Análise Comparativa entre Algoritmos de Aprendizado de Máquina aplicados à predição de mortalidade por COVID-19 em Salvador - Bahia

Vitor Emanuel Santos Lima
Análise e Desenvolvimento de Sistemas
Instituto Federal da Bahia
Salvador, Brasil
Email: dados.veslima3@gmail.com

Pablo Vieira Florentino
Departamento de Computação
Instituto Federal da Bahia
Salvador, Brasil
Email: pablovf@ifba.edu.br

***Resumo*—A pandemia de COVID-19 gerou um impacto global significativo, elevando a importância da identificação de fatores de risco e da predição de mortalidade entre pacientes hospitalizados. O uso de algoritmos de aprendizado de máquina se mostrou promissor nessa tarefa, auxiliando na tomada de decisões médicas. Este estudo compara o desempenho de diferentes algoritmos de aprendizado de máquina na predição de mortalidade por COVID-19, utilizando dados clínicos e socioeconômicos de Salvador-BA. Além disso, enfatiza a importância da Inteligência Artificial Explicável (XAI), permitindo a compreensão dos modelos de predição, o que facilita a confiança dos profissionais de saúde nas decisões automatizadas. O trabalho utiliza diferentes métricas para avaliar o desempenho dos modelos e traz análises dos resultados obtidos em comparação com outros trabalhos de foco semelhante. O objetivo geral é desenvolver um modelo que, além de prever a mortalidade, forneça explicações claras sobre as razões das previsões, facilitando a gestão de recursos públicos e a tomada de decisões em intervenções médicas.**

Keywords*—*COVID-19, Coronavírus, Aprendizado de máquina, Inteligência artificial, Predição de mortalidade hospitalar, IA Explicável, Saúde Coletiva.

I. Introdução

A pandemia de COVID-19, detectada pela primeira vez em dezembro de 2019 na cidade de Wuhan, na China, rapidamente se espalhou pelo mundo, resultando em milhões de mortes e sobrecarregando sistemas de saúde a nível global [1] . Desde então, identificar fatores de risco e prever a mortalidade em pacientes hospitalizados tornou-se um desafio fundamental para a medicina e a gestão de recursos em saúde coletiva [2]. Em resposta a esses desafios, o uso de algoritmos de aprendizado de máquina foi aplicado como uma abordagem promissora para prever desfechos clínicos, oferecendo suporte na tomada de decisões [3].

Pesquisas anteriores indicam que a mortalidade em pacientes com COVID-19 está associada a diversos fatores, como comorbidades pré-existentes e a gravidade dos sintomas no momento da admissão hospitalar [4]. Além disso, estudos demonstram que modelos preditivos podem identificar precocemente os pacientes com maior risco de complicações, facilitando intervenções médicas oportunas e potencialmente reduzindo o número de óbitos [5].

Este estudo tem como objetivo comparar e explicar o desempenho de diferentes algoritmos de aprendizado de máquina na predição de mortalidade por COVID-19, utilizando dados abertos, tanto socioeconômicos quanto clínicos, coletados com pacientes da rede de saúde pública em Salvador-BA [6]. Os modelos gerados a partir dos algoritmos foram analisados e os dados tratados para evitar desbalanceamento e otimizar resultados. A avaliação dos modelos foi feita através de métricas como precisão, sensibilidade e especificidade, enquanto que a validação do modelo escolhido foi conduzida utilizando técnicas como validação cruzada estratificada.

Este trabalho busca responder às seguintes questões: Qual algoritmo permite uma modelagem que oferece o melhor desempenho preditivo na análise de mortalidade por COVID-19? Quais são os principais fatores preditivos de mortalidade entre os pacientes hospitalizados com COVID-19?

O trabalho se organiza em três partes principais. Na primeira parte, é realizada uma revisão teórica sobre a engenharia de dados e sobre o aprendizado de máquina, abordando suas principais características, bem como uma introdução a conceitos chave na realização deste trabalho. A segunda parte detalha a metodologia empregada desde a revisão teórica até o tratamento dos dados, métodos de avaliação, além da análise comparativa entre os modelos preditivos. Na última parte, são discutidos os desafios, resultados, respostas ao problema e passos envolvidos no processamento, tratamento e utilização dos dados deste trabalho, bem como as técnicas e ferramentas utilizadas nas predições.

II. Fundamentação Teórica

A. Conceitos Gerais

1) Mineração de Dados: Segundo Han, Kamber e Pei, a mineração de dados é uma área interdisciplinar e pode ser definida de várias formas. Embora o termo "mineração de dados"tenha sido amplamente adotado, a expressão "Knowledge Discovery from Data"ou "descoberta de conhecimento a partir de dados"(KDD) também é utilizada para descrever esse processo. A mineração de dados refere-se à descoberta de padrões estatisticamente relevantes e conhecimento a partir de grandes volumes de dados, que podem ser originados de diversas fontes ou ainda serem transmitidos dinamicamente a um sistema [7].

2) Engenharia de Dados: A ascensão da computação na sociedade contemporânea tem propiciado avanços significativos na capacidade de geração e coleta de dados, advindos de diversas fontes. Esse fenômeno gera uma vasta quantidade de dados que permeia inúmeros aspectos da vida cotidiana, impulsionando o crescimento exponencial no armazenamento e circulação de informações. Nesse cenário, surge a necessidade de técnicas e ferramentas que possam processar grandes volumes de dados, convertendo-os em informações e conhecimento [7].

A engenharia de dados destaca-se como uma disciplina voltada para a extração, transformação e carregamento de grandes volumes de dados, além da identificação de padrões implícitos em bancos de dados, data warehouses, na web e outros repositórios ou fluxos de dados massivos [8].

Reis e Housley afirmam que, apesar da popularidade crescente da engenharia de dados, ainda não existe um consenso sobre sua definição ou sobre as atividades desempenhadas pelos profissionais da engenharia de dados. De acordo com os autores, a engenharia de dados já existia quando as empresas começaram a trabalhar com dados para análise preditiva e descritiva, ganhando destaque com a popularização da ciência de dados na década de 2010. Para eles, a engenharia de dados abrange o desenvolvimento, implementação e manutenção de sistemas e processos que transformam dados brutos em informações de alta qualidade, consistentes e prontas para consumo por cientistas de dados, analistas e outros profissionais [8].

Ainda que não haja consenso sobre a engenharia de dados como campo de estudo, neste trabalho, seguindo a visão de Reis e Housley, consideramos a engenharia de dados como uma disciplina distinta da ciência de dados, aprendizado de máquina e análise de dados. No entanto, essas áreas são complementares, e a engenharia de dados antecede essas disciplinas, fornecendo os insumos que são utilizados na entrega de valor [8].

A engenharia de dados é multidisciplinar, tendo ênfases diferentes, a depender do contexto, em áreas como segurança, governança de dados, DataOps, arquitetura de dados e engenharia de software. Essa característica multifacetada requer conhecimento de várias ferramentas, bem como a compreensão de como elas são orquestradas no ciclo de vida da engenharia de dados. Além disso, é importante entender como os dados são produzidos nos sistemas de origem e como serão consumidos por analistas e cientistas de dados após seu processamento e limpeza. Por fim, a engenharia de dados envolve a constante otimização dos sistemas de processamento de dados ao longo de eixos como custo, agilidade, escalabilidade e interoperabilidade [8].

3) Pipelines de Dados : Densmore [9] apresenta uma visão prática sobre *pipelines* de dados e sua importância para o sucesso de análises e modelos de aprendizado de máquina. A coleta de dados de diversas fontes e o processamento adequado fazem toda a diferença para extrair valor desses dados. Por trás de dashboards, modelos preditivos e ideias que influenciam a estratégia empresarial, estão dados que precisam ser limpos, processados e combinados para gerar valor. A famosa frase "dados são o novo petróleo"exemplifica essa realidade, pois, assim como o petróleo, os dados precisam ser refinados para atingir seu potencial, o que requer *pipelines* eficientes em cada etapa de sua cadeia de valor [9].

Na engenharia de dados, *pipelines* são processos que movem e transformam dados de várias fontes para um destino, de onde se extrai novo valor. Elas são a base da ciência de dados e do aprendizado de máquina [10]. *Pipelines* incluem mecanismos de preparação e análise exploratória de dados, sendo que as tarefas de preparação envolvem a integração de dados heterogêneos e sua transformação em uma representação confiável por meio de limpeza, padronização e remoção de redundâncias. As tarefas de análise exploratória, por outro lado, consistem na extração de dados para criação de visualizações, análises estatísticas e pré-processamento para modelos preditivos. O termo ETL ("Extraction, Transformation, and Loading") pressupõe o uso de *pipelines*, através das quais os dados são extraídos de várias fontes, transformados, normalizados e carregados em um banco de dados centralizado [10].

4) Aprendizado de Máquina: Géron [11] aborda de maneira prática e abrangente as técnicas e ferramentas fundamentais para o aprendizado de máquina.

O referido autor define o Aprendizado de Máquina como um subconjunto da Inteligência Artificial (IA) que se concentra no desenvolvimento de algoritmos e aplicação de modelos estatísticos que permitem que computadores executem tarefas de forma automatizada com o intuito de aprimorar seu funcionamento quanto a uma tarefa. Ao contrário da programação baseada em regras tradicionais, onde as instruções são fornecidas explicitamente para resolver um problema, os algoritmos de aprendizado de máquina aprendem com dados de forma iterativa, melhorando seu desempenho ao longo do tempo [11].

No cerne do aprendizado de máquina, o conceito de aprendizado pela experiência envolve componentes fundamentais para o sucesso dos modelos, como dados, atributos / variáveis (*fetaures*), algoritmos, treinamento, avaliação e implantação (*deploy*). Dados são o ponto de partida para qualquer processo de aprendizado de máquina, abrangendo observações, medições e exemplos. Esses dados podem ser estruturados, semi-estruturados ou não estruturados e a qualidade e representatividade deles têm um papel crucial no desempenho dos modelos que serão testados [11].

Além disso, os atributos, ou variáveis de entrada, são características extraídas dos dados para previsões. A seleção e engenharia de atributos (*feature engineering*) adequados são essenciais para o sucesso de um modelo preditivo, uma vez que permitem identificar quais variáveis são mais relevantes para a tarefa em questão. O modelo, por sua vez, é uma representação matemática das relações entre os atributos e as previsões de saída. A escolha dos algoritmos varia conforme o problema e eles são modelados e treinados para aprender padrões subjacentes nos dados [11].

O treinamento envolve ajustar os parâmetros do modelo com base nos dados de treinamento, buscando minimizar o erro entre as previsões e os resultados reais. Após o treinamento, o modelo passa por uma fase de avaliação, na qual seu desempenho é medido com métricas como precisão e acurácia, além de ser submetido a técnicas como a validação cruzada, que é usada para evitar viés e inspecionar o grau de variância nos resultados da avaliação, garantindo assim a capacidade de generalização do modelo para novos dados [11].

A última etapa, a implantação ou *deploy*, refere-se à integração do modelo treinado em ambiente de produção, permitindo que ele faça previsões sobre dados não vistos [11]. No entanto, neste trabalho, não haverá uma etapa de *deploy*, visto que o objetivo deste trabalho não é criar um produto final, como um sistema de recomendação.

