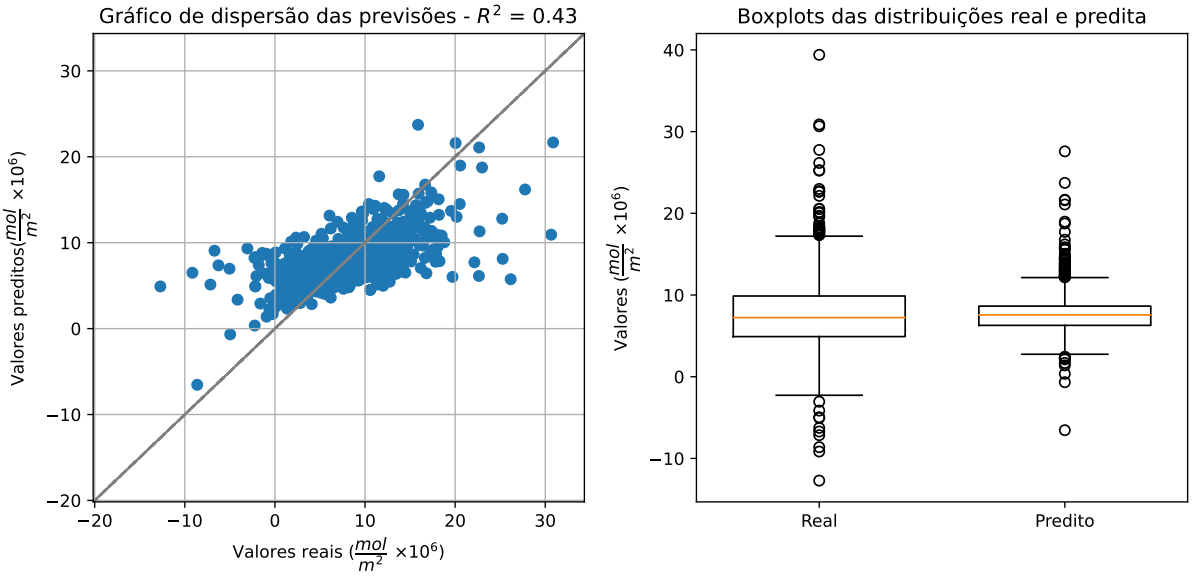
Os hiperparâmetros do melhor modelo estão fornecidos na tabela 14. As métricas obtidas pelo modelo XGBoost mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 15.

Tabela 13 – Resultados obtidos pelo melhor modelo de *Random Forest* ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.43	Adimensional
`r`	0.66	Adimensional
MSE	10.97	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.31	$\frac{mol}{m^2} \times 10^6$
MAE	2.30	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Figura 17 – Dispersão entre a distribuição real e os valores preditos pelo modelo Random Forest



Fonte: Produzido pelos autores

Adicionalmente, a Figura 18 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

4.5.1.4 Light Gradient Boosting Machine (LightGBM)

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 16. As métricas obtidas pelo modelo LightGBM mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 17.

Adicionalmente, a Figura 19 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

Tabela 14 – Hiperparâmetros para o melhor modelo XGBoost após o grid search

Hiperparâmetro	**Valor**
`colsample_bytree`	0.8
`eta`	0.1
`n_estimators`	400
`max_depth`	40
`subsample`	0.8

Fonte: Produzido pelos autores.

Tabela 15 – Resultados obtidos pelo melhor modelo de XGBoost ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.47	Adimensional
`r`	0.68	Adimensional
MSE	10.17	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.19	$\frac{mol}{m^2} \times 10^6$
MAE	2.00	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Tabela 16 – Hiperparâmetros para o melhor modelo LightGBM após o grid search

Hiperparâmetro	**Valor**
`boosting_type`	`gbdt`
`learning_rate`	0.05
`n_estimators`	1000
`num_leaves`	70

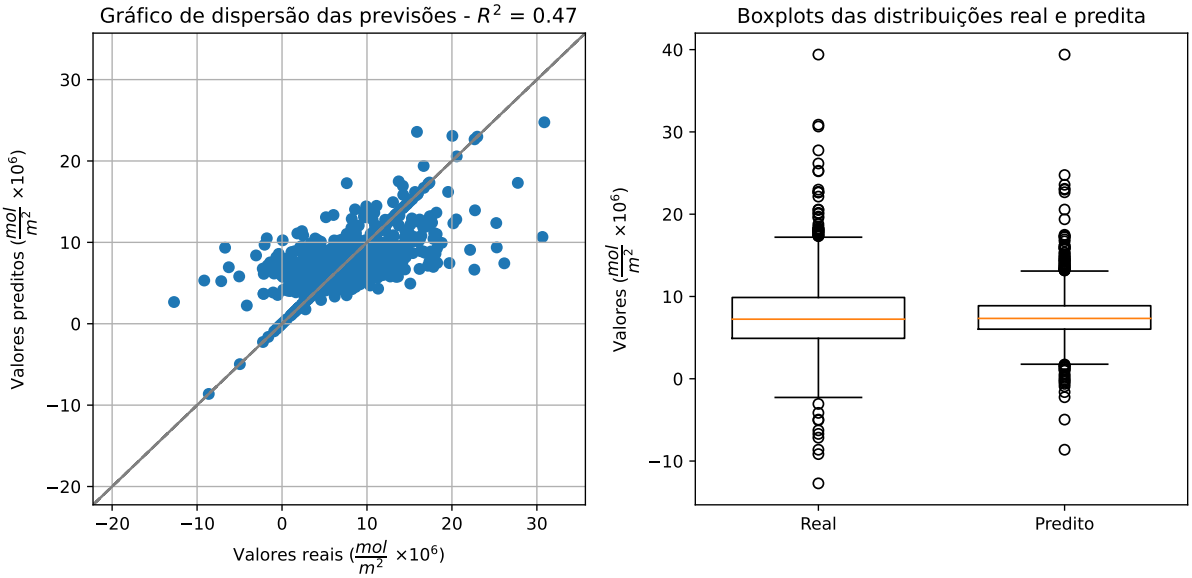
Fonte: Produzido pelos autores.

4.5.1.5 Convolução 1D (Conv1D)

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 18. As métricas obtidas pelo modelo de Conv1D mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 19.

