UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Faculdade de Engenharia e Ciências - Campus de Guaratinguetá

GIANLUCA LIMA PERINI

APRIMORAMENTO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE INTERNAÇÕES POR DOENÇAS RESPIRATÓRIAS EM CUIABÁ-MT

Guaratinguetá

2025

GIANLUCA LIMA PERINI

APRIMORAMENTO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE INTERNAÇÕES POR DOENÇAS RESPIRATÓRIAS EM CUIABÁ-MT

Trabalho de Conclusão de Curso apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia e Ciências, Guaratinguetá, para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador(a): Prof. Dra. Paloma Maria Silva Rocha Rizol

Guaratinguetá

2025

Perini, Gianluca Lima
P445a Aprimoramento de modelos de aprendizado de máquina para predição de internações por doenças respiratórias em Cuiabá-MT / Gianluca Lima Perini - Guaratinguetá, 2025.
58 f : il.
Bibliografia: f. 55-57

Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2025.

Orientadora: Profª. Drª Paloma Maria Silva Rocha Rizol

1. Aprendizado do computador. 2. Aparelho respiratório - Doenças. 3. Saúde pública. 4. Redes neurais (Computação). I. Título.

CDU 007.52

Luciana Máximo
Bibliotecária/CRB-8 3595

GIANLUCA LIMA PERINI

APRIMORAMENTO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE INTERNAÇÕES POR DOENÇAS RESPIRATÓRIAS EM CUIABÁ-MT

Trabalho de Conclusão de Curso apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia e Ciências, Guaratinguetá, para obtenção do título de Bacharel em Engenharia Elétrica.

Data da defesa: 14 / 11 / 2025

Banca Examinadora:

gov.br Documento assinado digitalmente
PALOMA MARIA SILVA ROCHA RIZOL
Data: 15/11/2025 17:08:01-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Dr. Paloma Maria Silva Rocha Rizol
UNESP - Faculdade de Engenharia e Ciências - Campus de Guaratinguetá

gov.br Documento assinado digitalmente
ALEX PISCIOTTA
Data: 17/11/2025 11:46:13-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Me. Alex Pisciotta
UNESP - Faculdade de Engenharia e Ciências - Campus de Guaratinguetá

gov.br Documento assinado digitalmente
EZEQUIAS COSTA ROCHA
Data: 15/11/2025 18:31:43-0300
Verifique em https://validar.iti.gov.br

_____

Prof. Eng. Ezequias Costa Rocha
UNESP - Faculdade de Engenharia e Ciências - Campus de Guaratinguetá

Dedico este trabalho àqueles que lutam por um mundo socialmente igualitário e pela democratização do acesso ao conhecimento.

AGRADECIMENTOS

Aos meus pais, Marli e Nadir, por todo suporte, educação, consciência, carinho e amor ao longo de toda minha trajetória.

À Sthefanie, minha companheira, que foi a luz nos momentos de breu e sustentação nas passagens mais turbulentas.

À toda minha família pelo suporte em toda eventualidade.

Ao meu irmão e ala, Rafael V. Gomes, por sempre me ajudar nos arremates, e a realizar uma decolagem tranquila e um pouso suave.

À Professora Paloma, que me confiou a realização desse projeto.

Ao meu irmão de curso Leonardo M. Nunes, que foi grande parceria na realização de todas as matérias do curso.

Ao meu primeiro tutor, Rodrigo Lima, que direcionou meu engatinhar no mundo da tecnologia.

Aos amigos João V. M. Silva, Ana Luísa Gaioli, Lucas M. Pereira, Yago Campos, Leonardo Oliveira por todas as horas estudadas juntos e os momentos de resiliência e aprendizado.

Aos irmãos da República 333, pelos 4 anos de risadas e bons momentos.

Ao João V. Zangrandi, Pedro H. Pedran, Francisco A. Gonçalves e Rafael, pelos bons momentos vividos na Alemanha.

Aos colegas da FEG Robótica, pelos dois anos de excelentes experiências.

À toda equipe da FEG, tanto funcionários quanto corpo docente.

À todos os professores que fizeram parte da minha trajetória até aqui, colaborando para construção de uma base sólida de conhecimento e desenvolvimento de senso crítico.

Ao apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

“Não podemos esperar construir um mundo melhor sem melhorar os indivíduos”
(Marie Curie).

RESUMO

Este trabalho tem como objetivo aprimorar modelos de aprendizado de máquina para previsão de internações por doenças respiratórias na cidade de Cuiabá-MT, originalmente desenvolvidos por Matheus Cerqueira (2023). Utilizaram-se dados do Sistema de Informação Hospitalar (SIHSUS), Sistema de Informação de Saúde Ambiental (SISAM), da mesma forma como executado pelo Cerqueira (2023) em seu trabalho, com a adição da base do Instituto Nacional de Meteorologia (INMET), referentes ao período de 2013 a 2018. As bases foram tratadas, integradas e analisadas com a linguagem Python e as bibliotecas Pandas, Scikit-learn e XGBoost. Foram avaliados os modelos de Regressão Linear, Árvore de Decisão, Floresta Aleatória, XGBoost e Máquinas de Vetores de Suporte (SVR) segundo as métricas MAE, RMSE e MAPE. O XGBoost apresentou o melhor desempenho, com redução significativa dos erros em relação ao trabalho anterior. Conclui-se que o uso de técnicas de aprendizado de máquina mais robustas e o ajuste de hiperparâmetros aumentam a acurácia das previsões, reforçando o potencial da inteligência artificial como instrumento de apoio à formulação de políticas públicas em saúde.

Palavras-chave: doenças respiratórias; poluentes do ar; aprendizado de máquina; saúde pública; xgboost; regressão linear; árvore de decisão; floresta aleatória; máquinas de vetores de suporte.

ABSTRACT

This study aims to improve machine learning models for predicting hospitalizations due to respiratory diseases in the city of Cuiabá, Mato Grosso, originally developed by Matheus Cerqueira (2023). Data from the Hospital Information System (SIHSUS) and the Environmental Health Information System (SISAM) were used, as in Cerqueira's (2023) study, with the addition of data from the National Institute of Meteorology (INMET) for the period from 2013 to 2018. The databases were processed, integrated, and analyzed using the Python and Pandas, Scikit-learn, and XGBoost libraries. Linear Regression, Decision Tree, Random Forest, XGBoost, and Support Vector Machines (SVR) models were evaluated according to the MAE, RMSE, and MAPE metrics. XGBoost performed best, with a significant reduction in errors compared to the previous study. It was concluded that the use of more robust machine learning techniques and hyperparameter tuning increase the accuracy of predictions, reinforcing the potential of artificial intelligence as a tool to support the formulation of public health policies.

Keywords: respiratory diseases; air pollutants; machine learning; public health; xgboost; linear regression; decision tree; random forest; support vector machines.

LISTA DE FIGURAS

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CSV	Comma Separated Values
DBC	DataBase Compact
DBF	DataBase File
IBGE	Instituto Brasileiro de Geografia e Estatística
INMET	Instituto Nacional de Meteorologia
SISAM	Sistema Integrado de Serviços Ambientais
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentual Error

SUMÁRIO

1 INTRODUÇÃO

O mundo atual vive uma crise ligada ao meio ambiente e à sociedade, onde o planeta e as pessoas são cada vez mais afetados. Climas estáveis e bem definidos, bem como ar limpo, essenciais para existência de toda forma de vida, sofrem forte pressão por causa das ações humanas, criando um processo cíclico de degradação da natureza e piora nas condições de saúde. Diante disso, três movimentos globais se juntam com risco crescente: derrubada intensa de florestas, maior poluição atmosférica e, como consequência, alta nos óbitos provocados por problemas pulmonares.

A poluição atmosférica é uma das ameaças mais sérias ao bem-estar da população, aparecendo entre os motivos centrais de óbitos precoces pelo planeta. Dados científicos comprovam claramente que a exposição a substâncias tóxicas no ambiente - seja por pouco ou muito tempo - aumenta consideravelmente problemas de saúde e óbitos. Os materiais particulados finos (PM2.5), sobretudo, causam mais estragos porque entram fundo nos pulmões e se espalham pelo sangue, provocando e agravando doenças crônicas (ORGANIZAÇÃO MUNDIAL DA SAÚDE, 2021). Pesquisas abrangentes, incluindo o estudo Global Burden of Disease (GBD), apontam que a má qualidade do ar mata milhões por ano, batendo indicadores de risco já reconhecidos. Tais fatalidades ligam-se especialmente a complicações no coração, como derrames e infartos, além da Doença Pulmonar Obstrutiva Crônica (DPOC) e tumores nos pulmões (COHEN et al., 2017; VOHRA et al., 2021).

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Este trabalho segue a mesma linha do trabalho do Matheus Cerqueira (2023): fazer uso de técnicas de aprendizado de máquina e redes neurais artificiais na predição do número de internações hospitalares de pacientes que foram atingidos por doenças respiratórias, estas causadas por poluição atmosférica. Utilizando bases de dados públicas: INMET, SISAM e DATASUS.

1.1.2 Objetivo específico

Em seu trabalho, Matheus Cerqueira (2023) propõe melhorias para seus modelos, como:

1. Aumentar a base de dados;
2. Analisar a influência das *features* não utilizadas, variando-as na base de dados;

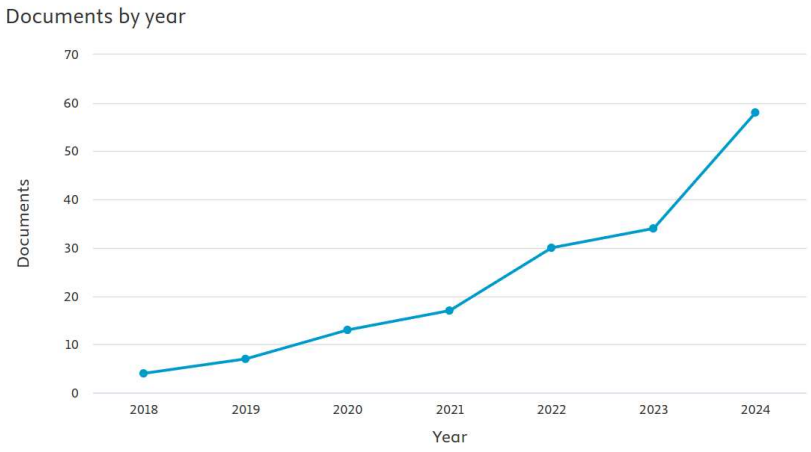
3. Analisar a influência da utilização de diferentes *lags*, variando entre 0 a 7;
4. Realizar *Hyperparameter boost* da *Random Forest*;
5. Utilizar outros modelos de *machine learning* como *XGboost* e *SVR*;
6. Aumentar a complexidade do modelo de LSTM.

Destas proposições, será analisada a implementação neste trabalho dos itens 1, 2, 4 e 5.

1.2 JUSTIFICATIVAS

Utilizando a plataforma Scopus para pesquisar artigos por título, palavra-chave e resumo, para verificar a existência de trabalhos semelhantes a este, números estes apresentados na Figura 1 abaixo. As palavras chaves utilizadas foram *machine learning*, *respiratory diseases* e *air quality*, com o conectivo *AND,* considerando o período de 2012 até 2024. Nesse contexto, utilizando as três palavras-chaves foram encontrados apenas 163 trabalhos. Ao adicionar a palavra "Cuiabá", o sistema não retorna nenhum resultado.

Figura 1 - Publicações de pesquisas por ano

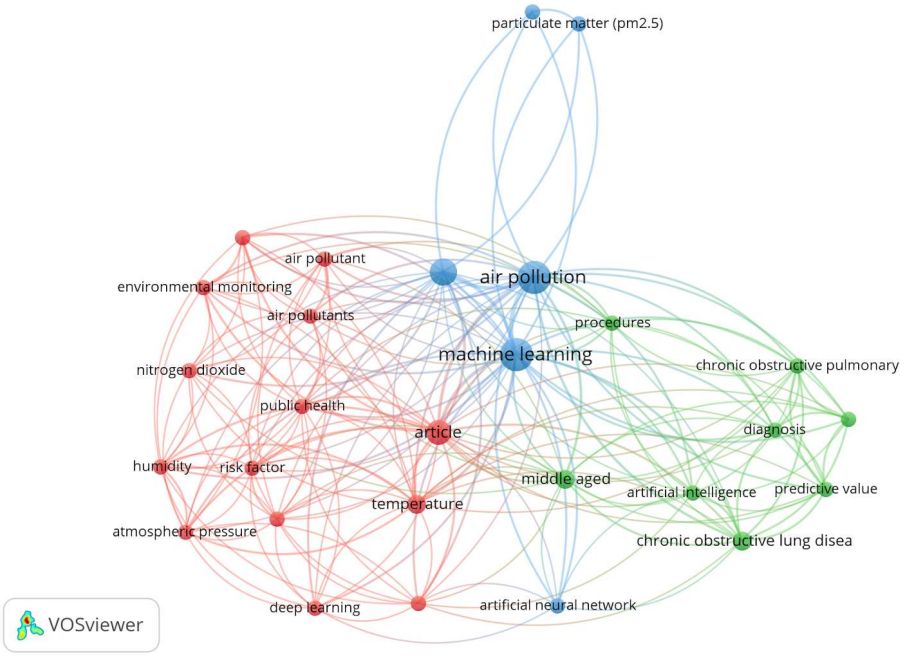


Fonte: SCOPUS (2025)

A baixa quantidade de trabalhos publicados é um indicativo de que essa área precisa ser mais estudada e aperfeiçoada.