O aprendizado de máquina abrange uma ampla gama de técnicas e metodologias, incluindo aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado, aprendizado por reforço e aprendizado profundo. Essas técnicas permitem que máquinas reconheçam padrões, façam previsões, agrupem dados, descubram estruturas ocultas e otimizem decisões em diversos domínios e aplicações, que vão desde reconhecimento de imagem e processamento de linguagem natural até direção autônoma e sistemas de recomendação personalizados [11].

O aprendizado de máquina oferece uma abordagem eficaz para resolver diversos problemas com os quais métodos tradicionais de programação não conseguem lidar adequadamente. Por exemplo, em tarefas que envolvem padrões complexos, como o reconhecimento óptico de caracteres (OCR), pois a variabilidade nos estilos de escrita manuscrita torna inviável a criação de regras explícitas para distinguir os caracteres. Nesse contexto, algoritmos de aprendizado de máquina, como redes neurais convolucionais (CNNs), destacam-se por sua capacidade de aprender automaticamente representações hierárquicas a partir dos dados [12].

Além disso, o aprendizado de máquina é essencial para problemas sem solução algorítmica clara, como a tradução de idiomas e a análise de sentimentos em redes sociais, onde o contexto e a semântica desempenham papéis cruciais. Modelos como redes neurais recorrentes (RNNs) e transformadores são utilizados para capturar dependências temporais e contextuais em dados sequenciais [12]. Esses modelos permitem traduções mais precisas e classificações de sentimentos com base em grandes volumes de texto.

Em ambientes dinâmicos, como sistemas de detecção de fraude, a adaptabilidade dos modelos de aprendizado de máquina, especialmente aqueles com capacidade de aprendizado incremental, é crucial para o modelo se ajustar rapidamente a novos padrões de fraude [12]. Assim, a predição pode continuar eficaz mesmo quando as dinâmicas dos dados evoluem ao longo do tempo.

O aprendizado de máquina permite a análise de grandes e complexos volumes de dados, como registros de saúde e transações financeiras, possibilitando a extração de percepções valiosas por meio de diferentes técnicas [12]. Ao longo deste trabalho, serão enfatizadas as técnicas e metodologias relevantes para cumprir os objetivos propostos.

5) Aprendizado de Máquina Supervisionado: O aprendizado de máquina supervisionado é uma técnica central no campo do aprendizado de máquina, onde um modelo é treinado a partir de um conjunto de dados rotulados, que contêm exemplos de entradas (variáveis independentes) associadas às saídas corretas (variáveis dependentes), permitindo que o algoritmo aprenda padrões e associações entre essas variáveis [13]. O objetivo principal dessa abordagem é fazer com que o modelo seja capaz de generalizar a relação entre os dados de entrada e suas respectivas saídas, de modo a realizar previsões precisas quando novos dados, não observados anteriormente, forem apresentados [14].

Durante a fase de treinamento, o modelo ajusta seus parâmetros internos de forma iterativa, minimizando o erro entre suas previsões e os rótulos corretos dos dados de treinamento [15]. Isso é feito por meio de uma função de perda, que mede a discrepância entre a predição do modelo e a saída esperada. À medida que o processo de treinamento avança, o modelo melhora sua capacidade de prever corretamente com base nos exemplos fornecidos.

O aprendizado supervisionado pode ser aplicado a dois grandes tipos de problemas: classificação e regressão. Em problemas de classificação, o modelo é usado para prever a categoria ou classe de um dado, como, por exemplo, identificar se um e-mail é spam ou não [12]. Já nos problemas de regressão, o objetivo é prever um valor contínuo, como estimar o preço de uma casa com base em suas características físicas e localização [16]

Um dos maiores desafios no aprendizado supervisionado é garantir que o modelo tenha uma boa capacidade de generalização, ou seja, que ele consiga aplicar o conhecimento adquirido a partir dos dados de treinamento para fazer previsões corretas sobre novos dados [14].

Para evitar que o modelo memorize os dados de treinamento, uma consequência conhecida como *overfitting*, são aplicadas diversas técnicas de regularização para promover um bom equilíbrio entre a complexidade do modelo e sua capacidade preditiva [15].

Entre os algoritmos comumente utilizados no aprendizado supervisionado, destacam-se a regressão linear, que é amplamente usada para prever variáveis contínuas [16] e as árvores de decisão, que facilitam a interpretação dos processos de classificação ao dividir iterativamente os dados com base em características relevantes [17]. Outros tipos de algoritmos comumente usados incluem as máquinas de vetores de suporte (SVM), que são eficazes para problemas de classificação com alta dimensionalidade, e as redes neurais artificiais, que são capazes de capturar padrões complexos, tanto para tarefas de classificação quanto de regressão [14].

Assim, o aprendizado supervisionado desempenha um papel fundamental em diversas aplicações, como o reconhecimento de padrões em imagens, a predição de preços em mercados imobiliários e o reconhecimento automático de fala [13].

6) Classificação Binária: A classificação binária é uma técnica de aprendizado de máquina na qual um modelo é treinado para classificar instâncias em um de dois possíveis grupos ou classes. O objetivo é prever a qual categoria um determinado conjunto de entradas pertence. Esses dois grupos são frequentemente referidos como "classe positiva"e "classe negativa"[13].

Na classificação binária, o modelo aprende a partir de um conjunto de dados de treinamento com exemplos rotulados, ou seja, onde cada exemplo já foi classificado corretamente, e tenta identificar padrões nos dados que permitam a correta atribuição de novas observações [14].

As duas categorias de saída, são comumente rotuladas como 0 e 1, ou como "negativo"e "positivo": o modelo pode prever a probabilidade de uma instância pertencer à classe positiva ou negativa. Em muitas abordagens, como a regressão logística, a saída é uma probabilidade que, se maior que um limiar específico, resulta na previsão da classe positiva. Para avaliar o desempenho de um modelo de classificação binária, utilizam-se métricas como acurácia, precisão, *recall*, especificidade, AUC-ROC e outras [15].

7) Inteligência Artificial Explicável: A Inteligência Artificial Explicável (XAI) refere-se a uma subárea da inteligência artificial que busca desenvolver modelos cujas decisões possam ser compreendidas e interpretadas por seres humanos. O objetivo principal da XAI é garantir que os modelos de IA, especialmente os mais complexos, sejam transparentes e compreensíveis para os usuários. Isso é fundamental para aumentar a confiança, facilitar a adoção de sistemas de IA e garantir que os modelos possam ser auditados e responsabilizados, especialmente em contextos críticos como a saúde [18].

Uma das grandes vantagens da XAI é permitir que os profissionais compreendam o raciocínio por trás das decisões tomadas por modelos de aprendizado de máquina. Isso é particularmente importante quando esses modelos são usados em setores como a medicina, onde decisões automatizadas podem afetar diretamente a vida dos pacientes. O desenvolvimento de técnicas explicáveis, como a visualização dos processos de decisão e a simplificação dos modelos, possibilita que as previsões de IA sejam mais acessíveis e compreendidas pelos especialistas humanos, sem sacrificar a precisão [19].

De acordo com Melanie Mitchell, é crucial que os pesquisadores e profissionais de IA se empenhem em tornar os modelos mais interpretáveis, garantindo que seus impactos sociais e econômicos sejam compreendidos e monitorados [20].

8) Saúde Coletiva: De acordo com Souza [21], a saúde coletiva se diferencia da Saúde Pública tradicional ao propor uma abordagem mais ampla e crítica. Enquanto a saúde pública está mais centrada em diagnósticos, tratamentos e controle de doenças, a saúde coletiva busca compreender os determinantes sociais, econômicos e culturais que influenciam os problemas de saúde. Essa abordagem considera que as condições de saúde de uma população são moldadas por fatores sociais estruturais, como desigualdade econômica, políticas públicas e participação social. Dessa forma, a saúde coletiva visa integrar aspectos biomédicos e sociais, promovendo a participação ativa da sociedade na formulação e execução de políticas de saúde [22], [23].

9) Vigilância Epidemiológica: Conforme Albuquerque [24], a vigilância epidemiológica é definida como um conjunto de atividades que envolvem a coleta, análise e interpretação sistemática de dados relacionados à ocorrência de doenças e outros agravos à saúde. O objetivo principal da vigilância epidemiológica é fornecer subsídios técnicos e práticos para a tomada de decisões no controle de doenças, permitindo a detecção precoce de surtos e a implementação de ações de prevenção e controle. No contexto brasileiro, a vigilância epidemiológica tem sido historicamente institucionalizada como uma ferramenta central para o planejamento e a execução de políticas de saúde pública, sendo essencial para proteger a população de riscos à saúde [23].

B. Algoritmos de Aprendizado de Máquina

1) Regressão Logística: A Regressão Logística é um algoritmo utilizado na modelagem de variáveis categóricas binárias, estimando a probabilidade de uma instância pertencer a uma classe específica com base nas características de entrada. Seu nome deriva da função logística, que modela essa probabilidade. O objetivo principal ao aplicar a Regressão Logística é prever a classe com base em uma combinação linear das variáveis de entrada, seguida pela aplicação da função sigmoide, que transforma o resultado linear em uma probabilidade[15].

A fórmula apresentada na figura 1 (adaptada de [25]) representa a função logística, que calcula a probabilidade P(x) de um evento ocorrer, dada uma variável preditora x. Nesta equação, b0 representa o intercepto do modelo, enquanto b1 é o coeficiente que mede a influência da variável x. A combinação linear b0+b1x é transformada pela base e, de modo que o resultado final seja limitado ao intervalo entre 0 e 1, garantindo a interpretação como uma probabilidade. O termo b0+b1x no denominador realiza a normalização da fórmula, tornando-a apropriada para modelagem de variáveis binárias. Essa transformação permite prever e classificar eventos de forma eficiente, como no caso da identificação de e-mails como spam (P(x) > 0,5 ou não-spam P(x) <= 0,5. Esse processo de decisão binária é a essência da Regressão Logística [16].

$$P(x) = e^{b0+b1x}/(1 + e^{b0+b1x})$$

Figura 1. Regressão Logística.

2) Árvores de Decisão: Árvores de Decisão (*Decision Trees*) é um algoritmo de aprendizado supervisionado que utiliza uma estrutura em forma de árvore para tomar decisões. Esse algoritmo divide o espaço de características dos dados de maneira recursiva, gerando "nós"internos que representam decisões baseadas em uma única variável e "folhas"que representam a predição final [17]. A principal vantagem das Árvores de Decisão é a simplicidade de interpretação: é possível seguir o caminho da raiz até uma folha para ver como uma decisão foi tomada [26].

A fórmula matemática da probabilidade condicional, que representa a probabilidade de uma classe Ci dada uma entrada *x* e um nó *m* da árvore, é mostrada na figura 2 (adaptada de [27]).

$$\hat{P}(C_i|\mathbf{x}, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$

Figura 2. Árvore de Decisão.

Essa probabilidade é calculada como a razão entre Nim, que é o número de amostras no nó m pertencentes à classe Ci,

e Nm, que representa o número total de amostras no mesmo nó m.

Essa relação expressa a fração de amostras de uma classe específica em relação ao total de amostras do nó analisado, permitindo que o modelo atribua probabilidades às classes de maneira direta e interpretável [17].

Por exemplo, num cenário onde se deseja prever se um e-mail é spam ou não, a Árvore de Decisão poderá dividir os dados com base em variáveis numéricas que representam propriedades textuais dos e-mails, como frequência de palavras, frequência de caracteres e ocorrência de caracteres especiais, criando regras simples que facilitam a interpretação das decisões [15].