Adicionalmente, a Figura 20 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

Figura 18 – Dispersão entre a distribuição real e os valores preditos pelo modelo XGBoost



Fonte: Produzido pelos autores

Tabela 17 – Resultados obtidos pelo melhor modelo de LightGBM ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.44	Adimensional
`r`	0.67	Adimensional
MSE	10.67	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.27	$\frac{mol}{m^2} \times 10^6$
MAE	2.21	$\frac{mol}{m^2} \times 10^6$

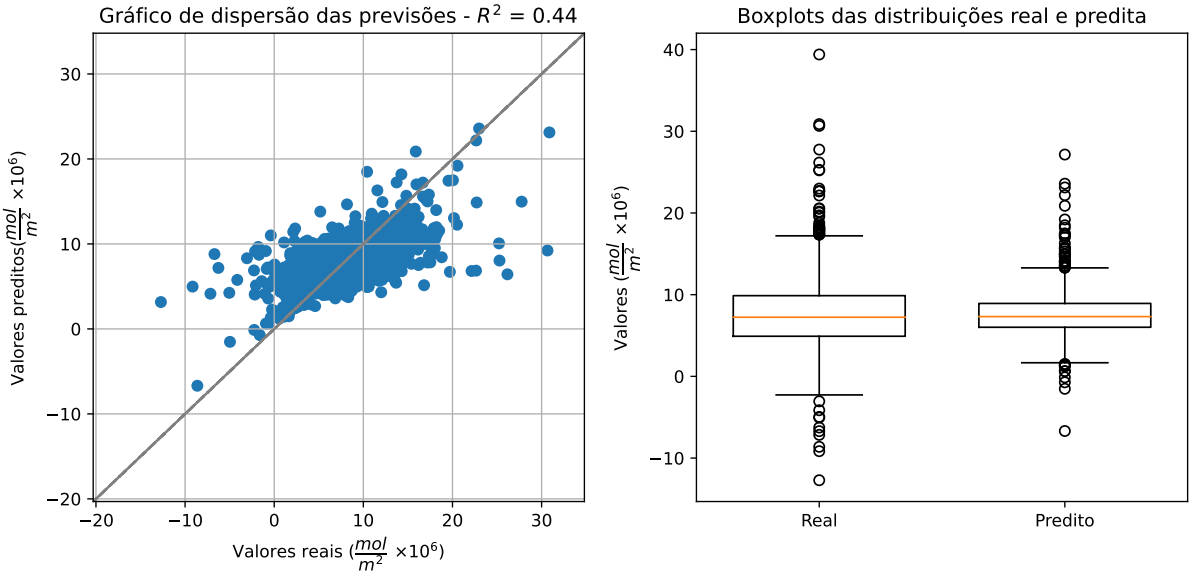
Fonte: Produzido pelos autores.

4.5.1.6 Gated Recurrent Unit (GRU)

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 20. As métricas obtidas pelo modelo GRU mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 21.

Adicionalmente, a Figura 21 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

Figura 19 – Dispersão entre a distribuição real e os valores preditos pelo modelo LightGBM



Fonte: Produzido pelos autores

Tabela 18 – Hiperparâmetros para o melhor modelo Conv1D após o grid search

Hiperparâmetro	Valor
`batch_size`	32
`convs`	`(8, 2, 1)`
`kernel_size`	23
`learning_rate`	0.001

Fonte: Produzido pelos autores.

4.5.1.7 Long Short-Term Memory (LSTM)

Os hiperparâmetros do melhor modelo estão fornecidos na tabela 22. As métricas obtidas pelo modelo LSTM mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 23.

Adicionalmente, a Figura 22 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

4.5.1.8 Convolutional Long Short-Term Memory (ConvLSTM)

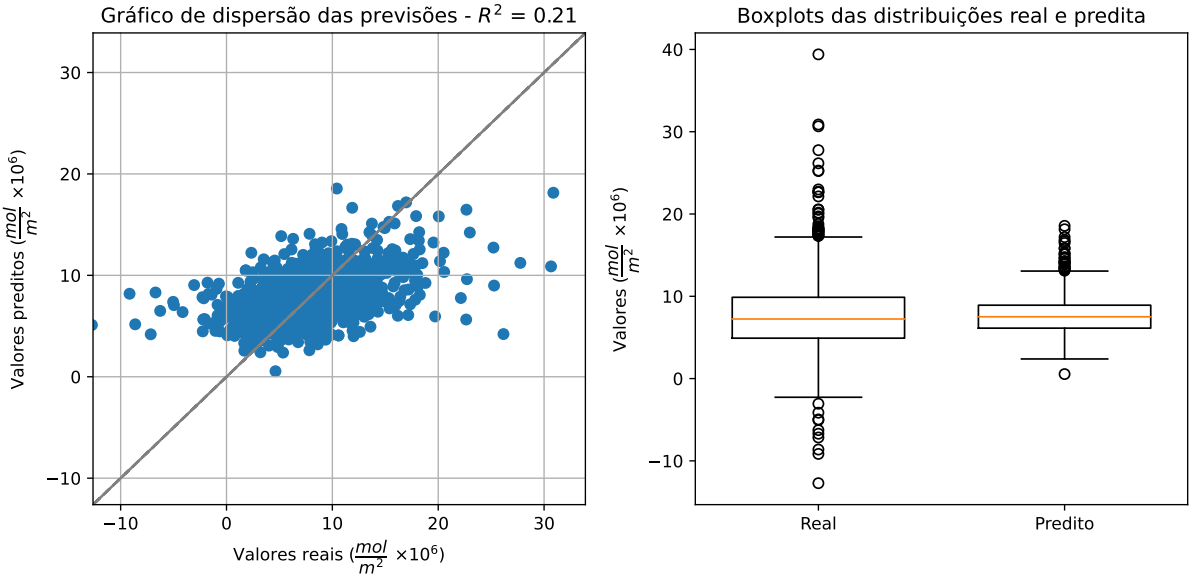
Os hiperparâmetros do melhor modelo estão fornecidos na tabela 24. As métricas obtidas pelo modelo ConvLSTM mais eficiente, após a execução da busca em grade (grid search), são exibidas na tabela 25.

Tabela 19 – Resultados obtidos pelo melhor modelo de Conv1D ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.21	Adimensional
`r`	0.46	Adimensional
MSE	15.16	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.89	$\frac{mol}{m^2} \times 10^6$
MAE	2.91	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Figura 20 – Dispersão entre a distribuição real e os valores preditos pelo modelo Conv1D



Fonte: Produzido pelos autores

Adicionalmente, a Figura 23 apresentam os gráficos de dispersão entre a distribuição real e os valores preditos.

4.5.2 Comparação entre os modelos

Nesta seção, procedemos à comparação e análise dos resultados alcançados por cada um dos modelos testados. O objetivo é identificar o modelo mais eficaz e compreender os motivos pelos quais ele pode ter apresentado um desempenho superior em relação aos demais. A Tabela 26 apresenta uma visão quantitativa comparativa dos resultados obtidos por cada modelo, destacando as principais métricas de avaliação.

Tabela 20 – Hiperparâmetros para o melhor modelo GRU após o grid search

Hiperparâmetro	**Valor**
`batch_size`	32
`hidden_dim`	32
`learning_rate`	0.01
`n_layers`	1

Fonte: Produzido pelos autores.