A Figura 2 foi gerada no software VOSViewer a partir dos dados bibliométricos exportados da pesquisa realizada no SCOPUS. Nela é apresentada uma nuvem de palavras onde se nota a correlação entre algumas palavras-chaves que são parte integrante e fundamental deste trabalho de graduação.

Figura 2 - Relação entre palavras-chave deste trabalho



Fonte: VOSViewer (2025)

Analisando a imagem, percebe-se a forte conexão entre *machine learning* e *air pollution*.

1.3 DELIMITAÇÕES DA PESQUISA

Os dados relativos a qualidade do ar, umidade, temperatura etc são obtidos através de plataformas como INMET e SISAM. O INMET possui apenas uma estação meteorológica na cidade de Cuiabá, e o SISAM obtém dados de monitoramento de queimadas através de satélites e dados meteorológicos através de estações do INPE. Portanto, não temos dados vastos para múltiplos pontos da cidade, tem de ser considerado então, que todos os habitantes estão sujeitos às mesmas condições climáticas registradas por esses órgãos.

A base que diz respeito à internação de pacientes foi retirada do Sistema de Informações Hospitalares do DataSUS, logo, não foram considerados diagnósticos que podem ter sido realizados na rede particular de hospitais.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho será divido em 5 capítulos, sendo o Capítulo 1 a introdução, onde foi apresentado o objetivo, justificativas e delimitações da pesquisa.

O Capítulo 2, será dedicado à esclarecer o fundamento sobre a linguagem de programação utilizada, os modelos empregados, as métricas e a relação entre inteligência artificial e saúde.

No Capítulo 3 tem-se a explicação sobre o porquê da escolha de Cuiabá como local de estudo, bem como a obtenção de todas as bases de informações utilizadas nesse estudo.

O Capítulo 4 é inteiramente dedicado à análise dos resultados obtidos a partir dos modelos executados a partir dos dados.

No Capítulo 5 explora-se os objetos motivadores desta pesquisa e termina-se a análise comparativa entre os resultados atuais e anteriores, bem como propor melhorias para o trabalho já executado.

Por fim, estão a Bibliografia, e os Anexos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados brevemente alguns conceitos utilizados na realização deste trabalho.

2.1 PYTHON E CIÊNCIA DE DADOS

Python é uma linguagem de programação de alto nível e código aberto (*open source*), conhecida por sua simplicidade e legibilidade, priorizando a produtividade e interpretação do código pelo programador ao rigor da sintaxe.

O Python virou a principal opção para codificação na área de dados, graças ao conjunto amplo de ferramentas específicas e material extensivo pela internet. Bibliotecas como Pandas ou NumPy trazem formas rápidas de lidar com números e conjuntos grandes, já o Scikit-learn é usado nas funções tradicionais de machine learning. Esse destaque aparece claro em levantamentos reais, como a pesquisa anual sobre ciência de dados feita pelo Kaggle mostrando que mais de 85% dos especialistas em ciência de dados usam essa linguagem no dia a dia (KAGGLE, 2022).

2.2 INTELIGÊNCIA ARTIFICIAL E SAÚDE

Agentes de inteligência artificial estão por toda parte no nosso dia a dia, desde embutidos em aplicativos como WhatsApp, como em ferramentas de pesquisa como Google. Mas o que é inteligência artificial?

De acordo com Russel e Norvig:

> O campo da inteligência artificial, ou IA, não se preocupa apenas em compreender, mas também em construir entidades inteligentes - máquinas que podem calcular como agir de forma eficaz e segura em uma ampla variedade de situações novas (RUSSELL; NORVIG, 2021, p. 1).

Na área da saúde, a inteligência artificial está inserindo uma nova forma de diagnósticos precoces. Ao invés de depender apenas dos médicos, sistemas autônomos vasculham uma grande gama de informações, como históricos digitais, exames de imagem, leitura de sensores ou mesmo sequências genéticas, e encontram padrões complexos demais para o cérebro humano capturar (THIRUNAVUKARASU et al., 2023). O foco não é substituir médicos por

robôs, mas sim, fornecer melhores ferramentas para atender os pacientes, aumentando a capacidade de atendimento e eficiência terapêutica.

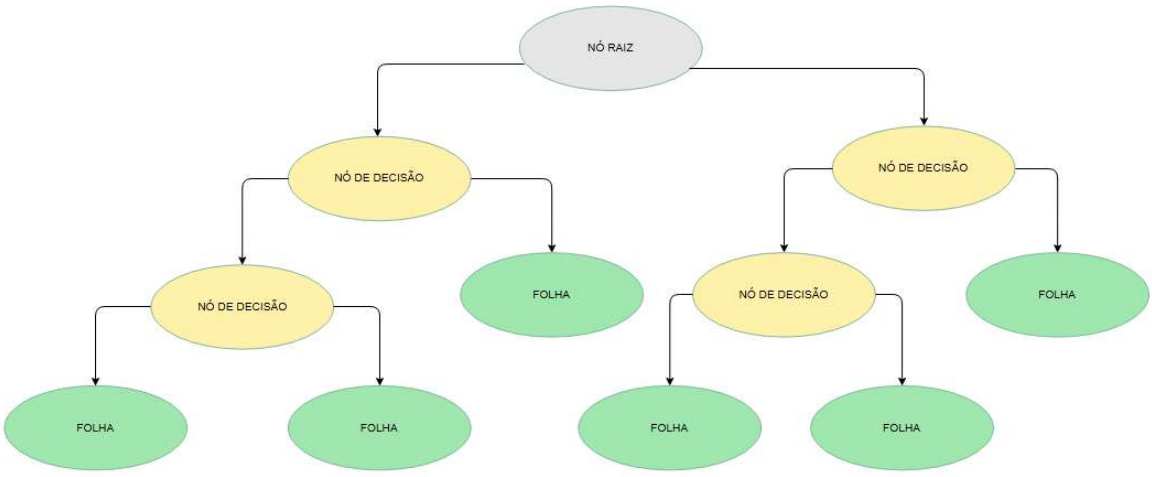
A área de diagnóstico por imagem é a mais beneficiada pelo avanço da tecnologia de deep learning, onde redes neurais convolucionais se mostram altamente precisas e eficientes na detecção de patologia e radiologia digital. Estudos recentes indicam que esses algoritmos já alcançam e algumas vezes superam os especialistas humanos, majoritariamente na detecção de cânceres, lesões dermatológicas e anomalias neurológicas (MAYER; ECKHARDT, 2024; WANG et al., 2023).

Abaixo, serão apresentados os modelos de aprendizado de máquina utilizados neste trabalho.

2.2.1 Árvore de decisão

Árvore de decisão é um modelo muito similar ao processo de tomada de decisão humano. A lógica dele consiste em segmentar os dados em subconjuntos menores se baseando nos dados das variáveis de entrada. Como visto na Figura 3, o ponto de partida é o nó raiz, onde fica contido todo o conjunto de dados de treinamento. À seguir, ele vai descendo para os nós de decisão de acordo com os valores relevantes dos dados que chegaram a este nó de decisão em questão. A última etapa da árvore é o nó folha, onde obtemos a previsão do modelo.

Figura 3 - Exemplo de estrutura de árvore de decisão



Fonte: Elaborado pelo autor (2025)

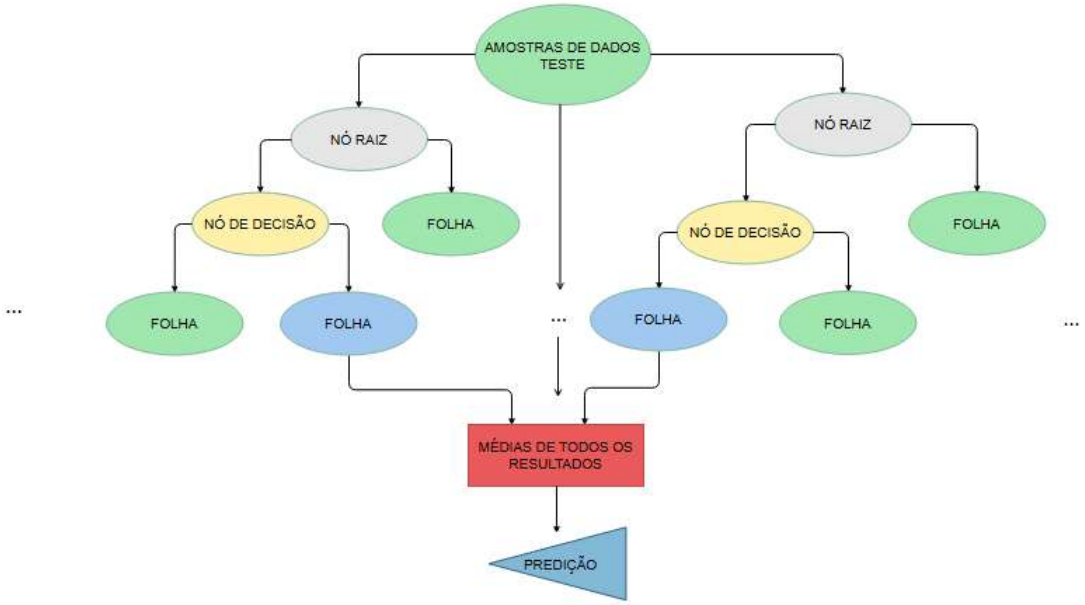
Este modelo é versátil, podendo atuar tanto para problemas onde as variáveis são categóricas quanto para modelos de regressão. Neste trabalho utilizaremos a função de regressão. Para a atuação em regressão, o modelo funciona dividindo os nós em subconjuntos cujo MSE seja o menor possível. Após ser realizada essa primeira divisão, esse método é aplicado recursivamente até que seja gerado um nó que atenda aos requisitos pretendidos.

2.2.2 Floresta aleatória

O algoritmo de floresta aleatória funciona de forma semelhante à árvore de decisão, sendo utilizado para problemas de regressão e classificação. Também é um modelo de aprendizado supervisionado e veio para suprir algumas falhas observadas no modelo de árvore de decisão.

O nome de “floresta” vem porque ele é um agrupamento de várias árvores de decisão. Elas são agrupadas tomando subconjuntos de dados aleatórios. Ao final, ele utiliza o valor médio de todos os resultados obtidos para compor o valor final da predição. Pode-se ter uma ideia de como funciona observando a Figura 4.

Figura 4 - Exemplo de estrutura de floresta aleatória



Fonte: Elaborado pelo autor (2025)

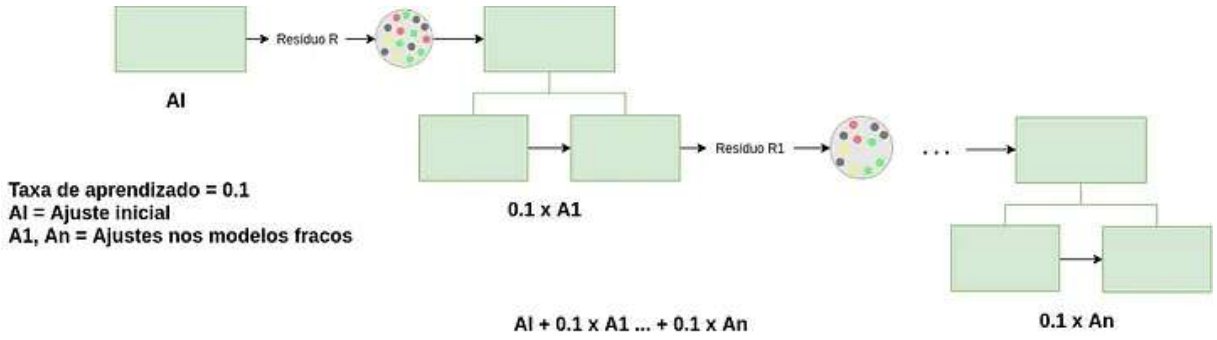
2.2.3 XGBoost Regressor

O nome *XGBoost* se origina de *Extreme Gradient Boosting*, e é um modelo que tem como base o algoritmo de árvores de decisão combinada com *gradient boosting*.