3) Florestas Aleatórias: Florestas Aleatórias (*Random Forest*) é um algoritmo de aprendizado de máquina, baseado no algoritmo de Árvores de Decisão, que combina previsões probabilísticas de várias árvores independentes. Ao calcular a média das probabilidades preditivas e selecionar a classe com maior valor médio, o modelo reduz o impacto de erros individuais de cada árvore, o que contribui para sua confiabilidade e maior capacidade de generalização [28]. O Random Forest é aplicado em problemas de classificação e regressão, sendo eficaz para evitar o overfitting, comum em Árvores de Decisão individuais [28].

A fórmula da figura 3 (adaptada de [25]) descreve o processo de agregação das previsões individuais das árvores de decisão em um modelo de Floresta Aleatória. Nessa fórmula, Z representa a classe final predita pelo modelo, a qual é obtida utilizando o operador *argmax*. Esse operador retorna a classe y que maximiza a média das probabilidades preditivas estimadas pelas árvores da floresta [28].

O termo 1/T é um fator de normalização que divide a soma das probabilidades preditivas pelo número total de árvores T na floresta, garantindo que a média seja calculada corretamente. A soma T Σ t=1 computa a soma das probabilidades preditivas pt(y/x) para a classe y em todas as árvores t. Aqui, pt(y/x) é a probabilidade estimada pela t-ésima árvore para a classe y, dado o conjunto de características x da entrada [28].

$$Z = argmax \frac{1}{T} \sum_{t=1}^{T} p_t(y/x)$$

Figura 3. Random Forest.

4) Support Vector Classifier: A Support Vector Classifier (SVC) é uma implementação prática do algoritmo de Máquinas de Vetores de Suporte (SVM) que visa a maximização da margem entre as classes [29].

$$f(x) = \sum_{i=1}^{n} c_i y_i \varphi(x_i)$$

Figura 4. Support Vector Classifier. Adaptada de [25]

O modelo busca um hiperplano definido pela equação w * x + b = 0, onde w é o vetor de pesos que determina a orientação do hiperplano, x representa os vetores de entrada, e b é o termo de viés que ajusta a posição do hiperplano no espaço [29].

O objetivo do SVC é encontrar w e b de forma que a margem, definida como a distância entre os vetores de suporte mais próximos ao hiperplano, seja maximizada. A maximização dessa margem equivale a minimizar $\|w\|^2$ (o quadrado da norma do vetor), sujeito às restrições yi (w * xi + b) >= 1, onde yi é o rótulo da classe associada ao vetor de entrada xi e assume valores +1 ou -1. Essas restrições garantem que os exemplos de cada classe sejam corretamente classificados e fiquem fora ou na borda do hiperplano, respeitando a margem[29].

Quando os dados não são linearmente separáveis, o SVC introduz uma variável de relaxação ξi para cada exemplo, permitindo que alguns pontos violem as margens, mas com penalidade proporcional à soma Σ ξi. O termo de regularização C controla o equilíbrio entre maximizar a margem e minimizar os erros de classificação. Assim, a fórmula otimizada para o SVC combina a minimização de $\|w\|^2$ com a penalização dos erros, garantindo que o modelo generalize bem mesmo em casos de sobreposição parcial entre as classes [29].

5) Gradient Boosting: O *Gradient Boosting* é um algoritmo de aprendizado de máquina que constrói modelos de forma sequencial, onde cada novo modelo busca corrigir os erros cometidos pelos modelos anteriores. Ao contrário do Random Forest, em que as árvores são construídas de forma independente, no Gradient Boosting cada nova árvore é treinada para minimizar os resíduos dos modelos já existentes, aprimorando progressivamente o desempenho geral [30].

A fórmula apresentada na figura 5 (adaptada de [30]), ilustra como a função F(x) pode ser decomposta em componentes que descrevem os efeitos individuais e interações entre as variáveis em um algoritmo de Gradient Boosting. Cada termo dessa decomposição é representado por uma soma que considera diferentes níveis de interação entre os atributos.

$$F(\mathbf{x}) = \sum_{j} f_j(x_j) + \sum_{j,k} f_{jk}(x_j, x_k) + \sum_{j,k,l} f_{jkl}(x_j, x_k, x_l) + \cdots.$$

Figura 5. Gradient Boosting

Essa abordagem iterativa permite criar modelos mais precisos ao ajustar continuamente as previsões, proporcionando um bom desempenho em diversos cenários [31].

O primeiro termo, Σj fj(xj), descreve os efeitos individuais de cada variável xj na função F(x). Em outras palavras, ele captura como cada atributo, isoladamente, contribui para o valor da função de previsão [30].

O segundo termo, Σj, k fjk(xj, xk), introduz os efeitos de interação entre pares de variáveis. Aqui, a influência conjunta de dois atributos xj e xk sobre a função F(x) é modelada, permitindo capturar relações não lineares entre eles [30].

O terceiro termo, Σj, k , l fjkl(xj, xk, xl), estende essa ideia para interações entre três variáveis, avaliando como os

três atributos impactam conjuntamente a função de previsão [30].

Esse padrão continua com termos adicionais que representam interações de ordens superiores, como as que envolvem quatro ou mais variáveis. No entanto, na prática, termos de interação de ordem mais alta frequentemente contribuem menos para o desempenho do modelo e podem ser ignorados para simplificação [30].

A decomposição apresentada reflete a capacidade do Gradient Boosting de capturar tanto os efeitos diretos das variáveis quanto suas interações, resultando em um modelo mais expressivo [30]. Essa abordagem é especialmente útil em situações onde as relações entre as variáveis não são triviais, permitindo que o modelo se adapte a padrões complexos nos dados [30].

6) Naive Bayes: O Naive Bayes é um algoritmo que funciona como um classificador probabilístico baseado no Teorema de Bayes, que assume a independência condicional entre as variáveis preditoras. Apesar dessa suposição simplificadora, o Naive Bayes frequentemente apresenta um desempenho robusto em diversas tarefas de classificação, notadamente em áreas como processamento de linguagem natural e filtragem de spam [13]. Seu princípio fundamental consiste em calcular a probabilidade de uma instância pertencer a uma determinada classe com base nas características observadas, permitindo classificações rápidas e eficientes, mesmo em cenários com dados limitados ou ruidosos [32].

Essa probabilidade pode ser calculada utilizando a fórmula apresentada abaixo, que aplica o Teorema de Bayes considerando as probabilidades condicionais de cada característica observada, onde p(x|C) representa a probabilidade condicional de um vetor de atributos x dado que a classe C foi observada. O vetor x consiste nas variáveis preditoras x1, x2, ... , xd, onde cada xj é uma característica específica [13].

A expressão p(xj|C) indica a probabilidade condicional de xj dado C, ou seja, a probabilidade de observar o valor da característica xj sabendo que a instância pertence à classe C. O produto d Π j=1 percorre todas as d características, multiplicando suas probabilidades individuais, de acordo com a suposição de independência condicional. Essa simplificação permite que o modelo trate as variáveis como independentes umas das outras ao calcular a probabilidade conjunta, facilitando a implementação e reduzindo a complexidade computacional [13].

$$p(\mathbf{x}|C) = \prod_{j=1}^{d} p(x_j|C)$$

Figura 6. Naive Bayes. Adaptada de [27]

7) XGBoost: O XGBoost (*eXtreme Gradient Boosting*) é um algoritmo de aprendizado de máquina baseado no Gradient Boosting e projetado para ser altamente eficiente em termos de tempo e uso de recursos computacionais [33]. Entre suas principais inovações, destacam-se a regularização adicional para melhorar a generalização do modelo, paralelização intrínseca que acelera o treinamento, suporte a processamento distribuído e a capacidade de lidar eficientemente com dados ausentes.

$$F(x_i) = \sum_{t=1}^{T} F_t(x_i)$$

Figura 7. XGBoost. Adaptada de [25]

Na fórmula apresentada na figura 8 , F(xi) denota a predição agregada para a instância xi. O índice t percorre o intervalo de 1 até T, onde T representa o número total de árvores de decisão no modelo. A função Ft(xi) refere-se à predição gerada pela t-ésima árvore para a mesma instância xi. A operação de soma T Σ t=1 combina as predições de todas as árvores para fornecer a previsão final, seguindo a abordagem de boosting, onde modelos simples são iterativamente ajustados para minimizar o erro residual da previsão anterior. Essa combinação gradual e ponderada de modelos permite capturar tanto padrões lineares quanto não-lineares, resultando em um modelo robusto e preciso [33].

Essa estrutura de combinação em etapas, juntamente com o controle de overfitting e algoritmos de redução de viés e variância, faz do XGBoost uma ferramenta padrão tanto em competições de ciência de dados quanto em sistemas de produção, sendo reconhecido por seu alto desempenho e flexibilidade em diversos cenários [11].

8) CatBoost: O CatBoost (*Categorical Boosting*) é um algoritmo de Gradient Boosting que se destaca por seu tratamento nativo de variáveis categóricas e resistência a *overfitting* [34].

$$h(\mathbf{x}) = \sum_{j=1}^{J} b_j \mathbf{1}_{\{\mathbf{x} \in R_j\}},$$

Figura 8. CatBoost.

Como ilustrado na figura 8, a função indicadora retorna 1 se a instância x pertence à região Rj e 0 caso contrário, bj representa o valor de predição associado à j-ésima folha da árvore, e J é o número total de folhas (regiões finais) na árvore [34], [35]. Ou seja, o CatBoost constrói seu modelo através de uma combinação aditiva de árvores de decisão binárias, onde cada árvore h(x) divide o espaço de características em J regiões separadas Rj, conforme expresso acima [34], [35].

O CatBoost transforma cada categoria em valores numéricos através de estatísticas calculadas sobre subconjuntos ordenados dos dados [34]. Ele incorpora ainda técnicas de regularização como coeficientes L2, taxa de aprendizado adaptativa, e um esquema de amostragem de instâncias e características que melhora o modelo gradativamente [35]. Estas características fazem do CatBoost uma escolha para conjuntos de dados com alta dimensionalidade categórica, desbalanceamento de classes, presença de ruído e valores ausentes [34], [35].

C. Conceitos Específicos

1) Função de perda: A função de perda (ou custo) é uma métrica essencial no aprendizado de máquina supervisionado, usada para quantificar a discrepância entre as previsões do modelo e os valores reais. O objetivo é minimizar essa função, o que melhora a capacidade de generalização e a precisão preditiva [14]. A escolha da função de perda varia conforme a natureza do problema.

Entre as mais comuns, o Erro Quadrático Médio (Mean Squared Error) é amplamente usado em regressão, medindo o quadrado da diferença entre previsões e valores reais, o que o torna sensível a grandes erros [36]. Já a Entropia Cruzada (*Cross-Entropy Loss*) é frequente em problemas de classificação, medindo a discrepância entre a distribuição real das classes e as previsões probabilísticas [15]. O Erro Absoluto Médio (MAE), por sua vez, é menos sensível a *outliers* e usado em regressão [16]. Para classificação binária, a Hinge Loss é comum, como no caso de SVMs, por maximizar a margem entre classes [29].