Tabela 21 – Resultados obtidos pelo melhor modelo de GRU ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.21	Adimensional
`r`	0.46	Adimensional
MSE	15.15	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.89	$\frac{mol}{m^2} \times 10^6$
MAE	2.88	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Tabela 22 – Hiperparâmetros para o melhor modelo LSTM após o grid search

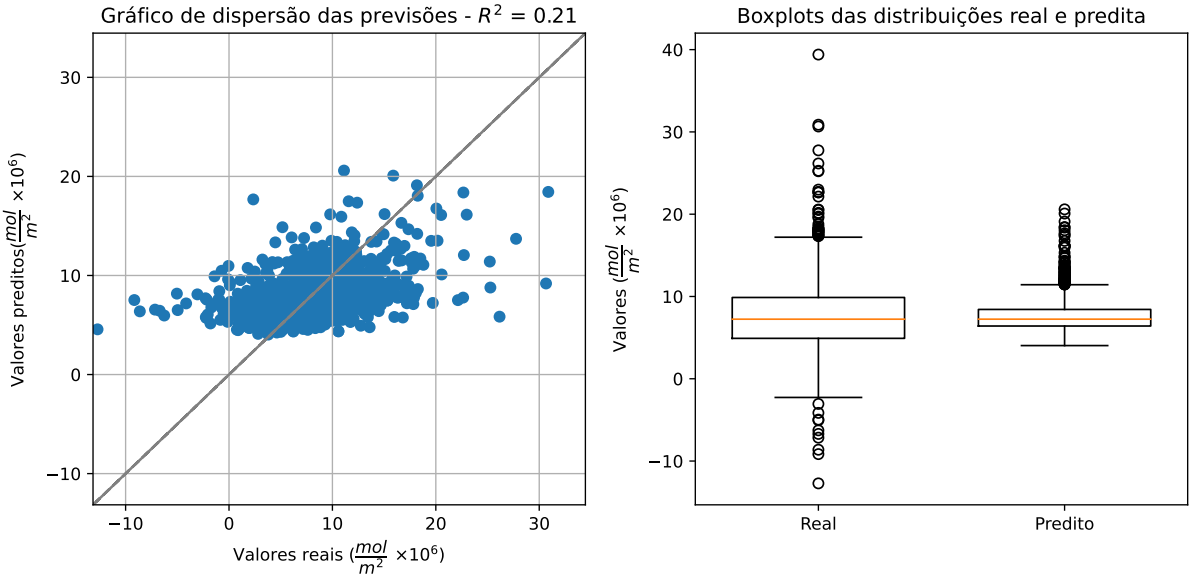
Hiperparâmetro	**Valor**
`batch_size`	128
`hidden_dim`	32
`learning_rate`	0.01
`n_layers`	1

Fonte: Produzido pelos autores.

Analisando a tabela 26, podemos observar que os modelos baseados em árvores de decisão, como Random Forest (RF), XGBoost e LightGBM, tiveram um desempenho superior em comparação com os modelos lineares e redes neurais, com base nas métricas R^2, MSE, RMSE e MAE. Especificamente, o XGBoost apresentou o melhor R^2 e as menores pontuações em MSE, RMSE e MAE, o que indica que ele foi capaz de capturar padrões mais complexos nos dados, talvez devido à sua habilidade em lidar com interações de características não lineares e sua robustez contra overfitting.

O Random Forest e o LightGBM também mostraram bons resultados, o que pode

Figura 21 – Dispersão entre a distribuição real e os valores preditos pelo modelo GRU



Fonte: Produzido pelos autores

Tabela 23 – Resultados obtidos pelo melhor modelo de LSTM ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.22	Adimensional
`r`	0.47	Adimensional
MSE	14.90	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.86	$\frac{mol}{m^2} \times 10^6$
MAE	2.86	$\frac{mol}{m^2} \times 10^6$
Desvio padrão do erro	3.85	$\frac{mol}{m^2} \times 10^6$

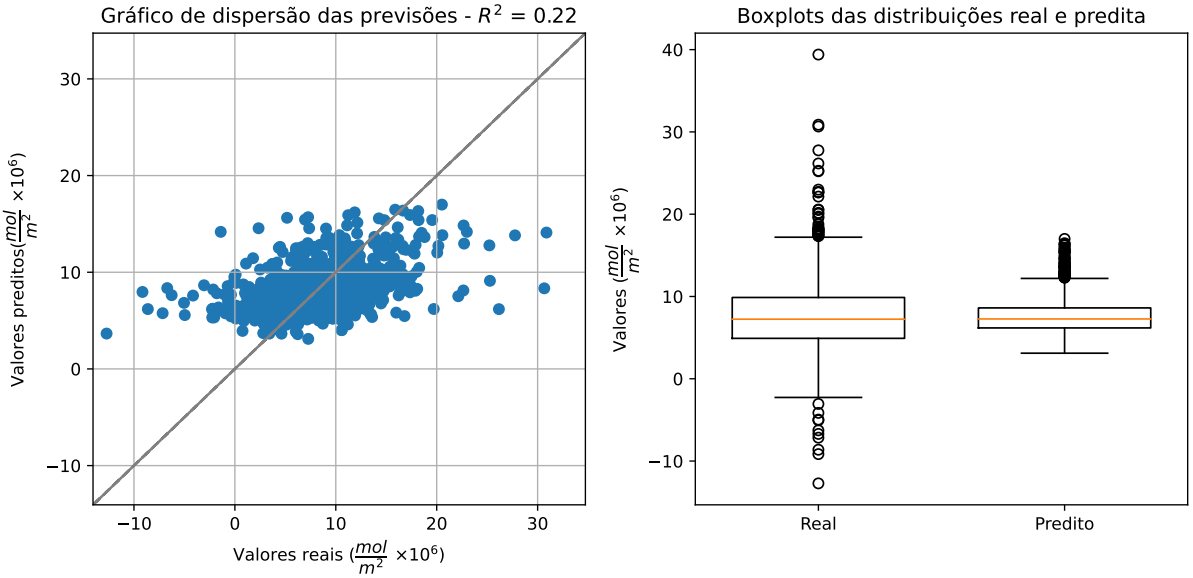
Fonte: Produzido pelos autores.

ser atribuído à sua eficiência em modelar relações não lineares sem exigir muita sintonia fina dos hiperparâmetros. Eles utilizam ensemble learning, que é a combinação de várias árvores de decisão, para produzir um resultado mais estável e confiável.

Por outro lado, o modelo Lasso, que é um modelo linear com regularização L1, teve o desempenho mais fraco. Isso pode ser devido à natureza dos dados que possuem relações complexas e não lineares que os modelos lineares não conseguem capturar.

Entre as redes neurais, as Conv1D, GRU, LSTM e ConvLSTM tiveram um desempenho inferior em comparação aos modelos baseados em árvores. Isso pode ser porque

Figura 22 – Dispersão entre a distribuição real e os valores preditos pelo modelo LSTM



Fonte: Produzido pelos autores

Tabela 24 – Hiperparâmetros para o melhor modelo ConvLSTM após o grid search

Hiperparâmetro	**Valor**
`batch_size`	32
`hidden_dim`	16
`learning_rate`	0.0001
`n_layers`	1

Fonte: Produzido pelos autores.

esses tipos de redes neurais são projetados para capturar dependências temporais ou sequenciais nos dados. Alguns fatores podem ter influenciado nesse resultado, como, por exemplo, a alta quantidade de valores faltantes nos dados de entrada e ao tamanho da série. Além disso, a imputação dos valores faltantes com a média da série, pode ter diminuído a complexidade dos dados, o que não favorece o treinamento efetivo dessas redes complexas. A ConvLSTM, sendo a mais complexa e projetada para capturar dependências espaciais e temporais, pode ter sofrido devido à falta de dados suficientes para revelar seu potencial ou devido ao overfitting.