O modelo consiste inicialmente em construir árvores de decisão com pouca profundidade (poucos nós). Em seguida, são calculados os erros dessas árvores. Então, o próximo modelo é focado em corrigir os erros das árvores anteriores. Esses dois modelos são então combinados e esse processo todo repetido até que é criada uma sequência de modelos que se complementam e o modelo final é a combinação de todos esses modelos “fracos”. Essa técnica é chamada *gradient boosting*.

A Figura 5 exemplifica a implementação de um modelo de *XGBoost*.

Figura 5 - Exemplo de funcionamento do *XGBoost*



Fonte: Equals Lab (2020)

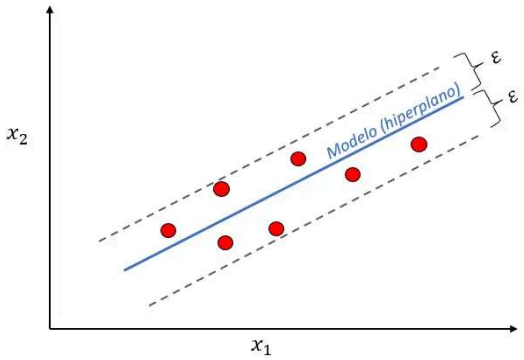
2.2.4 Support Vector Regression

O SVR é um algoritmo que pertence a classe do SVM (*Support Vector Machine*), mas enquanto o SVR é utilizado para problemas de regressão, o SVM é utilizado para problemas de classificação.

O objetivo deste algoritmo não é encontrar a minimização absoluta do erro entre os dados e o modelo, mas sim, encontrar a função (hiperplano) que melhor se ajuste entre os dados de treinamento e também gere uma fronteira de decisão, ou margem de erro, aceitável.

Um exemplo de função de SVR simples é observado na Figura 6.

Figura 6 - Exemplo de SVR



Fonte: Martin Melo Dias (2023)

2.3 SKLearn

O *Scikit-learn* é uma biblioteca amplamente utilizada para aprendizado de máquina em python. Ele é construído sobre outras bibliotecas científicas de código aberto dentro desta linguagem, como NumPy, SciPy e Matplotlib. Dentro dele estão incluídos todos os modelos de regressão que foram explicados acima.

Além disso, ele fornece outras ferramentas que auxiliarão os modelos a obter a melhor performance, como:

- Pré-processamento: *MinMaxScaler* para padronização das variáveis.
- Seleção dos dados: *train_test_split* para divisão dos dados e *GridSearchCV* para otimização de hiperparâmetros;
- Avaliação de desempenho: MSE, RMSE e MAPE;
- *Pipelines*: facilita e organiza a execução de múltiplas etapas de processamento e modelagem.

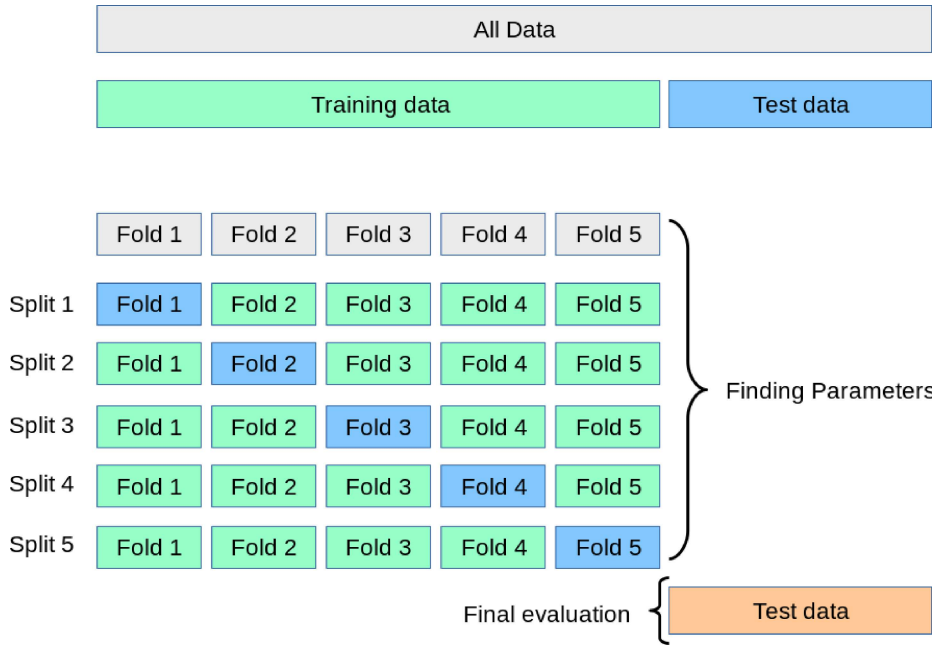
2.3.1 Validação Cruzada

A *cross validation* (validação cruzada) é uma técnica para avaliação de performance de modelos de forma mais robusta do que apenas a utilização da divisão 20/80% do *train_test_split*. Ela evita que se dependa da divisão aleatória dos dados, rodando múltiplos testes em diferentes subconjuntos de dados.

Ela funciona da seguinte maneira, na primeira rodada (*Split 1*) ele treina nos conjuntos 2,3,4 e 5 (*Folds*) e testa no conjunto 1 (*Fold 1*). Na próxima rodada (*Split 2*) ele treina nos conjuntos 1,3,4 e 5 (*Folds*) e testa no conjunto 2 (*Fold 2*), e assim sucessivamente até que as

rodadas se esgotem de acordo com os parâmetros fornecidos. Esses passos podem ser observados na Figura 7 a seguir.

Figura 7 - Funcionamento da validação cruzada



Fonte: SCIKIT-LEARN (2025)

Ao final das 5 rodadas, todos os subconjuntos de dados terão sido utilizados tanto para treino quanto para teste. Isso fornece uma estimativa mais confiável e estável para o resultado final, sendo que o desempenho final será calculado a partir das médias das pontuações do modelo para cada rodada.

2.3.2 *Pipeline*

O *pipeline* é uma ferramenta que permite executar múltiplas etapas de processamento e modelagem em um fluxo único. Então, substitui-se a necessidade de realizar todos os passos necessários várias vezes em todos os modelos para obtenção do resultado final. Uma vantagem clara é que ele evita que dados de teste "vazem" do conjunto, garantindo que a normalização, por exemplo, seja executada apenas no conjunto de treino de cada rodada da validação cruzada.

2.3.2.1 Escalonamento

A técnica utilizada em todos os modelos para normalização das variáveis será o *MinMaxScaler*. Ele consiste em redimensionar todos os dados para que fiquem em um intervalo entre 0 e 1. Isso é realizado para que todas as variáveis estejam dentro da mesma escala, isso faz com que todos os dados tenham o mesmo peso dentro dos modelos. Ele normaliza os dados utilizando a Equação 1:

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

sendo:

- x: o valor original da amostra;
- min(X): o valor mínimo de toda a coluna;
- max(X): o valor máximo de toda a coluna.

2.3.2.2 Seleção de Variáveis

A seleção de variáveis (*feature selection)* é um método onde o *scikit learn* seleciona subconjuntos das nossas variáveis de entrada que são mais relevantes para uso no modelo em questão. Neste trabalho é utilizado o método *SelectKBest* combinado com *f_regression* onde são selecionadas as 5 melhores variáveis de entrada para cada modelo baseada na pontuação que as variáveis performaram utilizando regressão.

2.3.4 *GridSearch*

O *GridSearch* é utilizado para automatizar a busca pelos melhores hiperparâmetros de cada modelo. O objetivo é que se encontre a melhor combinação dos parâmetros, sem ser necessária a execução manual de cada modelo com cada um dos parâmetros definidos.

Para o modelo de árvore de decisão, pode-se executar o modelo com os seguintes parâmetros:

- *feature_selection = [5, 7] :* tentará usar 5 ou 7 variáveis de entrada
- *max_depth = [None, 10, 20, 30]* : definir a profundidade máxima da árvore;

- *min_samples_split = [2, 5, 10]* : define o número mínimo de amostras que cada nó precisa ter para ser dividido entre mais nós;
- *min_samples_leaf = [1, 2, 4]* : garante que cada nó terá pelo menos 1, 2 ou 4 amostras após a divisão.

Neste contexto, o *GridSearch* irá testar todas as combinações desses parâmetros, ou seja, o modelo de árvore de decisão será executado 72 vezes, já que o número total de combinações se dá pela multiplicação dos parâmetros, nesse caso $2 \cdot 4 \cdot 3 \cdot 3$, resultando em 72 execuções.

A partir destas 72 execuções, ele irá selecionar a melhor combinação e utilizar posteriormente na etapa de predição.

Para cada modelo na seção de modelagem do problema serão explicados os hiperparâmetros utilizados.

2.4 MÉTRICAS PARA AVALIAR OS MODELOS

As métricas utilizadas para avaliar o desempenho do modelo funcionam relacionando o erro entre o valor previsto obtido pelo modelo e o valor esperado. Algumas métricas utilizadas foram o erro quadrático médio (*MSE - Mean Squared Error*), raíz do erro quadrático médio (*RMSE - Root Mean Squared Error*), erro médio absoluto (*MAE - Mean Absolute Error*) e percentual do erro médio absoluto (*MAPE - Mean Absolute Percentual Error*).

2.4.1 Erro quadrático médio - *MSE*

O erro quadrático médio é calculado pela Equação 2:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \quad (2)$$

sendo:

- n : número de amostras;
- y_i : valor de teste;
- $\widehat{y}_i$: valor da predição do modelo.

Ele calcula a média da diferença entre o valor predito pelo modelo e o real, como a diferença entre esses valores é elevada ao quadrado, valores que tenham uma diferença maior, ficam ainda maiores, penalizando mais os valores que não performaram bem com o modelo, os *outliers*.

2.4.2 Raiz do erro quadrático médio - *RMSE*

O *RMSE* é calculado pela Equação 3:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (3)$$

sendo:

- n : número de amostras;
- y_i : valor de teste;
- $\hat{y}_i$: valor da predição do modelo.

Seguindo a mesma premissa do *MSE*, onde quanto maior a diferença mais penalizado o dado vai ser, porém voltando a unidade do erro para a unidade de medição do dado.

2.4.3 Erro médio absoluto - *MAE*

O *MAE* é calculado através da Equação 4:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \quad (4)$$

sendo:

- n : número de amostras;
- y_i : valor de teste;
- $\hat{y}_i$: valor da predição do modelo.

O *MAE* mede a diferença entre o valor real e o valor predito, como está em módulo, não há valores negativos. Esta métrica não é afetada pelos *outliers*, ou seja, valores discrepantes não ficam tão aparentes quanto no *MSE* e *RMSE*.

2.4.4 Percentual do erro médio absoluto - *MAPE*

Calcula-se o *MAPE* pela Equação 5:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i - \widehat{y_i}\right|}{max(\varepsilon, |y_i|)} \quad (5)$$

sendo:

- n : número de amostras;
- y_i : valor de teste;
- $\widehat{y_i}$: valor da predição do modelo;
- ε : número de controle para evitar divisões por zero.

O *MAPE* calcula a porcentagem de erro em relação aos valores reais.

3 MATERIAIS E MÉTODOS

Neste capítulo será abordado a descrição do problema, desde a escolha da cidade de Cuiabá até os dados que serão utilizados nos modelos.

3.1 LOCAL DE ESTUDO

A cidade de Cuiabá é a capital do estado do Mato Grosso, localizado na região centro-oeste do país. Situada entre o Pantanal, a Amazônia e o Cerrado, ela serve como ponto de encontro desses três biomas brasileiros. De acordo com o censo agropecuário realizado em 2017, aproximadamente 1522,13 km^2 são usados para agropecuária (IBGE, 2017), levando em conta que a área total da cidade é de 4.327,22 km^2 (IBGE, 2022) , aproximadamente 35% de seu território é usado para plantio e pastos. A Figura 8 mostra a localização da cidade em destaque no estado de Mato Grosso.

Figura 8 - Localização de Cuiabá

Fonte: Wikimedia (2018)

Devido a múltiplas queimadas no Parque Nacional de Chapada dos Guimarães, foi realizado um estudo que detectou a concentração de até $86,3\ \mu g/m^3$ (MPMT, 2019), com média diária de $70,1\ \mu g/m^3$. A recomendação da OMS (Organização Mundial da Saúde) é que a concentração não supere $50\ \mu g/m^3$ (MPMT, 2019).

A Tabela 1 apresenta alguns dados econômicos e populacionais sobre a capital:

Tabela 1 - Dados da cidade de Cuiabá

Dados	Valores
Área territorial	4.327,220 km2 [2024]
População estimada	691.875 pessoas [2025]
Densidade demográfica	150,41 hab/km2 [2022]
Mortalidade infantil	11,69 óbitos por mil nascidos vivos [2023]
PIB per capita	47.700,88 R$ [2021]

Adaptado de: IBGE (2025)

Além da grande quantidade de queimadas que ocorrem na região, outros fatores que favorecem o aumento da concentração de material particulado no ar são a emissão pelas indústrias locais, queima de lixo urbano e o transporte de material particulado de cidades vizinhas.