Neste trabalho, os modelos utilizam funções de perda internas. A Regressão Logística usa a log-loss , que penaliza previsões probabilísticas incorretas [15]. Modelos feitos com Árvores de Decisão e Random Forests utilizam métricas como a Entropia para medir a pureza dos nós [28], enquanto o modelo baseado em Gradient Boosting otimiza diretamente uma função de perda, como log-loss ou MSE (mean square error ou erro quadrático médio) [30].

Quanto ao modelo que usa o XGBoost, ele também usa a log-loss em problemas de classificação [15], enquanto a SVC usa a Hinge Loss [29] e o Naive Bayes maximiza a verossimilhança condicional [13].

Para problemas de classificação binária, a função de perda mais adequada é geralmente a log-loss, que penaliza previsões incorretas e é adequada para modelos que utilizam probabilidades (Regressão Logística, Gradient Boosting e XGBoost, por exemplo), ajudando a calibrar as previsões e melhorar o desempenho geral [15].

2) Overfitting: O *overfitting*, ou sobreajuste, ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, capturando tanto os padrões gerais quanto ruídos e *outliers* presentes no conjunto de dados. Isso faz com que o modelo apresente um desempenho excelente no conjunto de treinamento, mas falhe ao ser testado com novos dados, pois não consegue generalizar adequadamente [14]. Goodfellow, Bengio e Courville destacam que o overfitting é mais comum em modelos complexos, como redes neurais profundas, onde a capacidade do modelo de memorizar detalhes específicos do conjunto de treinamento é alta [14]. Um exemplo típico de overfitting ocorre quando o modelo "memoriza"os exemplos de treinamento, em vez de aprender padrões que podem ser aplicados a novos dados. Esse problema pode ser especialmente recorrente em conjuntos de dados pequenos ou ruidosos, nos quais a quantidade de informações genuínas é limitada. Estratégias comuns para mitigar o overfitting incluem o uso de técnicas de regularização, além da utilização de validação cruzada para saber se há necessidade de ajustar o modelo [11].

3) Underfitting: O *underfitting,* ou sobajuste*,* ocorre quando o modelo é muito simples ou inflexível para capturar os padrões subjacentes dos dados. Quando um modelo sofre de *underfitting*, ele falha em aprender a relação subjacente entre as variáveis de entrada e de saída, resultando em um desempenho insatisfatório tanto nos dados de treinamento quanto nos de teste [11]. Géron observa que o *underfitting* é comum em modelos muito restritos, como a regressão linear simples, quando usada para capturar padrões complexos. Esse problema pode ser causado por escolhas inadequadas de algoritmos de aprendizado de máquina ou pela falta de ajuste dos hiperparâmetros. Para mitigar o *underfitting*, é possível aumentar a complexidade do modelo, incluir mais variáveis ou ajustar hiperparâmetros como a profundidade no caso das árvores de decisão [11].

4) Viés: O viés é uma medida da capacidade de um modelo de simplificar demais o problema que está tentando resolver. Modelos com alto viés tendem a fazer suposições muito fortes sobre os dados e, como resultado, não conseguem capturar as complexidades do conjunto de dados, resultando em *underfitting* [15]. De acordo com Hastie, Tibshirani e Friedman, modelos com alto viés apresentam uma tendência a subestimar a variabilidade dos dados e a ignorar características importantes, o que os impede de capturar os padrões mais sutis nos dados de treinamento. Um exemplo de alto viés seria o uso de um modelo de regressão linear para um problema que envolve uma relação não linear. Nesse caso, o modelo não conseguirá representar adequadamente os dados, e seu desempenho será fraco, tanto nos dados de treinamento quanto nos dados de teste [15].

5) Variância: A variância de um modelo refere-se à sensibilidade que ele apresenta a pequenas variações no conjunto de dados de treinamento. Um modelo com alta variância é excessivamente dependente dos dados de treinamento e, portanto, tende a apresentar grandes flutuações no desempenho quando aplicado a diferentes subconjuntos de dados [16]. Isso significa que o modelo pode apresentar um desempenho excelente em alguns conjuntos de treinamento, mas falha em outros, caracterizando o problema de *overfitting*. James *et al.* afirmam que modelos com alta variância geralmente são excessivamente complexos, capturando ruídos e variações irrelevantes no conjunto de dados de treinamento, o que prejudica sua capacidade de generalização [16]. Para lidar com alta variância, técnicas como a validação cruzada e o uso de métodos de regularização, como *ridge regression* ou *lasso (que serão explicadas na parte sobre regularização)*, podem ser aplicadas para reduzir a complexidade do modelo sem sacrificar sua capacidade preditiva [16].

6) Desvio Padrão: O desvio padrão é uma medida estatística que quantifica a variação dos dados em torno da média. Um desvio padrão baixo indica que os valores estão próximos da média, sugerindo menor variabilidade e mais consistência nos dados; já um desvio padrão alto indica uma maior dispersão dos valores [16] [15]. Essa medida é essencial para avaliar a estabilidade de um conjunto de dados e o desempenho de modelos, especialmente na análise de erros, onde altos desvios podem indicar problemas de *overfitting* ou *underfitting* [16] [15].

O desvio padrão também é usado na engenharia de atributos (*feature engineering*), ajudando a padronizar variáveis e facilitar o treinamento de algoritmos. A normalização, que envolve subtrair a média e dividir pelo desvio padrão, é uma

prática comum para alinhar as escalas das variáveis e melhorar a interpretação dos dados pelo modelo [11].

7) Generalização: No contexto de aprendizado de máquina, a generalização refere-se à capacidade de um modelo de ter bom desempenho com dados não vistos durante o treinamento [11]. Um modelo é considerado bom em generalização quando consegue aplicar o conhecimento adquirido em um conjunto de dados de treinamento para fazer previsões precisas em novos dados, ou seja, dados fora da amostragem. A generalização depende de vários fatores, como a qualidade dos dados de treinamento, a complexidade do modelo e a técnica de regularização empregada [12].

Quando um modelo tem uma boa capacidade de generalização, ele evita dois problemas comuns mencionados anteriormente: overfitting e underfitting [11]. A avaliação da capacidade de generalização de um modelo é feita por meio de métricas de desempenho em conjuntos de dados de validação e teste [11].

8) Regularização: Regularização é uma técnica usada em modelos de aprendizado de máquina para evitar o overfitting. A regularização age adicionando um termo de penalidade à função de custo, que limita a magnitude dos coeficientes do modelo. Isso incentiva o modelo a buscar soluções mais simples, resultando em melhor desempenho em novos conjuntos de dados [11].

Existem duas formas principais de regularização. A primeira é a regularização L1, também conhecida como *Lasso*, que adiciona a soma dos valores absolutos dos coeficientes à função de custo [16]. Essa abordagem tende a gerar modelos esparsos, eliminando coeficientes de variáveis que são irrelevantes para o problema. A segunda forma é a regularização L2, chamada de *Ridge*, que adiciona a soma dos quadrados dos coeficientes à função de custo. Embora não elimine variáveis, a regularização L2 reduz a magnitude dos coeficientes, evitando que o modelo dependa demais de características específicas, o que é útil em cenários com dados ruidosos [16].

9) Entropia: Em aprendizado de máquina, a entropia, mede a incerteza ou desordem de um conjunto de dados ou de uma variável aleatória, especialmente em algoritmos de árvore de decisão. A entropia é usada para quantificar a impureza dos nós (ou seja, a mistura de diferentes classes em um nó) [37]. Ela ajuda a decidir onde dividir os dados de forma a reduzir a incerteza e a aumentar a homogeneidade dos grupos criados [38].

Reduzir a entropia em nós sucessivos ajuda a melhorar a precisão e a generalização do modelo, tornando-o menos propenso a problemas de overfitting ou underfitting [38].

10)Validação Cruzada: A validação cruzada é uma técnica bastante utilizada para avaliar a capacidade de generalização de um modelo. O processo envolve dividir o conjunto de dados em várias partes (ou *folds*) e, em cada iteração, utilizar uma parte para treinar o modelo e outra para testar. Esse método ajuda a evitar o *overfitting*, oferecendo uma avaliação sólida do desempenho do modelo ao usar diferentes porções dos dados para validação [39]. Kohavi [39] ressalta que métodos como o K-Fold Cross-Validation, que divide os dados em K subconjuntos, permitem que cada instância dos dados seja utilizada tanto para treinamento quanto para teste, proporcionando uma estimativa mais confiável do desempenho do modelo em dados não vistos. A validação cruzada é especialmente valiosa em situações onde a quantidade de dados é limitada, pois maximiza o uso eficiente das informações disponíveis [39].

11) Análise de Dispersão: A análise de dispersão dos resultados na validação cruzada é um procedimento estatístico importante para avaliar a estabilidade e a confiabilidade de um modelo preditivo. Essa análise envolve o exame da variação nos resultados de diferentes execuções de validação cruzada, onde o desempenho do modelo é medido em várias partições do conjunto de dados [40]. De acordo com Kuhn e Johnson [40], o objetivo principal é verificar se o modelo é robusto e consistente, ou se é excessivamente sensível a pequenas mudanças nos dados de treinamento, o que poderia indicar problemas como *overfitting* ou *underfitting*.

Na prática, a validação cruzada divide o conjunto de dados em subconjuntos chamados *folds*, e treina o modelo em diferentes combinações desses *folds*. A análise de dispersão foca em entender a variação dos resultados, como acurácia, precisão e recall, entre essas iterações [40]. Kuhn e Johnson explicam que a dispersão é calculada, geralmente, por métricas como desvio padrão ou intervalo de confiança em torno da métrica de desempenho média.

Uma pequena variação entre os *folds* sugere que o modelo é estável e generaliza bem para novos dados. Segundo Kuhn e Johnson, isso significa que o modelo está capturando padrões relevantes nos dados e não é excessivamente dependente de exemplos específicos. Um desvio padrão baixo reflete que o modelo tem um desempenho consistente em diferentes subconjuntos dos dados [40].

Por outro lado, uma grande variação nos resultados indica alta variância do modelo, sugerindo que ele está superajustado aos dados de treinamento [40]. Kuhn e Johnson destacam que isso pode ocorrer especialmente em conjuntos de dados pequenos ou desbalanceados, onde os exemplos em cada *fold* variam significativamente.

A análise de dispersão, portanto, é crucial para a seleção adequada do modelo, bem como para o ajuste de hiperparâmetros, e pode auxiliar na identificação de problemas de generalização do modelo [40].

12)Amostragem: A amostragem é um processo estatístico que visa selecionar uma parte representativa de uma população para análise [41]. Esse procedimento é implementado em pesquisas estatísticas, permitindo que conclusões sejam tiradas sobre a população inteira sem a necessidade de observar ou medir cada indivíduo [41]. A escolha de uma amostra adequada é crucial para garantir que os resultados sejam válidos e generalizáveis [41]. Existem várias técnicas de amostragem, como amostragem aleatória simples, estratificada, por conglomerados e sistemática, cada uma com suas próprias vantagens e inadequações, dependendo do tipo de estudo e das características da população [41].

A amostragem é essencial quando se trabalha com grandes volumes de dados, permitindo a realização de análises e treinamentos de modelos com menos custos computacionais [15]. Quando bem executada, a amostragem mantém a integridade dos padrões e características dos dados originais, garantindo a confiabilidade das análises subsequentes [15].