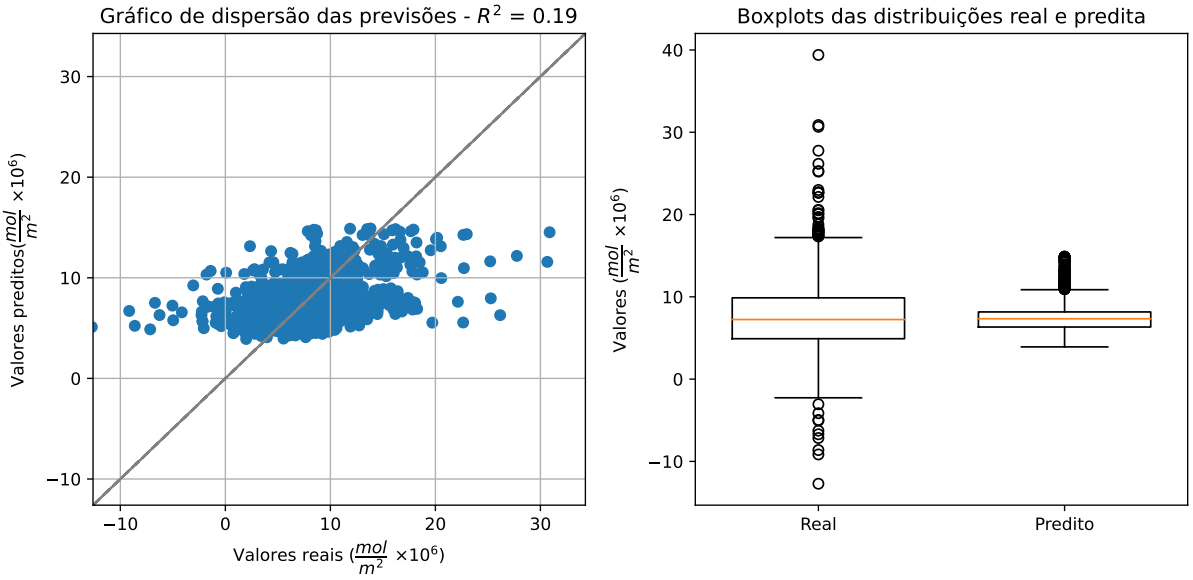
Os boxplots da Figura 24 fornecem uma visão da distribuição dos erros de todos os modelos estados. Nota-se que os modelos LightGBM, XGBoost e Random Forest exibem os menores erros absolutos, evidenciando um desempenho superior, com medianas inferiores e menor variabilidade nos erros.

Tabela 25 – Resultados obtidos pelo melhor modelo de ConvLSTM ao final do grid search.

Métrica	**Valor**	**Unidade**
R^2	0.19	Adimensional
`r`	0.43	Adimensional
MSE	15.53	$\left(\frac{mol}{m^2} \times 10^6\right)^2$
RMSE	3.94	$\frac{mol}{m^2} \times 10^6$
MAE	2.92	$\frac{mol}{m^2} \times 10^6$

Fonte: Produzido pelos autores.

Figura 23 – Dispersão entre a distribuição real e os valores preditos pelo modelo ConvLSTM



Fonte: Produzido pelos autores

Por outro lado, as redes neurais GRU, LSTM e ConvLSTM apresentam maiores erros absolutos e maior variabilidade, possivelmente necessitando de mais dados ou de um conjunto de dados mais complexos, isto é, com menos valores faltantes.

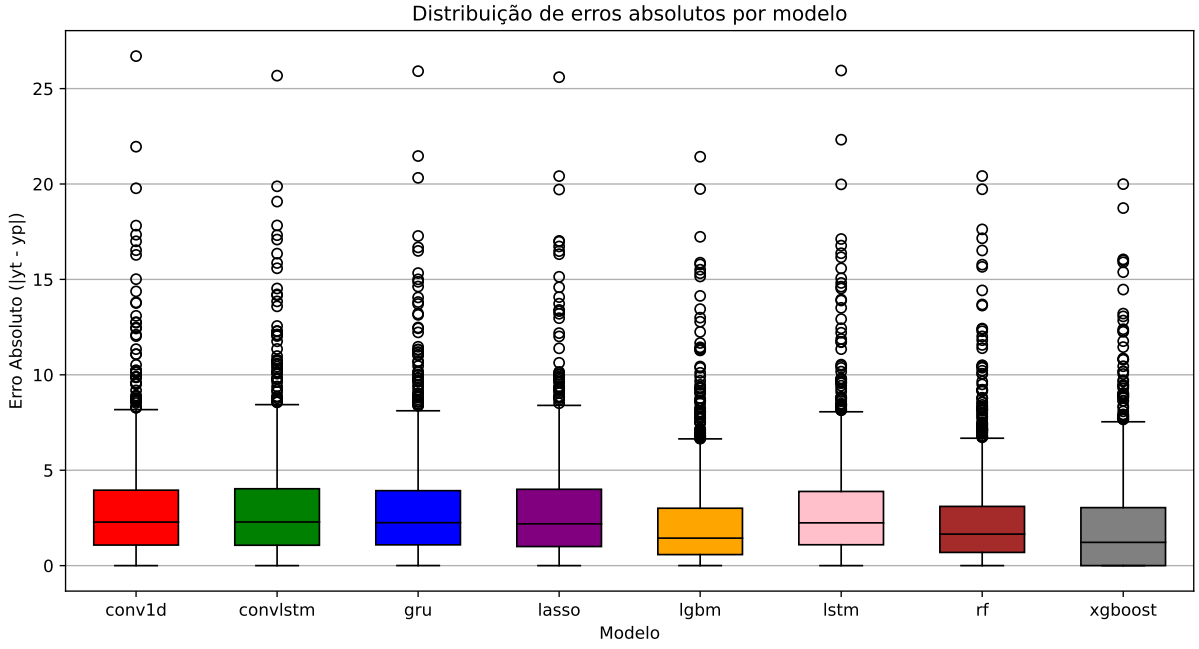
O modelo Lasso, apesar de ter um desempenho comparável ao das redes neurais em termos de erro absoluto, apresenta uma distribuição de erro ligeiramente mais concentrada. Outliers são notáveis em todos os modelos, especialmente nos modelos Random Forest, GRU, LSTM e ConvLSTM, sugerindo que, ocasionalmente, esses modelos produzem previsões com erros significativamente grandes. Em geral, os modelos baseados em árvores de decisão são os que apresentam melhor consistência e precisão para este conjunto

Tabela 26 – Resultados obtidos por todos os modelos testados.