Devido aos fatores acima que aumentam a poluição atmosférica, tomamos o estudo de Seinfeld e Pandis (2016), demonstra que altas temperaturas atuam como aceleradores da reação na qual gases como dióxido de enxofre (SO_2) e óxidos de nitrogênio (NO_x) se convertem em material particulado. Devido às altas temperaturas, a velocidade de vento também é baixa, o que faz com que o material particulado presente na cidade não se dissipe. O resultado é a alta taxa de inalação desses componentes pelos habitantes da cidade.

De acordo com o exposto acima, Cuiabá é um alvo promissor para o avanço no objetivo deste trabalho.

3.2 OBTENÇÃO E TRATAMENTO DOS DADOS

Os dados utilizados para a modelagem de todo o problema foram retirados de três bancos de dados:

-Sistema Integrado de Serviços Ambientes (SISAM);

-Instituto Nacional de Meteorologia (INMET);

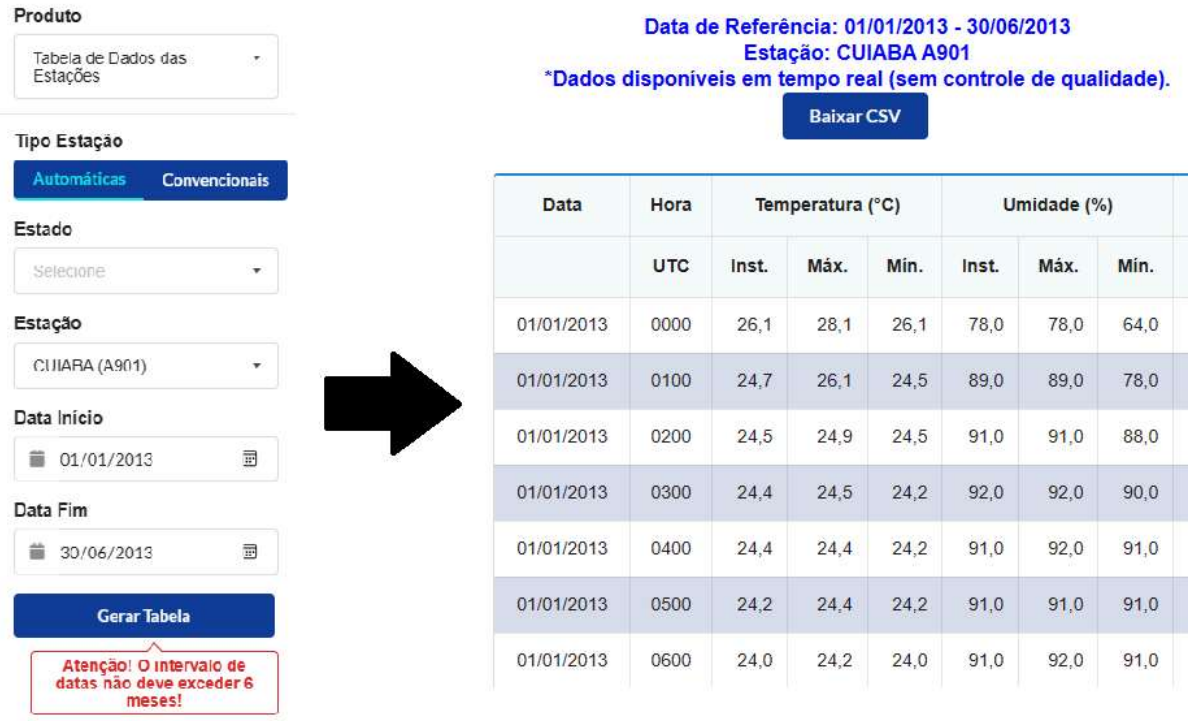
-Sistemas de Informações Hospitalares do SUS (SIHSUS).

Seguindo uma das propostas de melhoria indicadas pelo Matheus Cerqueira (2023). Houve a adição da base do INMET, que vai trazer mais variáveis de entrada relativas à informações meteorológicas.

3.2.1 INMET

Ao acessar o site do INMET, é necessário selecionar o “Produto”, no caso deseja-se a base de dados “bruta”, então selecionamos “Tabela de Dados das Estações”. Em seguida clicamos em “Automáticas”, pois é o modo como a estação meteorológica localizada em Cuiabá funciona. A seguir, selecionamos a estação “Cuiabá (A901)” da lista de estações. Por limitação do site, deve-se fazer a exportação dos dados em períodos de 6 meses. Ao colocar o primeiro período, 01/01/2013 à 30/06/2013, clicamos em “Gerar Tabela”. Os passos anteriores podem ser vistos na Figura 9.

Figura 9 - Obtenção de dados INMET



Adaptado de: INMET (2025)

Após clicar em “Gerar Tabela”, somos direcionados à tela seguinte mostrada no lado direito da Figura 9. Deve-se então clicar em “Baixar CSV”, em seguida o arquivo contendo todos os dados de poluentes é transferido para nosso computador.

Deve-se repetir os passos acima para períodos de seis meses até o fim do ano de 2018, onde obtém-se o último conjunto de dados que se necessita do INMET. Como resultado tem-se 12 arquivos que irão compor toda a base de dados do INMET.

Após a obtenção de todos os arquivos, o próximo passo é concatená-los em uma única tabela para facilitar o uso para os modelos.

Quanto às colunas da base de dados do INMET, as de interesse para estudo estão no Quadro 1, todas as outras são descartadas nesse escopo:

Quadro 1 - Descrição das variáveis do INMET

Variável	Tipo	Descrição
Data	Numérico	Data da medição
Temp. Ins. (C)	Numérico	Temperatura em °C
Umi. Ins. (%)	Numérico	Umidade em %
Chuva (mm)	Numérico	Precipitação em mm

Fonte: Elaborado pelo autor (2025)

Como observado na Figura 9 acima, todas as medições são realizadas 24 vezes ao dia, uma medição por hora. Então para se ter um melhor agrupamento dos dados, é feita a transformação de 24 medições por dia para apenas uma, realizando as seguintes alterações:

- A coluna “Temp. Ins. (C)" é convertida em duas colunas, “MinTemp” contendo a temperatura mínima e “MaxTemp” a temperatura máxima registrada no dia.
- A coluna “Umi. Ins. (%)” é transformada para a média de todas as medições do dia.
- Similar ao feito na coluna “Umi. Ins. (%)”, a coluna “Chuva (mm)” também é transformada em uma linha por dia contendo apenas a média de todos os valores registrados.

Com as colunas “MinTemp” e “MaxTemp”, calculamos a nova coluna “AmpTemp” que vai auxiliar na elaboração de um modelo com uma precisão maior. Essa coluna é calculada conforme a Equação 6 mostrada a seguir:

$$AmpTemp = MaxTemp - MinTemp \quad (6)$$

Ao final do tratamento dos dados do INMET, obtém-se a base com os dados organizada como se vê na Figura 10:

Figura 10 - Dados tratados do INMET

	Data	MinTemp	MaxTemp	AmpTemp	MedUmi	MedChu
0	2013-01-01	23.9	33.3	9.4	75.91667	0.08333
1	2013-01-02	24.2	31.8	7.6	73.50000	0.00000
2	2013-01-03	24.5	33.9	9.4	73.45833	0.20000
3	2013-01-04	24.2	30.7	6.5	75.83333	0.07500
4	2013-01-05	23.7	31.8	8.1	78.25000	0.05833
...	...	...	...	...	...	...
2186	2018-12-27	24.0	31.9	7.9	74.12500	0.00000
2187	2018-12-28	23.0	29.1	6.1	81.00000	2.16667
2188	2018-12-29	23.3	28.7	5.4	81.30435	0.32174
2189	2018-12-30	25.7	27.9	2.2	69.80000	0.00000
2190	2018-12-31	23.4	25.0	1.6	88.37500	1.45833

2191 rows × 6 columns

Fonte: Elaborado pelo autor (2025)

Com esses dados organizados e prontos para uso, a etapa seguinte é o download das bases do SISAM.

3.2.2 SISAM

No momento do download, era possível selecionar um período de datas de até um ano, após isso selecionava-se o estado e a cidade, e as variáveis meteorológicas e poluentes atmosféricos. Em seguida, clicando no botão “Gerar arquivo CSV” é feito o download de todos os dados selecionados para o computador.

Para o projeto, foi selecionado o período de 2013 até 2018, o que resultou em seis arquivos CSV, um para cada ano. Com todos os arquivos já baixados, é necessário juntá-los em um único arquivo, para facilitar o tratamento e utilização dos dados. Isso foi feito utilizando a biblioteca Pandas.

As estações meteorológicas fazem 4 medições por dia para todas as variáveis, então a melhor maneira de lidar com essa quantidade de dados foi calcular a média diária de todos esses valores. A Figura 11 apresenta os dados antes do tratamento, onde possui 8.764 linhas, e também apresenta valores nulos como pode ser visto nas colunas de *precipitacao_mmdia*, *co_ppb* e *focos_queimada*, por exemplo.

A Figura 12 apresenta os dados depois do tratamento, onde foram agrupados por dia, todos os valores diários são a média dos quatro valores mostrados na Figura 11. Os dados que antes eram nulos, foram substituídos por zero. Assim, o primeiro conjunto de dados

atmosféricos e de poluentes ficou com 2.191 linhas, que são todos os dias do período de seis anos.

Inicialmente, tem-se na base um total de 17 colunas, destas usaremos apenas as 9 indicadas e descritas no Quadro 2.

Quadro 2 - Descrição das variáveis do SISAM

Variável	Tipo	Descrição
datahora	Numérico	Data da medição
co_ppb	Numérico	Concentração de monóxido de carbono em partes por bilhão
o3_ppb	Numérico	Concentração de ozônio em partes por bilhão
pm25_ugm3	Numérico	Concentração de material particulado de 2,5 micrômetros ou menos no ar em microgramas por metro cúbico
so2_ugm3	Numérico	Concentração de dióxido de enxofre em microgramas por metro cúbico
temperatura_c	Numérico	Temperatura em graus celsius
umidade_relativa_percentual	Numérico	Umidade relativa percentual
precipitacao_mmdia	Numérico	Precipitação em mm por dia
focos_queimada	Numérico	Quantidade de focos de queimada

Fonte: Elaborado pelo autor (2025)

Figura 11 - Dados do SISAM antes do tratamento

	datahora	co_ppb	no2_ppb	o3_ppb	pm25_ugm3	so2_ugm3	precipitacao_mmdia	temperatura_c	umidade_relativa_percentual	focos_queimada	mun_geocod	mun_nome	mun_lat	mun_lon	mun_uf_nome	vento_direcao_grau	vento_velocidade_ms
0	2013-01-01 00:00	133.6	1.2	9.5	7.0	1.0	2.0	25.6	84	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	NaN	NaN
1	2013-01-01 06:00	152.3	2.5	6.2	8.4	1.7	NaN	23.7	94	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	NaN	NaN
2	2013-01-01 12:00	139.1	1.1	12.1	9.0	0.6	NaN	26.8	82	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	NaN	NaN
3	2013-01-01 18:00	116.6	0.1	16.7	8.3	0.2	NaN	29.7	67	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	NaN	NaN
4	2013-01-02 00:00	130.1	0.9	11.0	10.6	0.8	2.0	25.9	84	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
759	2018-12-30 18:00	NaN	NaN	NaN	NaN	NaN	NaN	30.0	90	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	313.0	3.8
760	2018-12-31 00:00	126.1	1.6	6.6	12.5	1.3	1.0	25.8	95	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	24.0	1.4
761	2018-12-31 06:00	NaN	NaN	NaN	NaN	NaN	NaN	23.8	98	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	348.0	2.3
762	2018-12-31 12:00	126.8	0.8	6.9	14.6	0.6	NaN	24.9	97	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	345.0	4.0
763	2018-12-31 18:00	NaN	NaN	NaN	NaN	NaN	NaN	25.7	96	NaN	5103403	CUIABÃ□	-15.5987	-56.0991	MATO GROSSO	278.0	3.7

64 rows × 17 columns

Fonte: Elaborado pelo autor (2025)

Figura 12 - Dados tratados do SISAM

	data	co_ppb	o3_ppb	pm25_ugm3	so2_ugm3	temperatura_c	umidade_relativa_percentual	precipitacao_mmdia	focos_queimada
0	2013-01-01	135.400	11.125	8.175	0.875	26.450	81.75	0.50	0.0
1	2013-01-02	155.050	10.925	11.675	1.000	26.400	82.00	0.50	0.0
2	2013-01-03	137.400	12.225	8.800	0.825	27.400	81.25	0.00	0.0
3	2013-01-04	128.300	13.750	11.650	0.925	27.750	76.25	0.25	0.0
4	2013-01-05	128.250	10.950	7.800	1.075	26.625	84.75	2.25	0.0
...	...	...	...	...	...	...	...	...	...
2186	2018-12-27	78.375	3.750	7.850	0.750	26.000	96.50	0.00	0.0
2187	2018-12-28	66.450	4.625	6.100	0.350	26.000	96.25	7.00	0.0
2188	2018-12-29	66.275	4.675	6.050	0.300	25.700	96.25	0.50	0.0
2189	2018-12-30	59.925	5.700	4.750	0.300	25.575	95.50	1.00	0.0
2190	2018-12-31	63.225	3.375	6.775	0.475	25.050	96.50	0.25	0.0