D. Ferramentas e Técnicas Utilizadas

1) Principal Component Analysis: A análise de componentes principais (PCA) é uma técnica de redução de dimensionalidade utilizada em aprendizado de máquina e estatística, particularmente quando a simplificação de conjuntos de dados complexos e de alta dimensão é necessária[42]. Em contextos em que o volume de dados é muito grande, a alta dimensionalidade pode ser um fator que dificulta o desempenho dos modelos de aprendizado, pois muitas variáveis interrelacionadas podem introduzir ruído e redundância [15].

Conforme descrito por Jolliffe, o PCA transforma os dados ao projetá-los em um novo sistema de coordenadas, no qual cada eixo representa um componente principal. Esses componentes são combinações lineares das variáveis originais, ordenados de forma que o primeiro componente capture a maior parte da variância dos dados [42]. O segundo componente captura a segunda maior parte, e assim por diante. A ordenação dos componentes facilita a escolha de um subconjunto que mantém a maior parte da informação relevante, reduzindo a complexidade dos dados sem perda significativa de informações [43].

Ao reduzir a dimensionalidade dos dados enquanto preserva o máximo de variância, o PCA facilita a visualização e interpretação [16]. Além disso, o PCA pode reduzir ruído, eliminando variações menos significativas, e mitigar a multicolinearidade (correlação excessiva entre as variáveis do modelo utilizado), transformando variáveis originais em um conjunto não correlacionado [15]. Essa transformação melhora a estabilidade dos modelos de aprendizado de máquina e facilita a interpretação dos resultados [36].

Apesar das vantagens, o PCA apresenta limitações. Os componentes principais, sendo combinações de variáveis originais, podem ser difíceis de interpretar de maneira prática [42]. Além disso, o PCA assume que os dados seguem uma estrutura linear, o que pode não se adequar a conjuntos de dados complexos e não lineares [16].

2) SMOTE: SMOTE (*Synthetic Minority Over-sampling Technique*) é uma técnica usada para lidar com o problema de classes desbalanceadas em conjuntos de dados de aprendizado de máquina [44]. Em situações nas quais uma classe é significativamente menor do que a outra — conhecidas como classes majoritária e minoritária — os modelos de aprendizado tendem a ser enviesados em favor da classe majoritária, o que resulta em uma baixa precisão na predição da classe minoritária [45]. Isso ocorre porque os modelos aprendem mais sobre a classe com mais exemplos, negligenciando a classe com menos dados [46].

Chawla et al. [44], explicam que o SMOTE cria novas instâncias sintéticas da classe minoritária em vez de simplesmente replicar as instâncias existentes, o que contribui para uma distribuição de classe mais balanceada. O método opera em três etapas principais: (1) seleção de exemplos da classe minoritária, (2) escolha de vizinhos mais próximos com base na distância euclidiana, e (3) geração de novas instâncias por interpolação entre as características dos exemplos e seus vizinhos [47]. Essas novas instâncias são então adicionadas ao conjunto de dados, aumentando a proporção da classe minoritária e facilitando a aprendizagem do modelo [47].

Fernández et al. [48], exemplificam que, em um cenário com 100 instâncias da classe majoritária e 10 da classe minoritária, o SMOTE pode gerar novas instâncias sintéticas para aumentar o número de exemplos da classe minoritária, equilibrando assim a proporção entre as classes. Essa abordagem auxilia o modelo a prestar mais atenção à classe minoritária, resultando em maior precisão preditiva [48] [49] .

Entre as vantagens do SMOTE está sua capacidade de reduzir o viés em favor da classe majoritária, promovendo uma melhor generalização para a classe minoritária, algo essencial em cenários nos quais o desbalanceamento afeta significativamente o desempenho do modelo [45]. No entanto, o SMOTE também apresenta limitações, como o aumento do risco de *overfitting*, especialmente quando há poucos dados representativos na classe minoritária [46]. Além disso, o SMOTE não resolve diretamente o desbalanceamento da classe majoritária, sendo muitas vezes necessário combiná-lo com técnicas de *undersampling* para alcançar melhores resultados [49].

3) Stratified K-Fold: O Stratified K-Fold é uma variação do método K-Fold que se destaca por garantir que a proporção das classes em cada subconjunto (ou *fold*) seja preservada. Essa abordagem é particularmente benéfica em conjuntos de dados desbalanceados, onde uma classe pode ser significativamente mais predominante do que a outra [15]. Hastie, Tibshirani e Friedman afirmam que, ao preservar a distribuição das classes em cada fold, o Stratified K-Fold resulta em uma avaliação mais estável e representativa do desempenho do modelo. Por exemplo, se um conjunto de dados contém 70% de amostras da classe "A"e 30% da classe "B", o Stratified K-Fold assegura que cada fold mantenha essa proporção, permitindo uma análise mais precisa e menos enviesada do modelo. Essa técnica é essencial para evitar a introdução de variabilidade nos resultados de validação que poderia ocorrer se uma classe menos frequente fosse sub-representada em alguns folds [15].

4) Feature Engineering: Feature engineering refere-se ao processo de seleção, transformação e criação de variáveis (*features*) que melhor representam o problema que se deseja resolver, com o objetivo de melhorar o desempenho preditivo do modelo [40]. Esse processo envolve a identificação de quais características nos dados têm maior relevância preditiva, além de possíveis modificações para torná-las mais informativas ou adequadas ao algoritmo escolhido [11] [40]. As etapas incluem a limpeza dos dados, o preenchimento de valores ausentes, a normalização, a categorização e a codificação de variáveis categóricas, além da criação de novas variáveis derivadas de outras existentes [7].

5) Otimização de Hiperparâmetros: A otimização de hiperparâmetros envolve a seleção dos melhores valores para os parâmetros que controlam o comportamento de um modelo [50] [11]. Esses parâmetros podem influenciar significativamente o desempenho do modelo e incluem aspectos como a profundidade de uma árvore de decisão, a taxa de aprendizado em algoritmos de *boosting*, entre outros [50]. A escolha apropriada desses hiperparâmetros pode resultar em melhorias substanciais na precisão e na capacidade de generalização do modelo [11]. Segundo Bergstra e Bengio, essa otimização pode ser realizada de maneira manual ou automatizada, sendo a busca exaustiva uma abordagem comum, onde várias combinações de hiperparâmetros são testadas para identificar a

configuração que oferece o melhor desempenho. Essa prática é fundamental para o desenvolvimento de modelos robustos e eficazes, maximizando seu potencial preditivo [50].

6) GridSearchCV: O GridSearchCV é uma técnica utilizada para otimizar hiperparâmetros de forma automática em modelos de aprendizado de máquina. Ele realiza uma busca exaustiva por combinações de hiperparâmetros e utiliza validação cruzada para determinar a melhor configuração [11]. De acordo com Pedregosa et al., o GridSearchCV permite que sejam ajustados parâmetros, testando várias combinações em busca da que oferece o melhor desempenho geral [51]. O uso dessa técnica contribui significativamente para que o modelo atinja seu máximo potencial, pois um ajuste adequado dos hiperparâmetros pode resultar em uma melhora nas métricas de avaliação [11]. Além disso, o GridSearchCV também fornece uma maneira estruturada de comparar diferentes modelos e suas respectivas configurações, facilitando a identificação da melhor abordagem para um determinado problema [11].

7) SHAP: SHAP (SHapley Additive exPlanations) é uma técnica baseada na teoria dos jogos, utilizada para explicar previsões de modelos de aprendizado de máquina. Seu conceito central deriva dos valores de Shapley, propostos por Lloyd Shapley em 1953 [52], e busca medir a contribuição marginal de cada variável para a previsão final [53]. Em modelos de aprendizado de máquina, cada variável é tratada como um fator que contribui para a decisão final, sendo que o valor de Shapley mede essa contribuição em diversas combinações de entradas [53].

O SHAP é reconhecido por fornecer explicações tanto locais quanto globais: as explicações locais detalham a contribuição de cada variável para uma previsão específica, enquanto as globais mostram a relevância média dessas variáveis em todo o modelo [54]. Além disso, o método é consistente e localmente preciso, garantindo que a soma das contribuições de SHAP seja equivalente à previsão do modelo [54]. Isso o torna uma ferramenta adequada para interpretar modelos complexos, como florestas aleatórias e redes neurais [54].

8) Permutation Importance: Permutation Importance é uma técnica para avaliar a importância de variáveis em modelos de aprendizado de máquina [55]. Ela consiste em embaralhar os valores de uma variável de cada vez e medir o impacto disso no desempenho do modelo [55] [28]. Se a permutação de uma variável causa uma queda significativa na performance, essa variável é considerada importante [55] [28]. A técnica é aplicável a diferentes tipos de modelos e oferece uma interpretação direta da relevância das variáveis [55]. No entanto, pode ser computacionalmente custosa, especialmente com grandes conjuntos de dados [55].

9) Boruta: Boruta é um algoritmo de seleção de variáveis usado em aprendizado de máquina que visa identificar as variáveis realmente importantes em um modelo preditivo [56]. A ideia principal do *Boruta* é avaliar a relevância de cada variável original em comparação com *shadow features* (versões "aleatórias"ou "fantasmas") [56]. Essas *shadow features* são criadas embaralhando os valores originais das variáveis, o que garante que elas não contenham informações úteis para a previsão [56].

O processo começa com a criação de cópias embaralhadas das variáveis reais [56]. Em seguida, um modelo de aprendizado de máquina é treinado com as variáveis reais e as cópias embaralhadas [56]. O algoritmo compara a importância de cada variável real com as *shadow features* [56]. Se uma variável real tiver uma importância significativamente maior que as *shadow features*, ela é considerada importante. Caso contrário, é marcada como irrelevante [56].

A vantagem do *Boruta* é que ele captura interações entre variáveis e lida com dados de alta dimensionalidade sem fazer suposições sobre a linearidade [56]. Ele também é menos suscetível ao viés de *overfitting*, pois usa um processo iterativo que aumenta a precisão da seleção de variáveis importantes [56].

Segundo Kursa e Rudnicki, que propuseram o algoritmo, o *Boruta* é ideal para tarefas em que a seleção de variáveis precisa ser feita de maneira exaustiva, sem negligenciar possíveis interações entre as variáveis preditoras [56].

10) LIME Explainer: O método LIME (*Local Interpretable Model-agnostic Explanations*) é uma abordagem proposta para fornecer explicações interpretáveis para as previsões de modelos complexos de aprendizado de máquina [57]. Ele busca resolver o problema de interpretabilidade, permitindo que os usuários compreendam as razões por trás das previsões de classificadores de forma clara e intuitiva, independentemente da complexidade do modelo utilizado [57]. O LIME funciona criando modelos lineares locais que aproximam o comportamento do modelo original em uma parte específica da amostra em questão, oferecendo uma interpretação simplificada da decisão para cada exemplo individual [58] . No entanto, apesar de sua popularidade, o LIME tem algumas limitações, como a sensibilidade a variações nos dados e a dificuldade de generalização das explicações geradas [58]. Essas limitações reforçam a necessidade de cuidado na interpretação dos resultados obtidos com o LIME, especialmente em contextos de alto risco [58].