Modelo	R^2	r	MSE	RMSE	MAE
Lasso	0.23	0.48	14.70	3.83	2.84
RF	0.43	0.66	10.97	3.31	2.30
XGBoost	**0.47**	**0.68**	**10.17**	**3.19**	**2.00**
LightGBM	0.44	0.66	10.66	3.26	2.20
Conv1D	0.21	0.46	15.16	3.89	2.90
GRU	0.21	0.45	15.15	3.89	2.88
LSTM	0.22	0.47	14.90	3.86	2.86
ConvLSTM	0.19	0.43	15.53	3.94	2.92

Fonte: Produzido pelos autores.

Figura 24 – Distribuição de erros absolutos por modelo, mostrando a comparação de desempenho entre LightGBM, XGBoost, Random Forest (RF), GRU, Lasso, ConvLSTM e LSTM

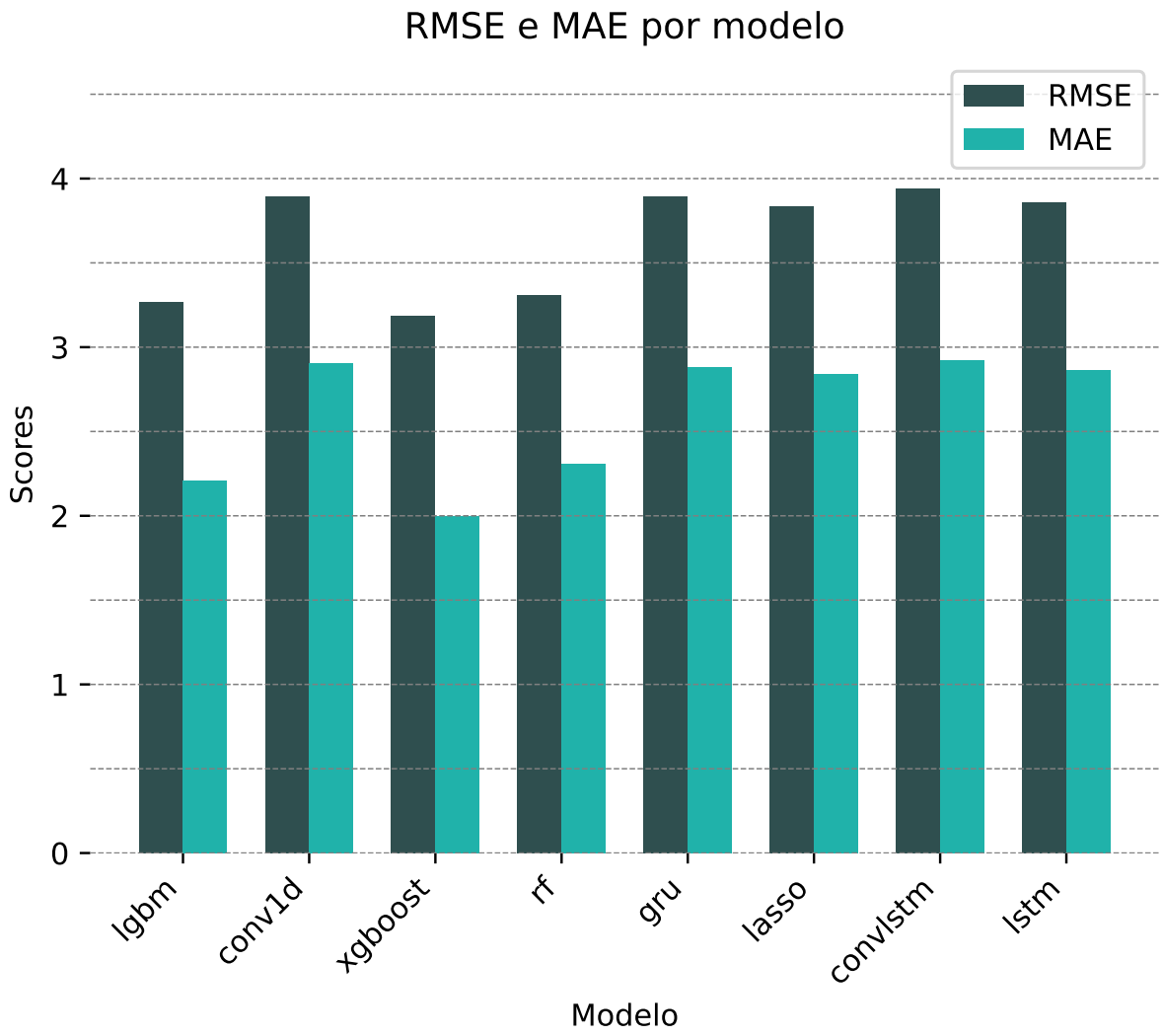


Fonte: Produzido pelos autores

específico de dados.

O gráfico 25 compara as métricas de erro RMSE e MAE entre os modelos. Observa-se que, para todos os modelos, os valores de RMSE são consistentemente mais altos do que os de MAE, o que é esperado dado que RMSE penaliza mais erros grandes devido ao quadrado na fórmula. Modelos como XGBoost, LightGBM e Random Forest (RF) têm pontuações relativamente mais baixas em ambas as métricas, o que sugere um melhor

Figura 25 – Comparação dos erros RMSE e MAE entre os modelos, incluindo LightGBM, XGBoost, Random Forest (RF), GRU, Lasso, ConvLSTM e LSTM



Fonte: Produzido pelos autores

desempenho na previsão da concentração de NO2. Isso pode indicar que esses modelos são mais eficazes em capturar as complexidades dos dados. Por outro lado, modelos como GRU, ConvLSTM e LSTM exibem as pontuações mais altas, implicando um desempenho menos preciso. Isso pode refletir uma menor adequação desses modelos de redes neurais para os dados em questão ou uma necessidade de mais ajustes nos hiperparâmetros e na arquitetura da rede.

4.6 Análise de Data Drift

O termo Data Drift refere-se a variações na distribuição dos dados ao longo do tempo. Essa ocorrência implica mudanças nas características estatísticas dos dados utilizados

para treinamento e validação de modelos preditivos. A análise de drift busca identificar e compreender essas mudanças, sendo importante para garantir a consistência e precisão contínua dos modelos em ambientes dinâmicos.

Com o intuito de avaliar a capacidade de generalização dos modelos baseado nos dados de entrada, foram conduzidas duas análises distintas de Data Drift: uma nas *features* de entrada dos modelos e outra no *target*, ou seja, nos dados de NO2. Para realizar essa análise, os dados no intervalo de 2019 a 2021 foram selecionados como o conjunto de referência. Adicionalmente, todas as amostras disponíveis para o ano de 2022 foram consideradas como "novas" instâncias, visando quantificar a magnitude do drift de um ano para outro.

Para realizar as análises, a biblioteca Evidently (AI, 2023) foi utilizada para conduzir análises sistemáticas de Data Drift, proporcionando métricas e visualizações para avaliação de dados ao longo do tempo. Assim, as métricas selecionadas para avaliar o drift serão descritas nas subseções a seguir.