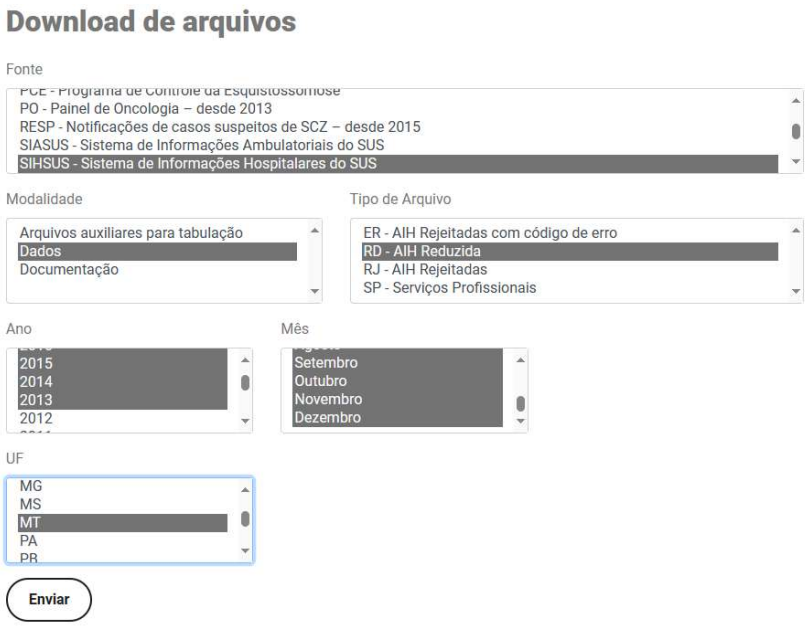
2191 rows × 10 columns

Fonte: Elaborado pelo autor (2025)

3.2.3 SIHSUS

O conjunto de dados relativos às doenças respiratórias foram obtidos pelo sistema de transferência de arquivos presente no site do Datasus. Onde é necessário selecionar a fonte: “Sistema de informações hospitalares do SUS (SIHSUS)”, modalidade: “Dados”, tipo de arquivo: “RD - AIH Reduzida”, o período: de 2013 até 2018, os meses do ano: janeiro a dezembro. A Unidade Federativa escolhida é a de MT (Mato Grosso), onde se encontra a cidade de Cuiabá, conforme mostrado na Figura 13.

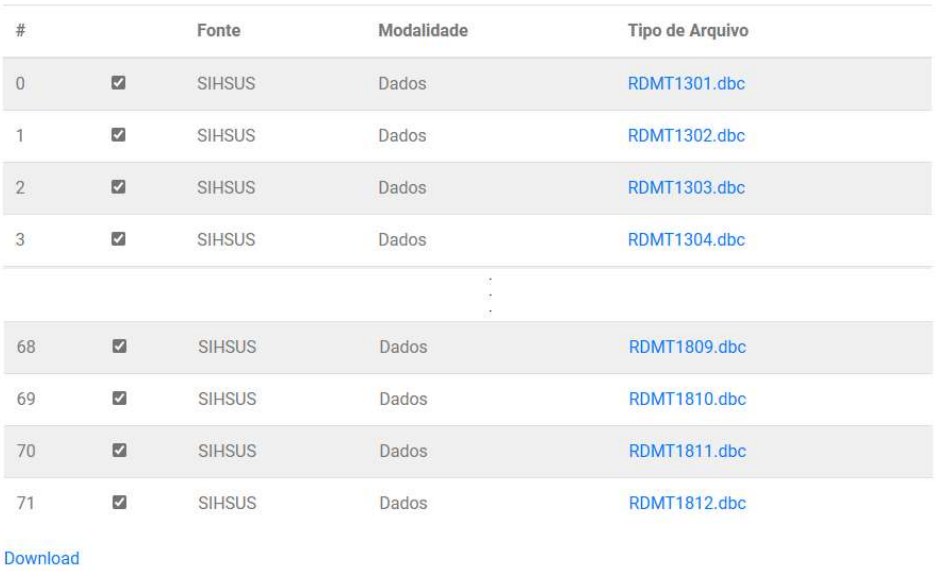
Figura 13 - Interface DATASUS



Fonte: DATASUS (2024)

Após clicar no botão enviar, é mostrada uma nova lista com os arquivos que podem ser baixados, onde foram retornados 72 arquivos, relativos a todos os meses do período de seis anos do projeto. Na Figura 14 aparecem os quatro primeiros arquivos, bem como os quatro últimos.

Figura 14 - Arquivos disponíveis para download DATASUS



#		Fonte	Modalidade	Tipo de Arquivo
0	☑	SIHSUS	Dados	RDMT1301.dbc
1	☑	SIHSUS	Dados	RDMT1302.dbc
2	☑	SIHSUS	Dados	RDMT1303.dbc
3	☑	SIHSUS	Dados	RDMT1304.dbc
⋮				
68	☑	SIHSUS	Dados	RDMT1809.dbc
69	☑	SIHSUS	Dados	RDMT1810.dbc
70	☑	SIHSUS	Dados	RDMT1811.dbc
71	☑	SIHSUS	Dados	RDMT1812.dbc

Download

Fonte: DATASUS (2024)

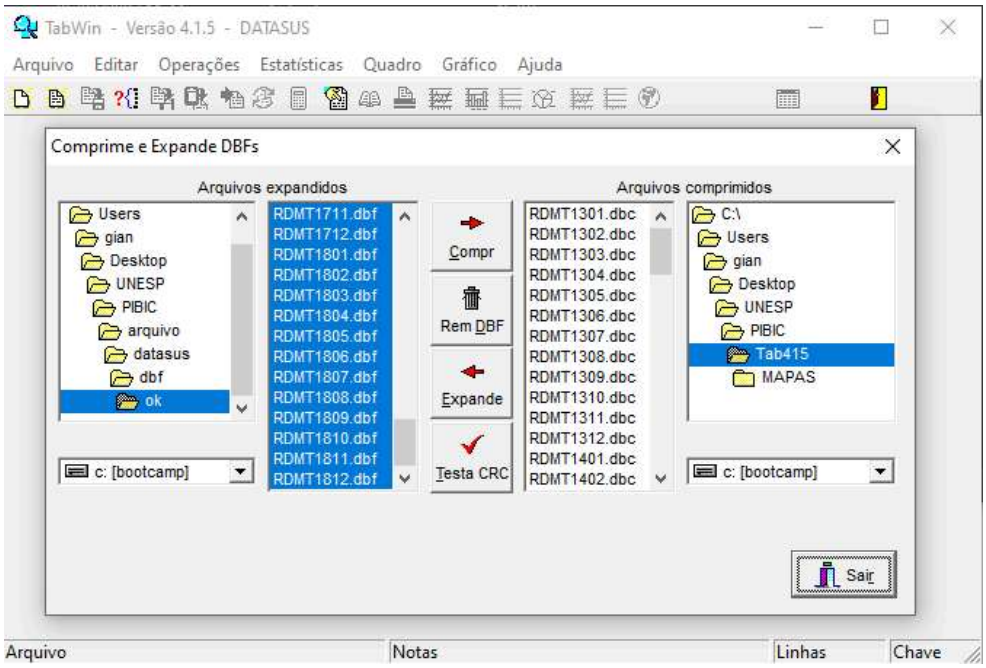
Ao clicar no botão “Download” no canto esquerdo da página, os 72 arquivos são transferidos para o computador, porém, eles estão no formato dbc, o que impossibilita sua utilização direta utilizando a biblioteca Pandas. Para solucionar isso, é utilizado o software TABWIN fornecido pelo próprio Datasus. Nele, os arquivos dbc são expandidos para outro formato próprio, o dbf. Após isso, realiza-se a última conversão, de dbf para csv, um formato que é compatível com a ferramenta Pandas. Ambos processos podem ser observados nas Figura 15 e 16.

Agora, com a base de dados do Datasus em csv, é possível realizar a filtragem dos dados de interesse, sendo estes as colunas “UF_ZI”, onde se registra o município de origem do paciente, “DT_INTER”, relativa a data de internação do paciente e a coluna “DIAG_PRINC”, que contém o código do diagnóstico.

A coluna de município foi filtrada pelo código da cidade de Cuiabá fornecido pelo IBGE, sendo esse ‘510340’. A fim de selecionar somente as doenças respiratórias, foi necessário consultar o índice Classificação Internacional de Doenças (CID-10), onde observa-se que a seção “J” contém todos os códigos de doenças respiratórias conforme

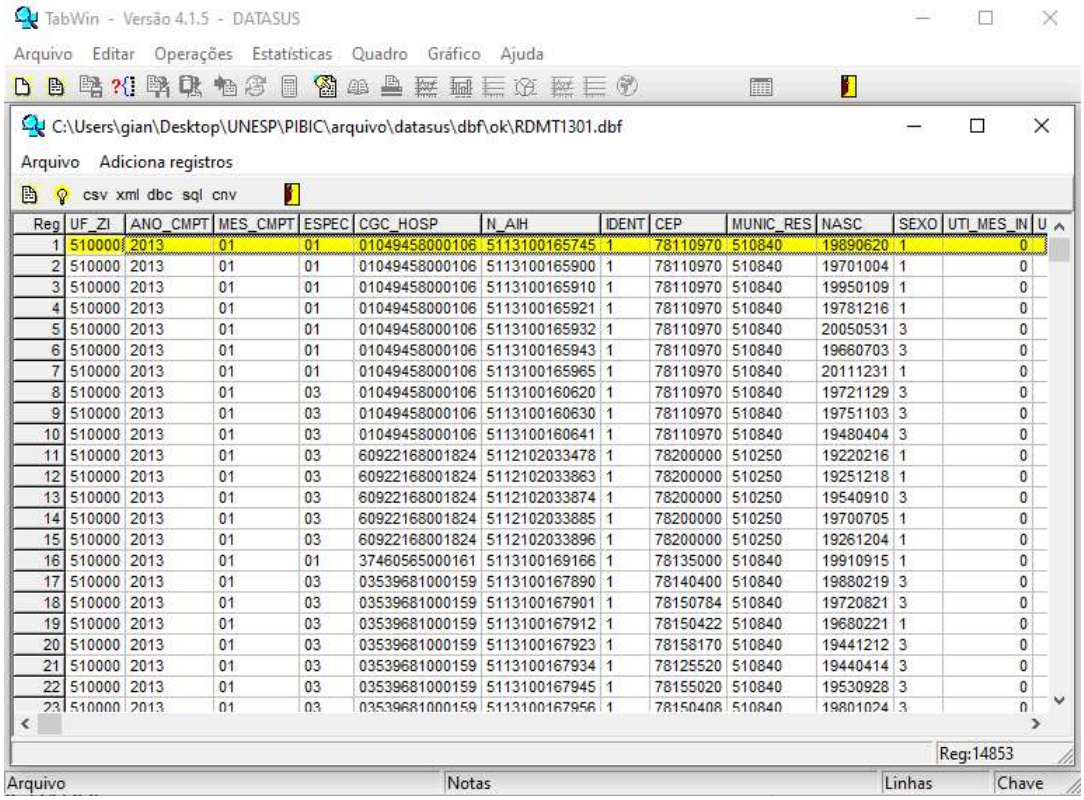
mostrado na Figura 17. Após isso, fez-se a seleção apenas das internações que começam pela letra J da base do Datasus.

Figura 15 - Compressão dos arquivos para .dbc



Fonte: DATASUS (2024)

Figura 16 - Conversão dos arquivos para csv



Fonte: DATASUS (2024)

Figura 17 - Lista de doenças CID-10

165-179	**Doenças do aparelho respiratório**	**J00-J99**
165	Faringite aguda e amigdalite aguda	J02-J03
166	Laringite e traqueíte agudas	J04
167	Outras infecções agudas das vias aéreas superiores	J00-J01, J05-J06
168	Influenza [gripe]	J09-J11
169	Pneumonia	J12-J18
170	Bronquite aguda e bronquiolite aguda	J20-J21
171	Sinusite crônica	J32
172	Outras doenças do nariz e dos seios paranasais	J30-J31, J33-J34
173	Doenças crônicas das amígdalas e das adenóides	J35
174	Outras doenças do trato respiratório superior	J36-J39
175	Bronquite, enfisema e outras doenças pulmonares obstrutivas crônicas	J40-J44
176	Asma	J45-J46
177	Bronquiectasia	J47
178	Pneumoconiose	J60-J65
179	Outras doenças do aparelho respiratório	J22, J66-J99

Fonte: DATASUS (2024)

A base do Datasus de janeiro de 2013 apresenta algumas internações relativas ao final do ano de 2012, essas entradas foram removidas da base de dados depois da filtragem. A última etapa a ser feita com esses dados é somar todas as internações que ocorreram no mesmo dia. O resultado final da filtragem pode ser visto na Figura 18.