E. Métricas Utilizadas na Avaliação de Desempenho

1) Acurácia: A acurácia é uma métrica fundamental para avaliar modelos de classificação, pois mede a proporção de previsões corretas em relação ao total de previsões feitas [36]. Essa métrica considera tanto os verdadeiros positivos quanto os verdadeiros negativos, refletindo quantas instâncias foram corretamente classificadas em um determinado conjunto de dados [36]. No entanto, a acurácia pode ser enganosa, especialmente em problemas com classes desbalanceadas [36]. Murphy destaca que, em situações onde uma classe é muito mais frequente do que a outra, um modelo pode alcançar uma alta acurácia simplesmente prevendo sempre a classe majoritária. Portanto, em contextos de desbalanceamento, métricas adicionais como precisão ou recall ou F1-score tornam-se essenciais para uma avaliação mais precisa do desempenho do modelo [36].

2) Precisão: A precisão, também conhecida como "positive predictive value", é uma métrica que avalia a proporção de previsões positivas corretas em relação ao total de previsões positivas realizadas pelo modelo [11]. Um baixo valor de precisão indica que o modelo comete muitos falsos positivos, o que pode levar a decisões erradas [11].

3) Recall (Sensibilidade): O recall, ou sensibilidade, é uma métrica que mede a capacidade de um modelo de identificar corretamente as instâncias positivas dentro de um conjunto de dados [11]. Essa métrica calcula a proporção de verdadeiros positivos em relação ao número total de exemplos positivos reais [11]. Bishop menciona que, em um cenário onde um modelo detecta 90 casos de câncer entre 100 pacientes, o recall seria de 90%. [12]. A alta importância do recall em contextos médicos enfatiza a necessidade de garantir que os casos positivos não sejam negligenciados, mesmo que isso possa resultar em um aumento dos falsos positivos [12].

4) F1 Score: O F1 Score é uma métrica de desempenho utilizada em problemas de classificação, especialmente em cenários com classes desbalanceadas [11]. Ele combina duas métricas essenciais, a precisão e o recall, em uma única medida, oferecendo uma visão mais equilibrada sobre o desempenho de um modelo de aprendizado de máquina [11].

O F1 Score é especialmente útil em situações onde há um desequilíbrio entre as classes, pois penaliza modelos que possuem precisão ou recall baixos. Seu valor varia de 0 a 1, sendo que 1 representa o melhor desempenho possível [11]. A vantagem dessa métrica em relação ao uso isolado de precisão ou recall é que ela equilibra os dois fatores, proporcionando uma avaliação menos tendenciosa, especialmente em situações críticas [11]

Por exemplo, um modelo com alta precisão, mas baixo recall, pode parecer eficaz inicialmente, mas falha em identificar muitas instâncias positivas. Já um modelo com alto recall, mas baixa precisão, captura a maioria das instâncias positivas, mas gera muitos falsos positivos [11].

5) Área sob a curva e Curva ROC: A métrica AUC-ROC (Área Sob a Curva da Característica de Operação do Receptor) combina a Curva ROC (Receiver Operating Characteristic) e a AUC (Área Sob a Curva). A Curva ROC mostra a relação entre a taxa de verdadeiros positivos (True Positive Rate, TPR) e a taxa de falsos positivos (False Positive Rate, FPR). A AUC, por sua vez, representa a área sob a curva, que varia entre 0 e 1, sendo que, quanto mais próxima de 1, melhor o modelo distingue entre as classes [59].

A Curva ROC é um gráfico que plota o recall (TPR True Positive Rate), no eixo vertical (y), enquanto a FPR (False Positive Rate) é plotada no eixo horizontal (x). Um modelo ideal apresenta uma alta TPR e uma baixa FPR, resultando em uma curva que se aproxima do canto superior esquerdo do gráfico [59].

A AUC é a medida numérica que indica a área sob a Curva ROC. Essa métrica reflete a habilidade do modelo em discriminar entre as classes [60]. Um modelo que faz previsões aleatórias teria uma AUC próxima de 0,5, enquanto um modelo perfeito teria uma AUC de 1. O uso dessa métrica é vantajoso porque independe do limiar de decisão escolhido no modelo de classificação [60].

O AUC-ROC também pode ser interpretado de diferentes formas. Um modelo que atinge AUC igual a 1 é considerado perfeito [60]. Modelos com AUC entre 0,7 e 1 são considerados bons, enquanto valores entre 0,5 e 0,7 indicam um modelo fraco [60]. Por fim, quando a AUC é igual a 0,5, o desempenho do modelo é equivalente ao de previsões aleatórias [60].

A métrica AUC-ROC é, portanto, uma ferramenta útil para medir a capacidade de discriminação dos modelos, especialmente em problemas de classificação binária, pois fornece uma avaliação robusta do desempenho do classificador, independentemente dos limiares de decisão[59] [60].

III. Metodologia

A. Fundamentação Teórica

A metodologia deste estudo começou com uma revisão teórica voltada para as áreas de engenharia de dados e aprendizado de máquina. A pesquisa abrangeu livros, artigos e documentações que discutem técnicas e ferramentas para construção de *pipelines* de dados, manipulação e processamento de dados, além da aplicação de conceitos e práticas para treinar, otimizar e avaliar modelos de aprendizado de máquina. Foram explorados conceitos fundamentais de engenharia de dados e mineração de dados, que formam a base para o pré-processamento, etapa essencial para limpar, enriquecer e transformar dados antes da aplicação dos mesmos em modelos preditivos.

Além disso, a revisão abordou modelos tradicionais de aprendizado supervisionado, conceitos importantes para interpretar resultados e também foram discutidas abordagens pertinentes ao contexto do problema, como classificação binária, métodos de avaliação, métricas, ferramentas e técnicas de otimização de modelos preditivos.

B. Coleta de Dados

Como parte do processo de desenvolvimento deste estudo, foram realizados experimentos práticos que incluíram a construção de pipelines de dados para a realização da etapa de pré-processamento. Para isso, foram utilizados conjuntos de dados públicos provenientes de repositórios do IBGE, do Governo da Bahia, da Prefeitura de Salvador e do Brasil.io. A seleção desses dados foi feita de forma estratégica, visando incorporar fatores sociais e econômicos relevantes para a análise.

Os dados coletados passaram por processos de limpeza, transformação e enriquecimento, garantindo sua integridade e representatividade. Essas etapas foram cruciais para assegurar que as informações fossem adequadamente estruturadas e prontas para alimentar os modelos preditivos, maximizando assim a precisão e a relevância dos resultados obtidos na análise.

C. Aprendizado de Máquina

Os dados foram processados com Python e Apache Spark, enquanto que a análise de dados envolveu a aplicação de técnicas de aprendizado de máquina utilizando bibliotecas como Pandas, NumPy e Scikit-learn.

A eficácia dos modelos foi avaliada utilizando um pipeline que inclui a preparação de dados, o balanceamento de classes com SMOTE, a padronização de variáveis, seguida por uma análise comparativa dos modelos de aprendizado supervisionado. Foi aplicada validação cruzada randomizada com Stratified KFold e ajuste de hiperparâmetros via GridSearchCV para garantir a robustez das avaliações.

Dentre os algoritmos utilizados para criar modelos preditivos, se destacaram o Random Forest, o Gradient Boosting, e o XGBoost.

Para todos os modelos avaliados, as métricas de desempenho incluíram acurácia, F1-Score, precisão, especificidade, sensibilidade (recall) e AUC-ROC. Além disso, foram incorporados gráficos de importância das variáveis para os modelos destacados.

A avaliação dos resultados incluiu a verificação de overfitting e underfitting, comparando a acurácia no treino e teste. Foi analisado se os modelos apresentaram alta variância ou viés por meio da análise da dispersão dos resultados na validação cruzada.

Esta abordagem permite não apenas escolher o modelo com melhor desempenho geral, mas também garantir que ele generalize bem para novos dados, evitando problemas de overfitting ou baixo desempenho por underfitting.

Dessa forma, ao longo do processo, foi necessário fazer um estudo iterativo para comparar as métricas e identificar o modelo mais apropriado para o problema em questão, garantindo a melhor relação entre desempenho, estabilidade e generalização.

IV. Planejamento e Implementação

A. *Estrutura do projeto*

A estrutura do projeto foi inspirada em grande parte pela necessidade de criar pipelines capazes de processar, integrar e analisar quatro diferentes conjuntos de dados (*datasets*), com foco principal no *dataset* de pacientes de COVID-19.

Para realizar o carregamento para limpeza, enriquecimento e transformação dos dados, foi utilizado o Apache Spark, visto que o volume de dados era relativamente grande para utilizar a biblioteca Pandas nessa etapa. Dessa forma, foi possível garantir um bom desempenho no processamento inicial da massa e dados em estado bruto.

Os *datasets* considerados para a integração foram: dados de pacientes de COVID-19, CEPs com renda per capita, vacinas, e regiões censitárias da cidade de Salvador. O processo de ETL incluiu a coleta, exploração e pré-processamento desses dados para garantir sua integridade e qualidade antes da etapa de modelagem. O *dataset* principal (pacientes de COVID-19) foi enriquecido com o *dataset* de CEPs, a fim de associar informações socioeconômicas relevantes aos dados clínicos dos pacientes.

Após a integração dos *datasets*, o pipeline foi configurado para salvar os dados processados em uma plataforma de nuvem (Google drive), que foi a ferramenta de armazenamento escolhida em detrimento de provedores de nuvem de mercado como GCP, AWS ou Azure, pois o custo de armazenamento seria maior e desnecessário, visto tratar-se de um trabalho exploratório e de natureza experimental.

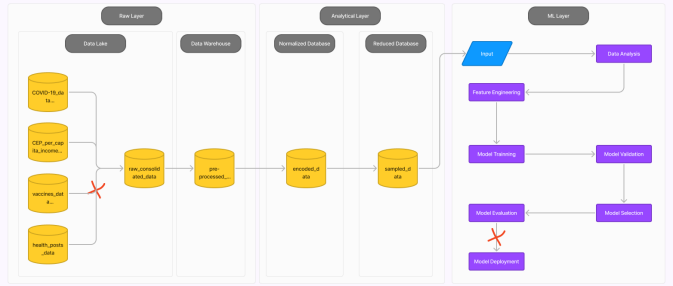

Figura 9. Dataflow do projeto

A implementação final envolveu realizar uma comparação das métricas, citadas anteriormente ao longo do trabalho, nos modelos de aprendizado de máquina aplicados na previsão de mortalidade por COVID-19, tendo sido realizada uma otimização dos hiperparâmetros para maximizar o desempenho dos modelos.

B. *Análise Exploratória de Dados*

Ao analisar os dados através de visualizações, foi possível extrair percepções sobre a composição demográfica e socioeconômica da população estudada.

A análise exploratória dos dados começa com o histograma que apresenta a distribuição da idade dos pacientes. Nele, é possível observar uma predominância de indivíduos na faixa etária de 30 a 50 anos, com um pico em torno dos 40 anos. A distribuição demonstra assimetria positiva, com frequências decrescentes para idades mais avançadas e poucos indivíduos acima dos 80 anos. Essa distribuição reflete uma concentração de pacientes adultos na base de dados, o que é relevante para a análise de predição de mortalidade, dado que condições associadas à idade, como comorbidades, desempenham um papel importante nos desfechos clínicos relacionados à COVID-19.