4.6.1 Distância de Wasserstein

A distância de Wasserstein, também conhecida como distância de Earth Mover's (EMD), é uma métrica utilizada para medir a dissimilaridade entre duas distribuições de probabilidade. A fórmula para calcular a distância de Wasserstein é definida como:

$$W(p,q) = \inf_{\gamma \in \Pi(p,q)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\gamma(x,y) \tag{4.2}$$

Onde $W(p,q)$ é a distância de Wasserstein entre as distribuições p e q, γ é um plano de transporte que descreve como a massa em p deve ser movida para q, $\Pi(p,q)$ é o conjunto de todos os planos de transporte entre p e q e $c(x,y)$ é a função de custo associada ao transporte da massa do ponto x para o ponto y.

Essencialmente, a fórmula representa o custo mínimo necessário para transformar uma distribuição na outra, levando em consideração a função de custo associada ao transporte de massa. Na detecção de data drift, p e q representam as distribuições de dados em diferentes períodos, e a magnitude da distância de Wasserstein reflete a magnitude da mudança nas distribuições ao longo do tempo. Na condução das análises, determinou-se que um drift seria identificado quando o valor da distância fosse superior a 0.1.

4.6.2 Índice de Estabilidade da População (PSI)

O Índice de Estabilidade da População (PSI) é uma métrica utilizada para avaliar a estabilidade entre duas distribuições de dados ao longo do tempo. Esse método é comumente empregado para detectar deslocamentos ou mudanças nas características de

uma população ao comparar duas amostras distintas, geralmente representando diferentes períodos temporais.

A fórmula do PSI é dada por:

$$PSI = \sum_{i=1}^{n} \left((p_i - q_i) \cdot \ln \left(\frac{p_i}{q_i} \right) \right) \tag{4.3}$$

Onde: p_i é a proporção da população na i-ésima faixa (bin) na amostra de referência, q_i é a proporção da população na i-ésima faixa na nova amostra.

O PSI compara a distribuição acumulativa das duas amostras, calculando a diferença nas proporções de cada faixa e ponderando pelo logaritmo da razão dessas proporções. O resultado é uma métrica que indica a magnitude da mudança nas distribuições.

Em termos de detecção de drift em dados, um PSI significativamente alto sugere uma alteração substancial nas características da população entre as duas amostras, indicando a presença de drift. Na condução das análises, determinou-se que um drift seria identificado quando o valor do PSI fosse superior a 0.1.

4.6.3 Resultados

A tabela 27 descreve os resultados obtidos pela análise de drift para as *features* do modelo.

Tabela 27 – Resultados da análise de drift utilizando a Distância de Wasserstein e o Índice de Estabilidade da População

Variável	Distância de Wasserstein	PSI
`tropospheric_NO2`	**0.11**	0.03
`Column_WV`	0.06	0.02
`Optical_Depth_047`	0.08	0.03
`evaporation_from_bare_soil_sum`	0.05	0.00
`precipitationCal`	0.01	0.00
`sm_surface`	0.08	0.02
`surface_latent_heat_flux_sum`	0.02	0.00
`temperature_2m`	0.08	0.02
`volumetric_soil_water_layer_1`	0.04	0.01

Fonte: Produzido pelos autores.

Observa-se que a variável `tropospheric_NO2` apresenta a maior Distância de Wasserstein, o que indica um drift significativo em sua distribuição em comparação à distribuição de referência. Isso sugere mudanças no comportamento ou nas condições

associadas a essa variável específica, o que pode afetar a habilidade de treinar um modelo generalista ao longo do tempo.

Outras variáveis como Column_WV, Optical_Depth_047, sm_surface e temperature_2m exibem valores moderados de Distância de Wasserstein, o que indica uma mudança menos pronunciada, mas ainda assim notável em suas distribuições.

Quanto ao PSI, todos os valores apresentados são inferiores a 0.03, o que denota uma estabilidade geral nas distribuições das variáveis ao longo do tempo. Valores de PSI próximos a zero indicam que a probabilidade dessas variáveis permaneceram praticamente inalteradas é alta, sinalizando uma confiabilidade contínua no uso destas características pelo modelo.

Em resumo, a análise indica que, embora a maioria das variáveis tenha mantido uma estabilidade significativa, a variável tropospheric_NO2 mostra um drift que pode requerer investigação adicional para entender as causas e ajustar o modelo conforme necessário para manter sua precisão e eficácia.

4.7 Disponibilização dos resultados

Uma das etapas mais importantes de uma pesquisa científica é a divulgação dos resultados e disponibilização dos dados para a comunidade científica. Isso permite que outros pesquisadores validem, reproduzam ou construam sobre o trabalho realizado, ampliando o escopo do conhecimento científico. Neste contexto, os princípios FAIR desempenham um papel crucial para garantir que os dados e resultados sejam compartilhados de maneira eficaz e responsável.

Os princípios FAIR são um conjunto de diretrizes que visam melhorar a capacidade de gerenciamento, compartilhamento e reutilização de dados digitais. Cada letra do acrônimo FAIR representa um aspecto fundamental dessas diretrizes:

- **F de Findable (Achável)**: Os dados devem ser fáceis de encontrar para humanos e computadores. Isso é geralmente alcançado através do uso de identificadores únicos e de uma descrição rica em metadados, que facilitam a localização dos dados.
- **A de Accessible (Acessível)**: Uma vez encontrados, os dados precisam ser acessíveis. Isso não significa que os dados precisam ser sempre de acesso livre, mas que as condições de acesso devem ser claras, e o máximo de dados possível deve estar disponível após a pesquisa.
- **I de Interoperable (Interoperável)**: Os dados devem ser compatíveis com outros conjuntos de dados, plataformas, aplicativos e fluxos de trabalho para análise, arma-

zenamento e processamento. Isso é possível através do uso de formatos padronizados, vocabulários e infraestruturas.

- **R de Reusable (Reutilizável)**: Os dados devem ser reutilizáveis para que seu valor possa ser maximizado. Para isso, os dados devem ser adequadamente anotados, ter clara proveniência, e serem publicados sob termos de licença aberta quando possível.

Ao aderir a estes princípios, a divulgação dos resultados e a disponibilização dos dados do modelo para a comunidade científica não apenas seguem as melhores práticas de pesquisa, mas também potencializam o impacto e a utilidade dos achados da pesquisa. Isso permite que a comunidade científica, de forma colaborativa, avance na compreensão do tema estudado, promovendo inovações e descobertas significativas.

Nesse sentido, a divulgação do projeto de formatura no GitHub (GITHUB, 2023) é uma etapa importante, promovendo a transparência, a colaboração e a inovação. Este processo não apenas permite que outros estudantes e profissionais acessem, revisem e contribuam para o projeto, mas também serve como um valioso portfólio digital para o graduando, destacando suas habilidades técnicas e capacidade de resolver problemas complexos. Essa prática alinha-se com os princípios da ciência aberta e desenvolvimento colaborativo, essenciais no cenário científico e tecnológico atual. Assim, o repositório (WESLEY, 2023) contém todo o código-fonte utilizado ao longo do presente projeto.