Figura 18 - Dados do Datasus antes e depois de tratados

	DIAG_PRINC	data	respiratory
0	J189	2012-08-09	1
1	J152	2012-08-10	1
2	J159	2012-08-26	1
3	J449	2012-08-28	1
4	J985	2012-09-03	1
...	...	...	...
17347	J449	2018-12-13	1
17348	J180	2018-12-13	1
17349	J189	2018-12-18	1
17350	J159	2018-12-18	1
17351	J189	2018-12-23	1

17352 rows × 3 columns

	date	respiratory
0	2013-01-01	0
1	2013-01-02	0
2	2013-01-03	0
3	2013-01-04	0
4	2013-01-05	0
...	...	...
2186	2018-12-27	0
2187	2018-12-28	0
2188	2018-12-29	0
2189	2018-12-30	0
2190	2018-12-31	0

2191 rows × 2 columns

Fonte: Elaborado pelo autor (2025)

A Figura 19 apresenta a tabela com todas as variáveis de entrada disponíveis.

Figura 19 - Tabela com as variáveis

	data	co_ppb	o3_ppb	pm25_ugm3	so2_ugm3	temperatura_c	umidade_relativa_percentual	precipitacao_mmdia	focos_queimada	MinTemp	MaxTemp	AmpTemp	MedUmi	MedChu	respiratory
0	2013-01-01	135.400	11.125	8.175	0.875	26.450	81.75	0.50	0.0	23.9	33.3	9.4	75.91667	0.08333	6.0
1	2013-01-02	155.050	10.925	11.675	1.000	26.400	82.00	0.50	0.0	24.2	31.8	7.6	73.50000	0.00000	8.0
2	2013-01-03	137.400	12.225	8.800	0.825	27.400	81.25	0.00	0.0	24.5	33.9	9.4	73.45833	0.20000	12.0
3	2013-01-04	128.300	13.750	11.650	0.925	27.750	76.25	0.25	0.0	24.2	30.7	6.5	75.83333	0.07500	9.0
4	2013-01-05	128.250	10.950	7.800	1.075	26.625	84.75	2.25	0.0	23.7	31.8	8.1	78.25000	0.05833	10.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2186	2018-12-27	78.375	3.750	7.850	0.750	26.000	96.50	0.00	0.0	24.0	31.9	7.9	74.12500	0.00000	0.0
2187	2018-12-28	66.450	4.625	6.100	0.350	26.000	96.25	7.00	0.0	23.0	29.1	6.1	81.00000	2.16667	0.0
2188	2018-12-29	66.275	4.675	6.050	0.300	25.700	96.25	0.50	0.0	23.3	28.7	5.4	81.30435	0.32174	0.0
2189	2018-12-30	59.925	5.700	4.750	0.300	25.575	95.50	1.00	0.0	25.7	27.9	2.2	69.80000	0.00000	0.0
2190	2018-12-31	63.225	3.375	6.775	0.475	25.050	96.50	0.25	0.0	23.4	25.0	1.6	88.37500	1.45833	0.0

2191 rows × 15 columns

Fonte: Elaborado pelo autor (2025)

Por fim, obtém-se uma base com as variáveis sendo descritas no Quadro 3, com um total de 2.191 linhas e 15 colunas, totalizando 32.865 dados.

Quadro 3 - Resumo de todas as variáveis do trabalho

Variável	Tipo	Mínimo	Máximo	Unidade
data	Numérico	01/01/2013	31/12/2018	dias
co_ppb	Numérico	42,05	1394,92	ppb
o3_ppb	Numérico	1,27	52,87	ppb
pm25_ugm3	Numérico	0,60	185,45	$\mu g/m^3$
so2_ugm3	Numérico	0,22	9,12	$\mu g/m^3$
temperatura_c	Numérico	12,72	32,70	°C
umidade_relativa_percentual	Numérico	32,50	98,50	%
precipitacao_mm dia	Numérico	0,00	18,75	mm/dia
focos_queimada	Numérico	0,00	6,25	quantidade de focos de queimada
MinTemp	Numérico	8,70	35,4	°C
MaxTemp	Numérico	14,50	40,40	°C
AmpTemp	Numérico	0,00	19,20	°C
MedUmi	Numérico	19,96	92,21	%
MedChu	Numérico	0,00	4,07	mm/dia
respiratory	Numérico	0,00	28,00	quantidade de internados por dia

Fonte: Elaborado pelo autor (2025)

É interessante salientar que nem todas as internações do tipo J no CID-10 tem necessariamente relação com poluição atmosférica.

3.3 MODELAGEM DO PROBLEMA

A modelagem do problema se inicia após todos os dados terem sido tratados. A biblioteca *scikit-learn* dispõe de funções que fazem a divisão dos dados para treinamento e

teste automaticamente, sendo 80% para treino e 20% para teste. O modelo de XGBoost é uma exceção, já que em sua hiperparametrização vamos ter uma relação de divisão treino e teste diferente. Para garantir que os dados de todos os anos sejam separados igualmente, faz-se a estratificação por ano, utilizando o parâmetro *stratify.*

Na Figura 20, analisa-se a porcentagem de dados que serão utilizados de cada ano da base de dados:

Figura 20 - Estratificação dos dados por ano

Year	Overall %	Stratified %	Random %	Strat. Error %	Rand. Error %
2013	16.66	16.63	16.63	-0.18	-0.18
2014	16.66	16.63	18.45	-0.18	10.76
2015	16.66	16.63	15.72	-0.18	-5.65
2016	16.70	16.86	16.17	0.91	-3.18
2017	16.66	16.63	15.49	-0.18	-7.02
2018	16.66	16.63	17.54	-0.18	5.29

Fonte: Elaborado pelo autor (2025)

Pode-se observar que os valores na coluna "*Stratified %*" são muito mais próximos do que os contidos na coluna "*Random %*", esta última sendo a porcentagem de cada ano que seria utilizada se os dados não fossem estratificados.

A Tabela 2 indica a correlação entre as variáveis de entrada e a variável de saída. Mesmo que nenhuma delas apresente forte correlação, isso não significa que ela é irrelevante para o modelo, apenas que ela não está variando linearmente com a variável de saída.

Tabela 2 - Correlação das variáveis de entrada com a variável de saída

variáveis de entrada	Correlação
so2_ugm3	0.112684
o3_ppb	0.07583
co_ppb	0.037394
AmpTemp	0.02445
MaxTemp	0.006039
MedUmi	0.002364
focos_queimada	-0.009951
MinTemp	-0.020526
temperatura_c	-0.028572
pm25_ugm3	-0.029726
precipitacao_mmdia	-0.064309
MedChu	-0.072379
umidade_relativa_percentual	-0.171462

Fonte: Elaborado pelo autor (2025)

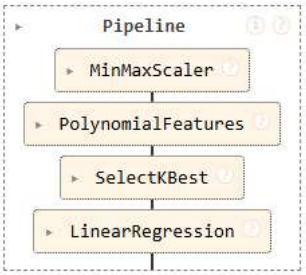
As variáveis de entrada utilizadas para cada modelo variam desde 3 até 13. Como será explicado abaixo a partir do item 3.3.1, cada modelos fará a seleção das variáveis que forem mais relevantes de acordo com os parâmetros otimizados retornados pelo *GridSearch*. A variável de saída será o número de pacientes internados no dia.

Abaixo, será mostrado a construção de pipelines para execução mais automática e organizada dos modelos, bem como o *GridSearch*, que nada mais é do que a hiperparametrização dos modelos.

3.3.1 Modelo de Regressão Linear

O primeiro modelo a ser utilizado é o de regressão linear, por ser um dos mais simples porém a hiperparametrização é a mais complexa. A Figura 21 mostra o como o pipeline fica após sua construção:

Figura 21 - *Pipeline* do modelo de regressão linear



Fonte: Elaborado pelo autor (2025)

Para esse modelo, o pipeline foi construído com os seguintes parâmetros:

- *MinMaxScaler*: normaliza os dados no intervalo de 0 e 1;
- *PolynomialFeatures*: permite a criação de novas variáveis para que sejam capturadas relações não-lineares ou relações mais complexas entre as variáveis.
- *SelectKBest*: recebe todas as variáveis lineares e não-lineares e seleciona as que tem melhor relação com a variável de saída;
- Por fim, *LinearRegression*: é o estimador de regressão linear.

Após a aplicação do pipeline, inicia-se o conjunto de parâmetros a serem explorados pelo *GridSearch.* São três conjuntos de parâmetros a serem testados, sendo eles:

1. *grid_linear:* o mais simples, com as seguintes atribuições:

a. *poly_degree = [1,2]* : determina que o número de variáveis polinomiais seja 1 ou 2;
b. *feature_selection = [3, 5, 7]* : define o número de variáveis de entrada seja 3, 5 ou 7;
c. *regression = LinearRegression* : define que o modelo utilizado seja de regressão linear

2. *grid_ridge*: utiliza os mesmos parâmetros *a* e *b* acima, adicionalmente:
    a. altera o modelo de *regression* para *Ridge*, que aplica penalidades maiores aos coeficientes mais distantes do alvo;
    b. necessita do parâmetro *alpha = [0.1, 1, 10]* : utilizado pelo modelo Ridge para forçar normalização dos coeficientes;

3. *grid_lasso*: faz o mesmo que o *grid_ridge*, apenas alterando o modelo para *Lasso*

Após a definição de todos os três parâmetros, executamos a função *GridSearchCV*, onde ela encontra que os melhores parâmetros a serem utilizados são:

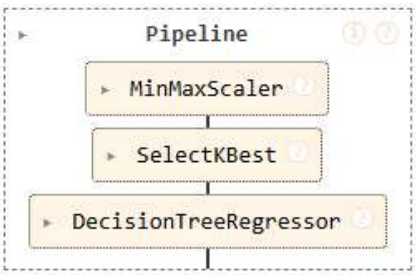
- *feature_selection* : 7 ;
- *poly_degree* : 2 ;
- *regression: LinearRegression.*

Finalmente, com esses parâmetros, a predição das variáveis de saída é executada.

3.3.2 Modelo de Árvore de Decisão

A árvore de decisão foi utilizada por ser mais robusta que a regressão linear. O *pipeline* utilizado se encontra na Figura 22.

Figura 22 - *Pipeline* do modelo da árvore de decisão



Fonte: Elaborado pelo autor (2025)

Para esse modelo, o pipeline foi construído com os seguintes parâmetros:

- *MinMaxScaler*: normaliza os dados no intervalo de 0 e 1;
- *SelectKBest*: recebe todas as variáveis lineares e não-lineares e seleciona as que tem melhor relação com a variável de saída;
- Por fim, *DecisionTreeRegressor*: é o estimador da árvore de decisão.

Após a aplicação do pipeline, inicia-se o conjunto de parâmetros a serem explorados pelo *GridSearch.* São três conjuntos de parâmetros a serem testados, sendo eles:

1. *feature_selection = [5, 10, 13]* : define que o número de variáveis de entrada seja 5, 10 ou 13;
2. *max_depth = [None, 10, 20, 30]* : define o número máximo de níveis que a árvore pode ter;
3. *min_samples_split = [2, 5, 10]* : controla o número mínimo que cada amostra precisa ter para se dividir;
4. *min_samples_leaf = [1, 2, 4]* : define o número mínimo de amostras que o nó precisa ter após a divisão.

Após a definição de todos os quatro parâmetros, executamos a função *GridSearchCV*, onde ela encontra os melhores parâmetros a serem utilizados, sendo eles:

- *feature_selection* : 5 ;
- *max_depth* : 10 ;
- *min_samples_split : 10;*
- *min_samples_leaf : 4.*

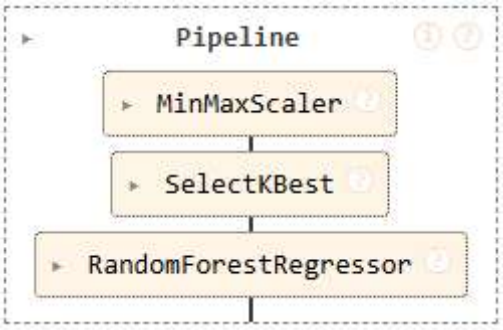
Finalmente, com esses parâmetros, a predição das variáveis de saída é executada.

3.3.3 Modelo de Floresta Aleatória

A floresta aleatória apresenta melhorias comparadas ao modelo anterior de árvore de decisão, e mecanismos que evitam o *overfitting*.