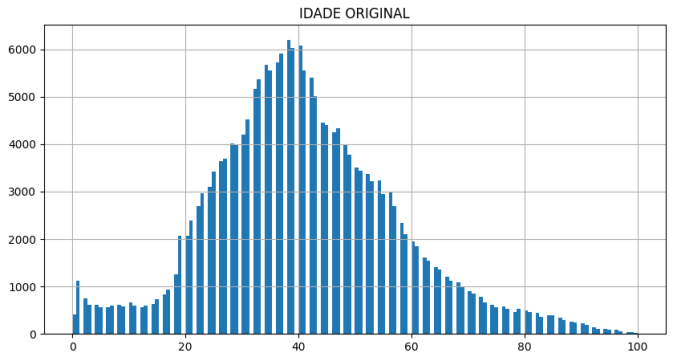

Figura 10. Distribuição normal

No *violin plot* que relaciona idade e sexo, ficou evidente que há uma maior densidade de indivíduos em faixas etárias mais jovens, principalmente entre 30 e 50, independentemente do sexo. Isso demonstra que a maior parte da amostra analisada é composta por pessoas adultas de meia idade.

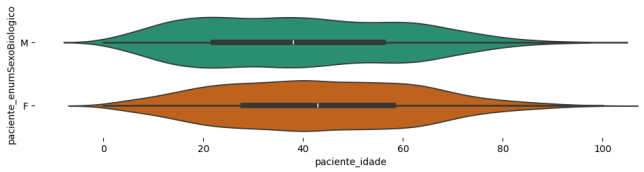

Figura 11. Violin plot por idade e sexo

Na análise comparativa entre os gêneros, o boxplot da distribuição da idade entre homens e mulheres indica similaridade nas medianas, o que sugere uma distribuição central uniforme entre os sexos. Os quartis superior e inferior estão equilibrados, reforçando que não há diferenças significativas entre os grupos nesse aspecto. No entanto, a presença de outliers em idades avançadas, particularmente acima dos 80 anos, merece destaque, pois indica uma pequena proporção de indivíduos mais idosos no conjunto de dados. Esse fator pode influenciar os resultados, especialmente ao considerar riscos mais elevados associados à idade avançada.

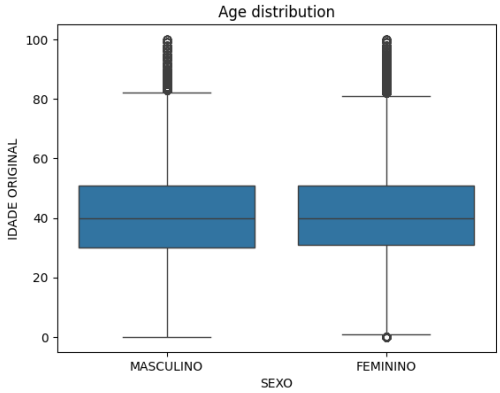

Figura 12. Boxplot por idade e sexo

O gráfico que cruza idade e raça/cor, trouxe à tona dinâmicas interessantes. Grupos como pardos e pretos apresentam uma maior concentração em idades jovens, especialmente até os 30 anos, enquanto brancos e amarelos possuem uma distribuição mais uniforme, predominando entre 30 e 60 anos. É interessante notar que a população indígena, além de ser menos representada na amostra, concentra-se em idades jovens, com uma queda significativa em faixas etárias mais avançadas. Esses padrões revelam não apenas diferenças demográficas, mas também desigualdades sociais e de saúde que afetam os grupos analisados.

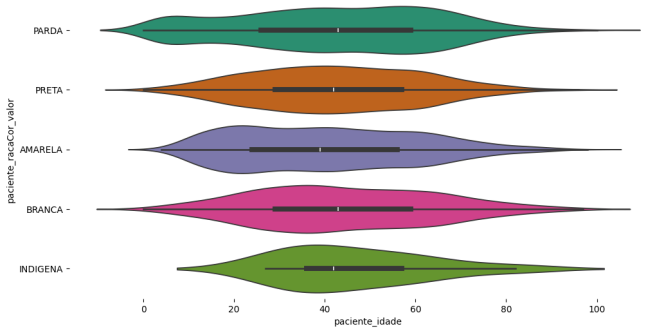

Figura 13. Violin plot por idade e raça/cor

O gráfico de dispersão georreferenciado traz uma análise da distribuição geográfica por CEP e renda em Salvador, evidenciando disparidades socioeconômicas. As áreas com menores rendas, representadas por tons mais escuros, apresentam maior densidade populacional, enquanto regiões com rendas mais altas, indicadas por tons mais claros, são menos densas. Esse padrão reflete desigualdades econômicas que podem ter impacto direto nos desfechos clínicos. Regiões de baixa renda, por exemplo, podem estar associadas a mais vulnerabilidade e a uma proporção de mortalidade hospitalar maior, reforçando a importância de incorporar esses fatores no modelo preditivo.

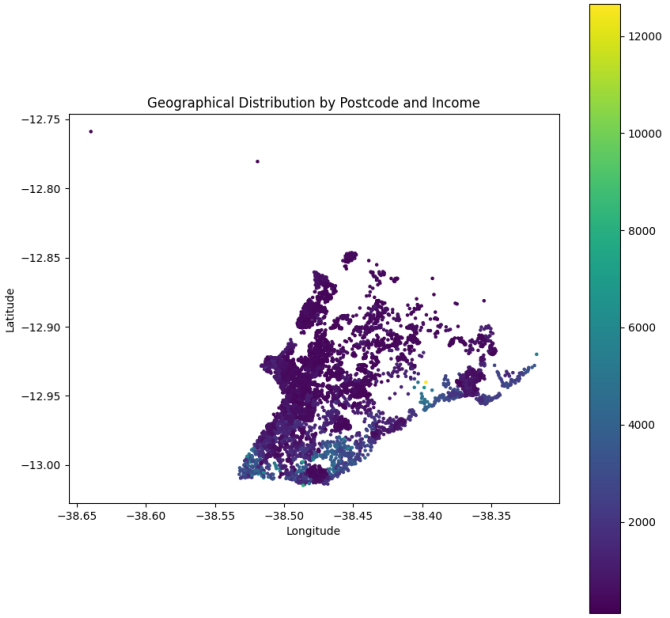

Figura 14. Gráfico de dispersão por CEP e renda per capita

Por fim, o histograma da renda per capita (censo de 2010), destacou a desigualdade econômica presente na população analisada. A maior parte dos indivíduos possui uma renda concentrada entre 0 e 1000, com uma parcela significativa com rendimentos abaixo de 500. Apesar de existirem CEPs com médias renda acima de 6000, esses casos representam uma minoria dentro do contexto geral. Essa distribuição assimétrica reflete um cenário de desigualdade econômica que pode ter implicações diretas no acesso a serviços essenciais, como saúde e educação.

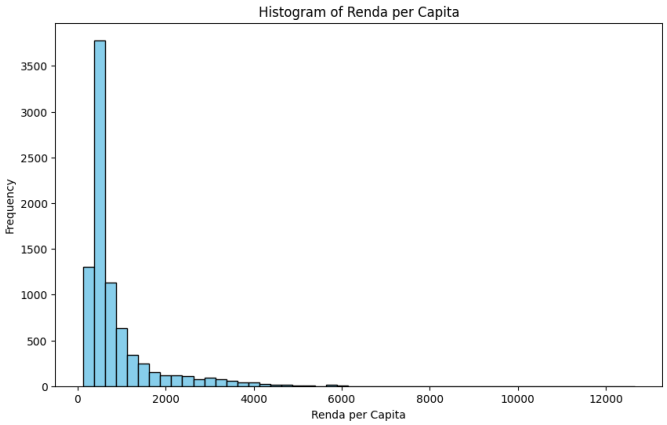

Figura 15. Histograma de renda per capital

C. Desafios Enfrentados

Durante a implementação dos pipelines, de pré-processamento e das métricas de avaliação dos modelos de aprendizado de máquina, diversos desafios surgiram.

O primeiro grande obstáculo foi a qualidade dos dados, principalmente os CEPs incompletos ou fora do padrão no *dataset* de CEPs com renda per capita. Ademais, havia uma quantidade significativa de dados faltantes no *dataset* de pacientes de COVID-19, além de preenchimentos sem padrão ou incorretos. Esses problemas comprometeram a integração com alguns dos *datasets* e, sem tratamento adequado, também afetariam a capacidade dos modelos de aprendizado de máquina de realizar predições precisas.

Outro desafio importante foi a impossibilidade de utilizar os *datasets* de vacinas e de regiões censitárias. Esses dados poderiam enriquecer o *dataset* principal ou servir para análises que explorassem a relação entre a taxa de vacinação, a região geográfica e o índice de mortalidade por COVID-19 na cidade de Salvador. No entanto, a ausência de um campo comum que pudesse ser usado como chave estrangeira entre o *dataset* de vacinação e o de pacientes de COVID-19 impediu a integração desses dados. Da mesma forma, os dados de regiões censitárias não puderam ser aproveitados para a plotagem de mapas ou para uma análise que identificasse a adequação da infraestrutura de Saúde Pública em áreas de maior densidade populacional.

Quanto ao pré-processamento dos dados, foi outra etapa com bastante detalhes. Foi necessário enriquecer o *dataset* ao substituir valores nulos em algumas colunas, remover *outliers*, realizar encoding de variáveis categóricas e extrair amostras balanceadas para utilizar Pandas e Scikit-learn sem consumo excessivo de memória, dado que o Pandas tem desempenho limitado com *datasets* volumosos.

Utilizando como referência o artigo de Moulaei [5], foi adotada uma limpeza dos dados que excluiu pacientes que estivessem abaixo de 18 anos durante o pré-processamento, visto que, segundo os autores, é um grupo que faz parte de um escopo mais restrito de estudo: a pediatria.

Dos quase 8 milhões de registros disponíveis, foi feita uma amostragem que reduzisse esse número para 0,1% (cerca de 8000) de registros extraídos de forma balanceada para evitar vieses. Além da amostragem, dados nulos na parte de confirmação de óbito foram preenchidos com uma proporção de 90% de sobrevivência e 10% de fatalidade. Essa proporção se baseou em duas pesquisas [61] [62] publicadas no *OurWorldInData.org* que retratam um cenário mundial da pandemia de COVID-19 com uma média de fatalidade (case fatality rate) de 10%.

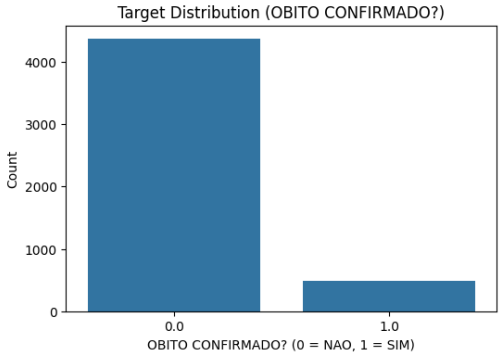

Figura 16. Proporção de sobreviventes e óbitos confirmados

Para finalizar o tratamento da amostragem após o preenchimento de dados faltantes com base nos estudos sobre fatalidade por COVID-19, também foi utilizado o SMOTE para evitar que os modelos se tornassem tendenciosos na predição, visto que, apesar do alto índice de mortalidade, na vasta maioria dos casos, os pacientes sobrevivem. Esse processo foi importante para garantir que os modelos pudessem ser treinados de maneira eficiente.