4.7.1 DataMap

O Datamap (PLATFORM, 2023), uma plataforma de dados desenvolvida pelo grupo de Big Data da Poli-USP, tem como função visualizar, descobrir, catalogar e processar datasets e modelos de aprendizado de máquina, provendo compartilhamento, pesquisa e visualização rica para dados bioclimáticos. Essa plataforma facilita o compartilhamento, a pesquisa e a visualização de dados bioclimáticos, tornando-se uma ferramenta importante para o gerenciamento de informações complexas.

Durante a execução do projeto, a plataforma Datamap foi utilizada para disponibilizar os datasets empregados no treinamento dos modelos. Os datasets específicos do projeto estão acessíveis dentro da plataforma com a identificação `NO2_TROPOMI/XXX`, onde "`XXX`" representa cada uma das variáveis usadas no treinamento dos modelos. Além disso, os conjuntos de dados nomeados como `NO2_TROPOMI_IMPUTED/YYY` correspondem àqueles que foram imputados pelo modelo desenvolvido ao longo deste projeto, demonstrando a aplicabilidade prática e a integração do modelo com as ferramentas de gestão de dados.

5 Considerações Finais

5.1 Conclusões do Projeto de Formatura e Contribuições

O projeto conseguiu evidenciar que a abordagem adotada para a estimativa da concentração da coluna troposférica de NO2 é promissora. Mesmo dentro do escopo limitado definido, os resultados alcançados foram significativos, destacando-se principalmente o desempenho dos modelos baseados em árvore e boosting. Essa constatação reforça a eficácia da metodologia escolhida, especialmente no que tange à aplicação de modelos de machine learning para análises ambientais. Os resultados obtidos com os modelos de deep learning, apesar de não serem tão promissores, fornecem uma base sólida para futuras investigações e melhorias.

Além disso, a importância desses dados vai além do campo técnico e científico, estendendo-se à sociedade e à indústria em geral. A capacidade de medir com precisão as concentrações de NO2 tem implicações significativas para a compreensão e mitigação das mudanças climáticas. Estes dados são essenciais para apoiar estudos sobre mudanças climáticas, contribuindo para a modelagem de previsões climáticas e o desenvolvimento de estratégias para mitigar seus efeitos adversos. Também são cruciais para compreender o balanço de carbono do planeta, monitorar a qualidade do ar e entender melhor o ciclo de carbono.

A saúde pública também se beneficia desses dados, já que a alta concentração de NO2 está associada a problemas de saúde como doenças respiratórias. Monitorar e analisar essas concentrações pode conduzir a políticas de saúde pública mais eficazes e melhorar a qualidade de vida. Para a indústria, entender essas concentrações é vital para garantir que as operações estejam em conformidade com as regulamentações ambientais e pode impulsionar o desenvolvimento de tecnologias mais limpas e sustentáveis.

Além disso, esses dados podem ser usados por governos e organizações internacionais para formular políticas ambientais mais eficientes, visando a redução de emissões e a melhoria da qualidade do ar. A divulgação dessas informações também aumenta a conscientização sobre as questões ambientais, promovendo a educação e a participação da sociedade em esforços de proteção ambiental.

Portanto, ao considerar a relevância desses dados para a sociedade e a indústria, o projeto não apenas destaca a importância técnica e científica da metodologia adotada, mas também sublinha a importância prática e social de tais estudos. Isso enfatiza a necessidade contínua de investir em pesquisas e tecnologias que possam contribuir para um futuro mais sustentável e saudável.

5.2 Perspectivas de Continuidade

Para a continuidade deste projeto, recomenda-se a ampliação do conjunto de dados e a melhoria das técnicas de processamento de dados para aprimorar o desempenho dos modelos de deep learning. Investigar a aplicabilidade dos modelos em conjuntos de dados mais amplos ou sob diferentes condições ambientais também é uma direção promissora, pois ajudaria a validar a generalização dos modelos desenvolvidos. Adicionalmente, a integração desses modelos com sistemas de monitoramento em tempo real poderia ser uma contribuição valiosa para estratégias mais efetivas de gestão ambiental, utilizando as metodologias desenvolvidas para fornecer insights mais precisos e em tempo hábil.

Referências

AI, E. *Evidently AI*. 2023. <https://www.evidentlyai.com/>. Accessed: 2023-09-20. Citado na página 70.

ARAGÃO, L. E. O. C. et al. 21st century drought-related fires counteract the decline of amazon deforestation carbon emissions. *Nature Communications*, v. 9, n. 1, p. 536, Feb 2018. ISSN 2041-1723. Disponível em: <https://doi.org/10.1038/s41467-017-02771-y>. Citado 2 vezes nas páginas 16 e 17.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>. Citado na página 23.

CHAN, K. L. et al. Estimation of surface no2 concentrations over germany from tropomi satellite observations using a machine learning method. *Remote Sensing*, v. 13, n. 5, 2021. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/13/5/969>. Citado na página 31.

CHANG, K. *Introduction to Geographic Information Systems*. McGraw-Hill, 2001. ISBN 9780072382693. Disponível em: <https://books.google.com.br/books?id=fRwvt0gZv5QC>. Citado na página 21.

CHE, Z. et al. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, v. 8, n. 1, p. 6085, Apr 2018. ISSN 2045-2322. Disponível em: <https://doi.org/10.1038/s41598-018-24271-9>. Citado na página 51.

CHEN, B. et al. A joint learning im-bilstm model for incomplete time-series sentinel-2a data imputation and crop classification. *International Journal of Applied Earth Observation and Geoinformation*, v. 108, p. 102762, 2022. ISSN 1569-8432. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0303243422000885>. Citado na página 29.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016. (KDD '16). Disponível em: <http://dx.doi.org/10.1145/2939672.2939785>. Citado na página 24.

CHO, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. Disponível em: <http://arxiv.org/abs/1406.1078>. Citado na página 25.

CORPORATION, M. *LightGBM*. 2023. <https://lightgbm.readthedocs.io/en/latest/index.html>. Accessed: 2023-09-19. Citado na página 53.

CORRêA, P. L. P. *Conceitos de Ciência dos Dados e Big Data - Experimentos*. [S.l.]: Project Shell/FAPESP/RCGI, 2022. Accessed: 2023-05-14. Citado na página 32.

CRUTZEN, P. J. The role of no and no2 in the chemistry of the troposphere and stratosphere. *Annual Review of Earth and Planetary Sciences*, v. 7, n. 1, p. 443–472, 1979.

Disponível em: <https://doi.org/10.1146/annurev.ea.07.050179.002303>. Citado na página 17.

DAVIDSON, E. A. et al. The amazon basin in transition. *Nature*, v. 481, n. 7381, p. 321–328, Jan 2012. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/nature10717>. Citado na página 18.