De modo similar ao anterior, o *pipeline* é apresentado na Figura 23.

Figura 23 - *Pipeline* do modelo da floresta aleatória



Fonte: Elaborado pelo autor (2025)

A única diferença entre o modelo de Floresta Aleatória e Árvore de Decisão é o estimador, que nesse caso usa-se o *RandomForestRegressor.*

Após a aplicação do *pipeline*, inicia-se o conjunto de parâmetros a serem explorados pelo *GridSearch.* São seis conjuntos de parâmetros a serem testados, sendo eles:

1. *feature_selection = [5, 10, 13]* : define que o número de variáveis de entrada seja 5, 10 ou 13;
2. *max_depth = [None, 10, 20, 30]* : define o número máximo de níveis que a árvore pode ter;
3. *min_samples_split = [2, 5, 10]* : controla o número mínimo que cada amostra precisa ter para se dividir;
4. *min_samples_leaf = [1, 2, 4]* : define o número mínimo de amostras que o nó precisa ter após a divisão.
5. *n_estimators = [100, 200, 300]* : define o número de árvores na floresta inteira;
6. *max_features = [5, 7, 10]* : número máximo de variáveis para encontrar a melhor divisão de nós.

Após a definição de todos os seis parâmetros, executamos a função *GridSearchCV*, onde ela encontra os melhores parâmetros a serem utilizados, sendo eles:

- *feature_selection* : 7 ;
- *max_depth* : 10 ;
- *min_samples_split : 10;*
- *min_samples_leaf : 4;*
- *n_estimators:* 300 ;
- *max_features:* 5.

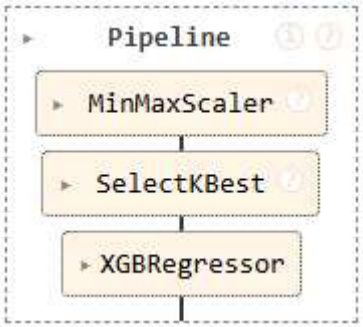
Finalmente, com esses parâmetros, a predição das variáveis de saída é executada.

3.3.4 Modelo XGBRegressor

O *XGBoost* foi usado com o intuito de obter uma melhor predição devido à forma mais completa com que o modelo trabalha.

O *pipeline* para o modelo se encontra na Figura 24.

Figura 24 - *Pipeline* do modelo XGBRegressor



Fonte: Elaborado pelo autor (2025)

Após a aplicação do pipeline, inicia-se o conjunto de parâmetros a serem explorados pelo *GridSearch.* São seis conjuntos de parâmetros a serem testados, sendo eles:

1. *feature_selection = [3, 5, 7]* : define que o número de variáveis de entrada seja 3, 5 ou 7;
2. *n_estimators = [100, 200, 300]* : define o número de árvores que o modelo vai construir sequencialmente;
3. *max_depth = [3, 5, 7]* : define o número máximo de níveis que a árvore pode ter;
4. *learning_rate = [0.01, 0.1, 0.2]* : é a taxa de aprendizado, nada mais é do que o peso que cada árvore terá ao ser adicionada ao modelo. Para valores menores, o aprendizado é mais lento e cauteloso, para maiores, é mais rápido porém com mais chance de ocorrer overfitting;
5. *subsample = [0.8, 0.9, 1.0]* : controla a fração do conjunto de treino que será utilizada no modelo;
6. *colsample_bytree = [0.8, 0.9, 1.0]* : controla a fração de colunas (variáveis de entrada) que serão utilizadas na construção de cada árvore.

Após a definição de todos os seis parâmetros, executamos a função *GridSearchCV*, onde ela encontra os melhores parâmetros a serem utilizados, sendo eles:

- *feature_selection* : 7 ;
- *n_estimators* : 200 ;
- *max_depth : 5 ;*
- *learning_rate : 0.01 ;*
- *subsample:* 0.8 ;
- *colsample_bytree:* 1.0 .

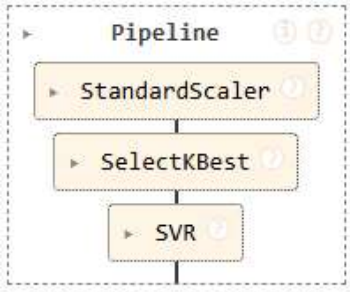
Finalmente, com esses parâmetros, a predição das variáveis de saída é executada.

3.3.5 Modelo Support Vector Regression

O SVR foi utilizado como método alternativo já que é um modelo que não busca minimizar o erro, diferentemente dos modelos anteriores.

O pipeline para construção do modelo é visto na Figura 20.

Figura 25 - *Pipeline* do modelo de SVR



Fonte: Elaborado pelo autor (2025)

Após a aplicação do pipeline, inicia-se o conjunto de parâmetros a serem explorados pelo *GridSearch.* São seis conjuntos de parâmetros a serem testados, sendo eles:

1. *feature_selection = [3, 5, 7]* : define que o número de variáveis de entrada seja 3, 5 ou 7;
2. *C = [0.1, 1, 10, 100]* : controla quanto peso do erro dentro do modelo, quanto menor, a margem de erro é maior, o que evita o overfitting mas pode trazer um modelo impreciso, e ao contrário, quanto maior o *C*, também é maior a chance de overfitting;
3. *epsilon = [0.01, 0.1, 0.5, 1]* : define a margem de tolerância ao erro, ou seja, define o hiperplano para que os pontos fora deste sejam penalizados de acordo com o *C*;

4. *kernel = [linear, rbf, poly]* : linear tenta encontrar um hiperplano, já o rbf (Radial Basis Function) é usado para relações não-lineares, por fim, o poly usa uma função polinomial para transformar os dados;
5. *gamma = [scale, auto]* : *scale* utiliza o número de variáveis e sua variância para definir a influência de um único ponto ao modelo, já *auto* utiliza apenas o número de variáveis para o cálculo.

Após a definição de todos os cinco parâmetros, executamos a função *GridSearchCV*, onde ela encontra os melhores parâmetros a serem utilizados, sendo eles:

- *feature_selection* : 7 ;
- *C* : 1;
- *epsilon : 0.1 ;*
- *kernel : rbf ;*
- *gamma : scale* .

Finalmente, com esses parâmetros, a predição das variáveis de saída é executada.

4 ANÁLISE DOS RESULTADOS

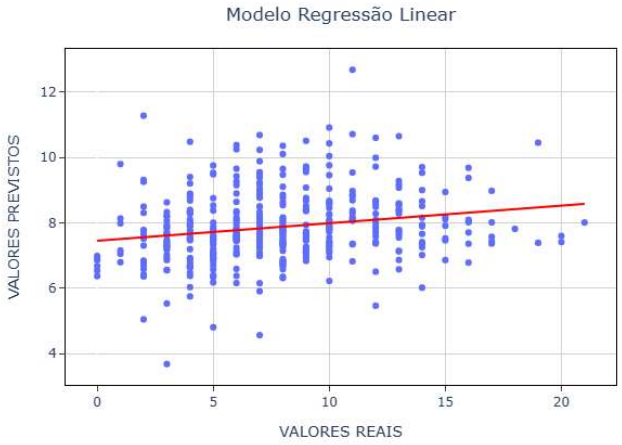
Neste capítulo serão analisados os resultados obtidos por cada um dos modelos, bem como comparar os erros e verificar qual deles obteve o melhor desempenho. Para cada modelo teremos um gráfico de dispersão que mostra os valores reais x valores previstos. Também existe a linha de tendência que ajuda a identificar a performance do modelo.

4.1 REGRESSÃO LINEAR

Conforme se observa na Tabela 3, apesar do modelo de regressão linear ser o mais simples, ele teve um bom desempenho, com RMSE de aproximadamente 4 internações, isso quer dizer que, em média, para cada ponto de valor real escolhido no gráfico, o modelo erraria em 5 internados para mais ou para menos naquele ponto.

Na Figura 26 pode-se ver o gráfico que relaciona os valores reais das internações com os valores previstos pelo modelo.

Figura 26 - Diagrama de dispersão de regressão linear



Fonte: Elaborado pelo autor (2025)

Tabela 3 - Métricas do modelo de Regressão Linear

RMSE	MAE	MAPE
3.8979	3.0819	3.37E+14

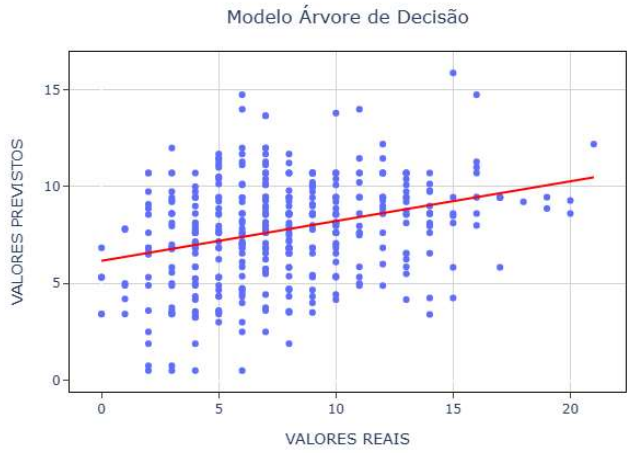
Fonte: Elaborado pelo autor (2025)

4.2 ÁRVORE DE DECISÃO

Diferente do modelo de regressão linear, a árvore de decisão teve um desempenho pior do que era imaginado, visto que deveria ser um modelo com capacidade superior ao de regressão linear devido à sua capacidade de correlacionar variáveis não lineares.

O RMSE obtido, como se vê na Tabela 4, foi de aproximadamente 4 internações.

Figura 27 - Diagrama de dispersão de árvore de decisão



Fonte: Elaborado pelo autor (2025)

Tabela 4 - Métricas do modelo Árvore de Decisão

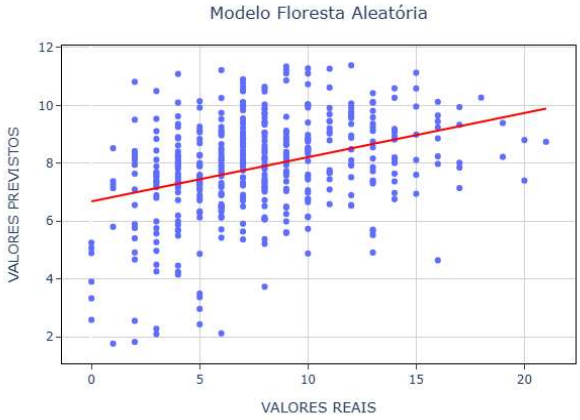
RMSE	MAE	MAPE
3.9959	3.1984	1.36E+14

Fonte: Elaborado pelo autor (2025)

4.3 FLORESTA ALEATÓRIA

O modelo de floresta aleatória trouxe um resultado melhor se comparado com os dois modelos anteriores. Na Tabela 5 se observa que seu RMSE foi de aproximadamente 4 internações.

Figura 28 - Diagrama de dispersão de floresta aleatória



Fonte: Elaborado pelo autor (2025)

Tabela 5 - Métricas do modelo Floresta Aleatória

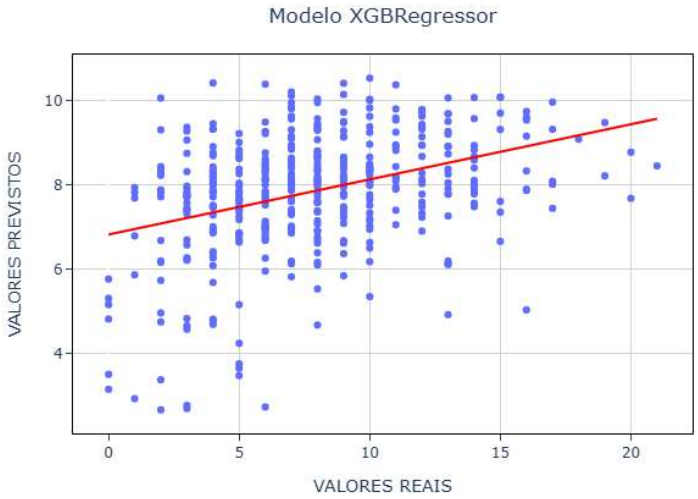
RMSE	MAE	MAPE
3.7346	2.9713	1.93E+14

Fonte: Elaborado pelo autor (2025)

4.4 XGBREGRESSOR

O modelo XGB teve um desempenho ainda melhor se comparado com os modelos que vieram antes, porém, essa melhora se deu apenas na casa dos décimos, e devido à aproximação, não pode-se ver uma melhora tão alta. De acordo com a Tabela 6, o modelo obteve uma média de aproximadamente 4 internações para mais ou menos.

Figura 29 - Diagrama de dispersão de XGBRegressor



Fonte: Elaborado pelo autor (2025)

Tabela 6 - Métricas do modelo XGB

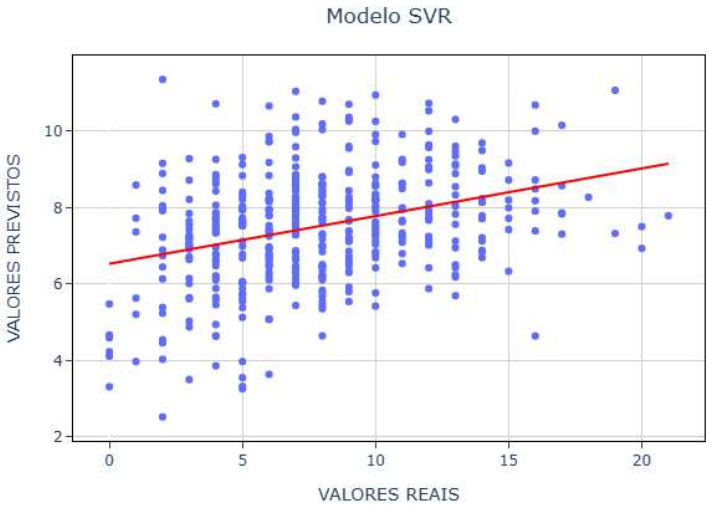
RMSE	MAE	MAPE
3.6904	2.9246	1.89E+14

Fonte: Elaborado pelo autor (2025)

4.5 SUPPORT VECTOR REGRESSION

Como é visto na Tabela 7, o modelo SVR também obteve um desempenho bom de acordo com o RMSE, tendo este sido de 4 internações para mais ou menos.

Figura 30 - Diagrama de dispersão de SVR



Fonte: Elaborado pelo autor (2025)

Tabela 7 - Métricas do modelo SVR

RMSE	MAE	MAPE
3.7547	2.9581	2.79E+14

Fonte: Elaborado pelo autor (2025)

4.6 COMPARAÇÃO COM O TRABALHO ANTERIOR

A Tabela 8 mostra os resultados deste trabalho para todos os modelos.

Tabela 8 - Métricas de todos os modelos

Modelo	RMSE	MAE	MAPE
Regressão Linear	3.8979	3.0819	3.37E+14
Árvore de Decisão	3.9959	3.1984	1.36E+14
Floresta Aleatória	3.7346	2.9713	1.93E+14
XGBRegressor	3.6904	2.9246	1.89E+14
SVR	3.7547	2.9581	2.79E+14

Fonte: Elaborado pelo autor (2025)

Já a Tabela 9, mostra os resultados obtidos pelo Cerqueira (2023).

Tabela 9 - Métricas de todos os modelos do Cerqueira (2023)

Modelo	RMSE	MAE	MAPE
Regressão Linear	14.9227	3.0487	4.31E+14
Árvore de Decisão	5.3442	4.1552	2.62E+14
Floresta Aleatória	3.2392	3.0881	2.85E+14
LSTM	3.2392	2.7203	1.50E+15

Fonte: Cerqueira (2023)

Pode-se observar a nítida melhora de desempenho que a hiperparametrização promoveu no modelo de regressão linear. Para o RMSE, tivemos uma melhora de 11 internações por dia, o indíce MAE não sofreu uma melhora tão significativa.

O modelo de árvore de decisão teve uma melhora de 2 internações no RMSE e 1 ponto a menos no MAE. A floresta aleatória foi o modelo que teve a menor das melhoras.

Os modelos SVR e XGBoost tiveram um bom desempenho, porém ficaram com métricas piores que os modelos de floresta aleatória e LSTM do Cerqueira (2023).

É possível que sejam comparados diretamente os resultados, já que o trabalho do Cerqueira (2023) foi realizado utilizando dados do mesmo período (2013 - 2018) e mesma localização (Cuiabá - MT)

5 CONCLUSÃO

O principal objetivo dessa pesquisa foi comparar e melhorar alguns métodos de aprendizado de máquina aplicados no trabalho de Cerqueira (2023). E com isso, verificar qual modelo obteve o resultado mais próximo da realidade.

Para base desta pesquisa foi utilizado o trabalho de graduação “Machine Learning para estimar o número de internações por doenças respiratórias em Cuiabá-MT” de Matheus Cerqueira de Jesus (2023). Este TCC deixou como proposta: o aumento da base de dados, o qual foi realizado ao acrescentar os dados do INMET às variáveis de entrada, o *hyperparameter boost,* que foi feito adicionando *parameters grid* aos modelos e testando diversas combinações e utilizar outros modelos de aprendizado de máquina, como XGBoost e SVR.

A realização da análise dos modelos pela métrica MAPE não funciona, já que temos muitos dias que não ocorrem internações, ou seja, acontece uma divisão por zero (0) no cálculo do mape, o que faz com que ele não seja utilizável nesse âmbito.

Utilizando como métrica o MAE, os modelos que tiveram melhor desempenho foram o XGB e o SVR, tendo uma diferença irrelevante entre os valores. Os modelos de floresta aleatória e árvore de decisão tiveram resultados bons também. Porém, a árvore de decisão obteve de longe os piores números, sendo este o pior modelo para esse trabalho.

Tomando como principal métrica o RMSE, o modelo XGB aparece novamente como um dos melhores, seguido pela floresta aleatória. A árvore de decisão se mostra novamente como um dos piores com essa métrica.

Assim, o modelo que teve o melhor resultado e conseguiu encontrar a melhor relação não-linear entre as variáveis de entrada e variáveis de saída foi o XGBRegressor.

Então, com os resultados observados e levando em consideração o problema que foi trabalhado, essa pesquisa conseguiu desempenhar bem utilizando diversos modelos. E cumprindo o proposto no objetivo, gerar dados para que a cidade de Cuiabá possa lidar melhor com os dias de altas temperaturas e alta taxa de poluição atmosférica.

5.1 PROPOSTAS DE MELHORAMENTO PARA TRABALHOS FUTUROS

- Utilizar outras variáveis de entrada para os modelos;
- Aplicar outros parâmetros na busca pela hiperparametrização;

- Ao invés da média de precipitação, somar todos os valores de precipitações para cada dia;
- Desenvolver outro método de tratamento de dados que lide melhor com dados faltantes.

REFERÊNCIAS

AHEMD, M. *et al*. A primer on large language models (LLMs) and chatgpt for cardiovascular healthcare professionals. **The Lancet Digital Health**, Londres, v. 5, n. 11, 2025. Disponível em: https://pubmed.ncbi.nlm.nih.gov/40433202/. Acesso em: 07 Nov. 2025.

BRASIL. Ministério Público do Estado de Mato Grosso. Queimadas elevam a concentração de poluição nas cidades. Mato Grosso, 2020. Disponível em: https://mpmt.mp.br/portalcao/news/731/89061/queimadas-elevam-a-concentracao-de-poluicao-nas-cidades/268. Acesso em: 07 nov. 2025.

CARVALHO, J. C. N. **Sobre o algoritmo xgboost**. [São Francisco, CA], 11 set. 2024. Disponível em: https://joaoclaudionc.medium.com/obre-o-algoritmo-xgboost-139305d1f1aa. Acesso em: 07 nov. 2025.

DUBOIS, C. Deep learning in medical image analysis: introduction to underlying principles and reviewer guide using diagnostic case studies in paediatrics. **British Medical Journal**, Londres. Disponível em: https://www.bmj.com/content/387/bmj-2023-076703. Acesso em: 30. Jan. 2025.

GÉRON, A. **Hands-on machine learning with scikit-learn, keras and tensorflow**. 3rd ed. Sebastopol, CA: O'Reilly, 2022.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **IBGE**: Cuiabá - Panorama. Rio de Janeiro: IBGE, 2022. Disponível em: https://cidades.ibge.gov.br/brasil/mt/cuiaba/panorama. Acesso em: 25 jan 2024.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **IBGE**: Cuiabá. Rio de Janeiro: IBGE, 2024. Disponível em: https://www.ibge.gov.br/cidades-e-estados/mt/cuiaba.html?. Acesso em: 25 jan. 2024.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **IBGE**: Cuiabá - Censo 2022. Rio de Janeiro: IBGE, 2022. Disponível em: https://cidades.ibge.gov.br/brasil/mt/cuiaba/pesquisa/10102/122229. Acesso em: 07 nov. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **IBGE**: Cuiabá - Censo Agropecuário 2017. Rio de Janeiro: IBGE, 2017. Disponível em: https://cidades.ibge.gov.br/brasil/mt/cuiaba/pesquisa/24/76693. Acesso em: 07 nov. 2025.

INTERNATIONAL ENERGY AGENCY. **IEA**: energy statistics data browser. Paris, França: IEA, 2024. Disponível em: https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser?country=WORLD&fuel=Energy%20supply&indicator=TESbySource. Acesso em: 15 fev. 2024.

IUNES, J. P. **Árvore de decisão**. 19 jul. 2023. Disponível em: https://colaborae.com.br/blog/2023/07/19/arvore-de-decisao/. Acesso em: 20 fev. 2024.

JESUS, M. C. de. **Machine learning para estimar o número de internações por doenças respiratórias em Cuiabá-MT**. 2023. Trabalho de Conclusão de Curso (Graduação em

Engenharia Elétrica) - Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2023. Disponível em: https://repositorio.unesp.br/entities/publication/e0248694-1d87-42ff-8586-d5348d00ce87. Acesso em: 15 dez. 2025.

KUBRUSLY, J. **Curso de machine learning -** gradiente boosting. [Boston, MA], 11 jan. 2023. Disponível em: https://bookdown.org/jessicakubrusly/curso_de_machine_learning/_book/05-floresta-aleatoria.html. Acesso em: 07 nov. 2025.

OLIVEIRA, C. J. de. **Prevendo números:** entendendo métricas de regressão. [São Francisco, CA], 12 dez. 2021. Disponível em: https://medium.com/data-hackers/prevendo-n%C3%BAmeros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70. Acesso em: 20 jan. 2024.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. **OMS**: ambient air pollution: a global assessment of exposure and burden of disease. Genebra, Suíça: OMS, 2016. Disponivel em: https://www.who.int/publications/i/item/9789241511353. Acesso em: 07 nov. 2025.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. **OMS**: global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. Genebra, Suíça: OMS, 2021. Disponível em: https://www.who.int/publications/i/item/9789240034228. Acesso em: 07 nov. 2025.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. **ONU**: temperatura média global tem 50% de chance de exceder 1.5°C até 2026. Brasília, DF: ONU, 2022. Disponível em: https://brasil.un.org/pt-br/181236-temperatura-m%C3%A9dia-global-tem-50-de-chance-de-exceder-15%C2%B0c-at%C3%A9-2026. Acesso em: 10 jan. 2024.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS. **ONU**: novas diretrizes da OMS sobre qualidade do ar reduzem valores seguros para poluição. Brasília, DF: ONU, 2021. Disponível em: https://brasil.un.org/pt-br/145721-novas-diretrizes-da-oms-sobre-qualidade-do-ar-reduzem-valores-seguros-para-polui%C3%A7%C3%A3o. Acesso em: 19 fev. 2024.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence:** a modern approach. 4th. ed. Hoboken, NJ: Pearson, 2021.

SILVA, J. **Uma breve introdução ao algoritmo de machine learning gradient boosting**. [São Francisco, CA], 22 jun. 2020. Disponível em: https://medium.com/equals-lab/uma-breve-introdu%C3%A7%C3%A3o-ao-algoritmo-de-machine-learning-gradient-boosting-utilizando-a-biblioteca-311285783099. Acesso em: 10 mar. 2024.

STANKEVIX, G. **Árvore de decisão em R**. [São Francisco, CA], 14 out. 2019. Disponível em: https://medium.com/@gabriel.stankevix/arvore-de-decis%C3%A3o-em-r-85a449b296b2. Acesso em: 21 fev. 2024.

STATE of machine learning and data science 2022. [São Francisco, CA, 2022]. Disponível em: https://www.kaggle.com/kaggle-survey-2022. Acesso em: 07 Nov. 2025.

VOHRA, K. *et al*. Global mortality from outdoor fine particle pollution generated by fossil fuel combustion: results from GEOS-Chem. **PubMed**, Maryland. Disponível em: https://pubmed.ncbi.nlm.nih.gov/33577774/. Acesso em: 07 Nov. 2025.

DADOS CURRICULARES

IDENTIFICAÇÃO	
	GIANLUCA LIMA PERINI 15/08/1998
Nacionalidade	brasileira
Nome em citações bibliográficas:	Perini, Gianluca Lima Perini, G. L.
Currículo Lattes	http://lattes.cnpq.br/9378947445453323
ORCID	https://orcid.org/0009-0004-0682-1886
PARTICIPAÇÃO EM EVENTOS CIENTÍFICOS	
1ª FASE DO XXXV CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNESP, 2024, Guaratinguetá. Comparação entre modelos de aprendizado de máquina sobre internações por doenças respiratórias em Cuiabá - MT. 2024. Congresso.	