DEKHTYAR, A. *Lecture Notes on Data Science - DATA 301*. 2016. <https://users.csc.calpoly.edu/~dekhtyar/>. Accessed: 2023-05-14. Citado na página 32.

DEVELOPERS xgboost. *XGBoost*. 2023. <https://xgboost.readthedocs.io/en/stable/index.html>. Accessed: 2023-09-19. Citado na página 54.

ESA Tropomi. 2023. <https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-5P/Tropomi>. Accessed: 2023-05-14. Citado na página 22.

FOUNDATION, P. S. *Python*. 2023. <https://www.python.org/>. Accessed: 2023-09-14. Citado na página 38.

GHAHREMANLOO, M. et al. Deep learning estimation of daily ground-level no2 concentrations from remote sensing data. *Journal of Geophysical Research: Atmospheres*, Wiley Online Library, v. 126, n. 21, p. e2021JD034925, 2021. Citado na página 18.

GITHUB, I. *GitHub*. 2023. <https://github.com/>. Accessed: 2023-12-01. Citado na página 73.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Disponível em: <https://books.google.com.br/books?id=Np9SDQAAQBAJ>. Citado na página 49.

GOOGLE. *Google Earth Engine API*. 2023. <https://developers.google.com/earth-engine/apidocs>. Accessed: 2023-09-14. Citado na página 38.

GOSWAMI, S.; SANGEETA, K. Anomalies in landsat imagery and imputation. In: *Proceedings of the Third International Symposium on Women in Computing and Informatics*. New York, NY, USA: Association for Computing Machinery, 2015. (WCI '15), p. 353–358. ISBN 9781450333610. Disponível em: <https://doi.org/10.1145/2791405.2791495>. Citado na página 29.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001. (Springer series in statistics). ISBN 9780387952840. Disponível em: <https://books.google.com.br/books?id=VRzITwgNV2UC>. Citado na página 49.

JENSEN, J. *Remote Sensing of the Environment: An Earth Resource Perspective 2/e*. Pearson Education, 2009. ISBN 9788131716809. Disponível em: <https://books.google.com.br/books?id=ge_nwDX-HBEC>. Citado na página 20.

KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>. Citado na página 24.

LILLESAND, T.; KIEFER, R.; CHIPMAN, J. *Remote Sensing and Image Interpretation, 7th Edition*. Wiley, 2015. ISBN 9781118919453. Disponível em: <https://books.google.com.br/books?id=eQXYBgAAQBAJ>. Citado na página 20.

LONG, S. et al. Estimating daily ground-level no2 concentrations over china based on tropomi observations and machine learning approach. *Atmospheric Environment*, v. 289, p. 119310, 2022. ISSN 1352-2310. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1352231022003752>. Citado na página 31.

LOPS, Y. et al. Spatiotemporal estimation of tropomi no2 column with depthwise partial convolutional neural network. *Neural Computing and Applications*, v. 35, n. 21, p. 15667–15678, Jul 2023. ISSN 1433-3058. Disponível em: <https://doi.org/10.1007/s00521-023-08558-1>. Citado 2 vezes nas páginas 29 e 30.

MALHI, Y. et al. Climate change, deforestation, and the fate of the amazon. *Science*, v. 319, n. 5860, p. 169–172, 2008. Disponível em: <https://www.science.org/doi/abs/10.1126/science.1146961>. Citado 2 vezes nas páginas 16 e 17.

MOSKOLAï, W. R. et al. Application of deep learning architectures for satellite image time series prediction: A review. *Remote Sensing*, v. 13, n. 23, 2021. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/13/23/4822>. Citado na página 51.

MURPHY, K. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2021. ISBN 9780262044660. Disponível em: <https://books.google.com.br/books?id=dAhkzQEACAAJ>. Citado na página 22.

NOBRE, C. A. et al. Land-use and climate change risks in the amazon and the need of a novel sustainable development paradigm. *Proceedings of the National Academy of Sciences*, v. 113, n. 39, p. 10759–10768, 2016. Disponível em: <https://www.pnas.org/doi/abs/10.1073/pnas.1605516113>. Citado na página 18.

NUSSBAUMER, H. *Fast Fourier Transform and Convolution Algorithms*. Springer Berlin Heidelberg, 2013. (Springer Series in Information Sciences). ISBN 9783662005514. Disponível em: <https://books.google.com.br/books?id=tnjpCAAAQBAJ>. Citado na página 24.

Pearson, K. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London Series I*, v. 58, p. 240–242, jan. 1895. Citado na página 28.

PLATFORM, . D. M. *DataMap - Scientific data analysis, for everyone.* 2023. <https://datamap-webapp.vercel.app/>. Accessed: 2023-11-19. Citado na página 73.

SCHAPIRE, R.; FREUND, Y. *Boosting: Foundations and Algorithms*. MIT Press, 2014. (Adaptive Computation and Machine Learning series). ISBN 9780262526036. Disponível em: <https://books.google.com.br/books?id=lL34DwAAQBAJ>. Citado na página 24.

SHAW, G. A.; BURKE, H. hua K. Spectral imaging for remote sensing. In: . [s.n.], 2003. Disponível em: <https://api.semanticscholar.org/CorpusID:360258>. Citado na página 21.

SHI, X. et al. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: CORTES, C. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015. v. 28. Disponível em: <https://proceedings.neurips.

cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf>. Citado na página 26.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, v. 58, p. 267–288, 1996. Citado na página 23.

WANG, S. et al. Sta-gan: A spatio-temporal attention generative adversarial network for missing value imputation in satellite data. *Remote Sensing*, v. 15, n. 1, 2023. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/15/1/88>. Citado na página 29.

WANG, W. et al. A machine learning model to estimate ground-level ozone concentrations in california using tropomi data and high-resolution meteorology. *Environment International*, v. 158, p. 106917, 2022. ISSN 0160-4120. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0160412021005420>. Citado na página 31.

WANG, Y.; FALOONA, I. C.; HOULTON, B. Z. Satellite no2 trends reveal pervasive impacts of wildfire and soil emissions across california landscapes. *Environmental Research Letters*, IOP Publishing, v. 18, n. 9, p. 094032, aug 2023. Disponível em: <https://dx.doi.org/10.1088/1748-9326/acec5f>. Citado na página 16.

WESLEY, P. A. *Imputação de Dados de NO2: Repositório de Código*. 2023. <https://github.com/WesPereira/no2-data-imputation/>. Accessed: 2023-12-12. Citado na página 73.

YANG, X.; ZHAO, Y.; VATSAVAI, R. R. Deep residual network with multi-image attention for imputing under clouds in satellite imagery. In: *2022 26th International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2022. p. 643–649. Citado na página 29.

ZENG, J.; MATSUNAGA, T.; SHIRAI, T. A new estimate of oceanic co_2 fluxes by machine learning reveals the impact of co_2 trends in different methods. *Earth System Science Data Discussions*, v. 2022, p. 1–27, 2022. Disponível em: <https://essd.copernicus.org/preprints/essd-2022-71/>. Citado na página 18.