No aprendizado supervisionado binário, a otimização dos modelos de aprendizado de máquina foi um desafio significativo, especialmente devido à tendência de alguns modelos, sobretudo os que utilizam *boosting*, ao *overfitting*, pois têm uma tendência maior a ter variância alta devido à quantidade grande de variáveis do modelo. Na busca de equilíbrio entre viés e variância, o ajuste dos hiperparâmetros com GridSearchCV foi muito útil, assegurando um desempenho elevado dos modelos e reduzindo o risco de *overfitting*, ou seja, controlando a variância excessiva e aumentando um pouco mais o viés dos modelos. Isso exigiu múltiplas execuções, testes com diferentes volumes de dados e, conforme apresentado anteriormente, técnicas de balanceamento, além de amostragens equilibradas e ajustes para garantir uma boa capacidade de generalização dos modelos sendo avaliados.

V. Resultados

A. Comparação de métricas entre modelos

Dentre os modelos escolhidos, se destacam o Random Forest, o XGBoost e o Gradient Boosting. No entanto, o XGBoost foi o que obteve o melhor desempenho geral conforme a figura 16:

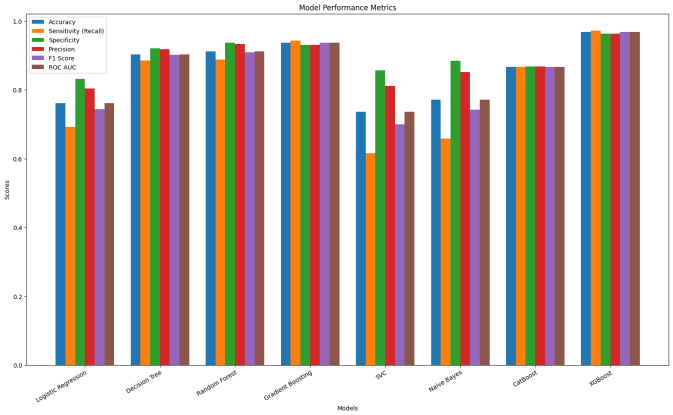

Figura 17. Comparação de métricas por modelo.

Os resultados da matriz de confusão indicam um bom desempenho geral do modelo, com baixas taxas de erro, sugerindo que o XGBoost foi eficaz na predição correta dos desfechos analisados, com 4.270 verdadeiros positivos e 4.211 verdadeiros negativos. Os erros foram baixos: 128 falsos negativos e 187 falsos positivos.

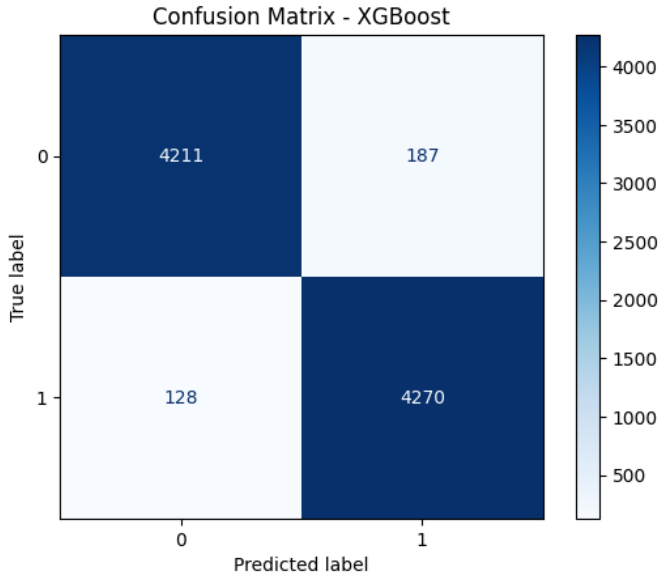

Figura 18. Matriz de confusão.

B. *Inspeção de Folds*

Após a escolha do XGBoost como algoritmo do melhor modelo preditivo para o problema apresentado, de acordo com as métricas adotadas, foram feitas validações cruzadas randomizadas divididas em dez folds, tanto para acurácia quanto para sensibilidade, no intuito de observar os resultados de teste e treino, a cada fold, para observar se haveria uma diferença muito grande entre as métricas de teste e treino. No entanto, isso não se constatou e os resultados comprovaram uma tendência do modelo e realizar boas generalizações.

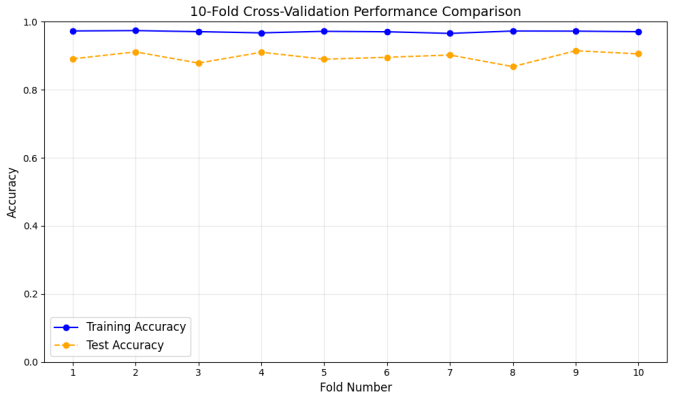

Figura 19. Comparação entre resultados de teste e treinamento

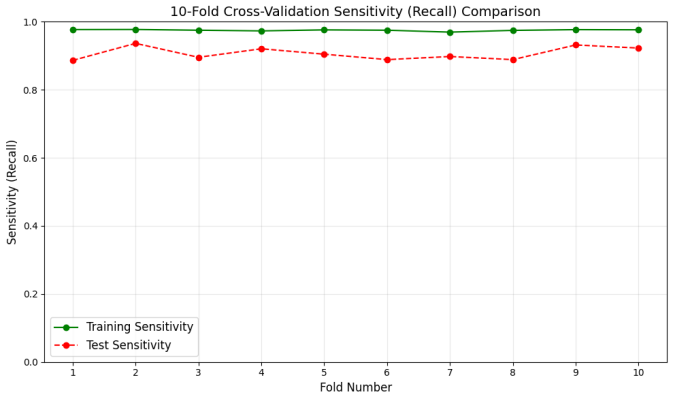

Figura 20. Comparação entre resultados de teste e treinamento

C. *Análise de Curva de Aprendizado*

Quanto à curva de aprendizado do XGBoost, foi notado que a acurácia das predições de treino aumentou sempre que o número de amostragem era elevado, mas teve uma leve queda de ganhos em torno de 6250 e uma retomada de ganhos a partir de 7000.

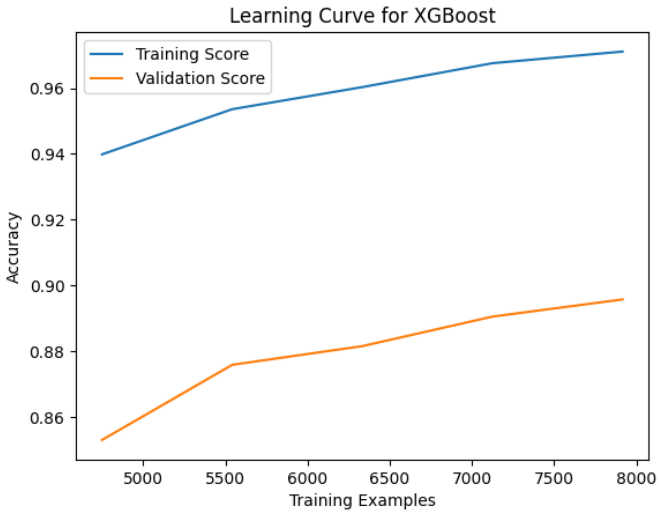

Figura 21. Curva de aprendizado do XGBoost

Por isso fez sentido escolher uma amostragem de cerca de 8000 (0.1%), conforme dito anteriormente, para otimizar o uso de memória durante o processamento de dados, visto que a pontuação não teria mais uma grande margem de progressão. Em adição a esse critério, como podemos observar na imagem, a pontuação de validação pode ser dividida em três dimensões de amostragem:

- Até 5500 amostras, a pontuação tem uma progressão significativa de praticamente 1 ponto percentual;
- Entre 5500 e 7000 a progressão da pontuação se estabiliza, mas tem uma leve queda de desempenho com os dados de treino. No entanto, com os dados de validação, a pontuação continua subindo;
- No terceiro grupo da amostragem que está acima de 7000, a pontuação de validação e treino se estabilizam e apenas têm uma leve tendência de aumento.

Essas tendências das amostragens fizeram com que o número de 8000 (0.1%) fosse escolhido como quantidade ideal para treinar e validar o XGBoost neste trabalho.

D. Variáveis preditoras nos três principais modelos

Quanto aos principais fatores preditivos de mortalidade entre os pacientes hospitalizados com COVID-19, foi feita uma análise dos três modelos preditores com os melhores desempenhos com BorutaPy e scikit-learn.

Conforme as imagens a seguir, podemos perceber que os principais fatores (vistos de modo individual e sem interação com outras variáveis) comuns aos três melhores modelos preditivos são: fatores geográficos (latitude, longitude e CEP), demográficos (raça/cor e densidade populacional) e socioeconômicos (renda per capita).

Para o Random Forest, podemos ver que além dos principais fatores supracitados, raça/cor também se destaca, apesar de não ter um peso individual equiparável às principais variáveis preditoras do modelo.

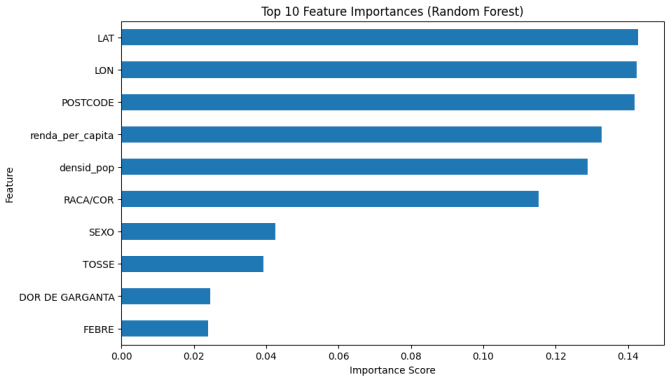

Figura 22. Importância preditiva de variáveis com o Random Forest

No caso do Gradient Boosting, nenhum dos outros campos, além de raça/cor, parece ter um peso individual significativo o suficiente para determinar uma predição mais precisa de forma isolada.

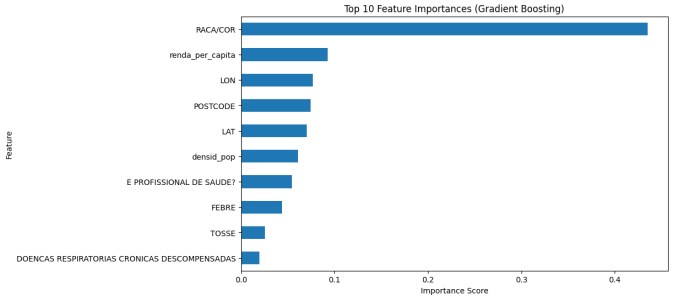

Figura 23. Importância preditiva de variáveis com o Gradient Boosting

Por fim, no caso do XGBoost, além da forte influência dos campos já mencionados nos outros modelos, há outros campos que ajudam na predição, dentre eles fatores que têm relação com sintomas clinicos (tosse, febre e dor de garganta, por exemplo), condições crônicas de saúde (diabetes, imunossupressão, doenças cardiovasculares), ou condições de maior vulnerabilidade (grávidas e profissionais de saúde, por exemplo).

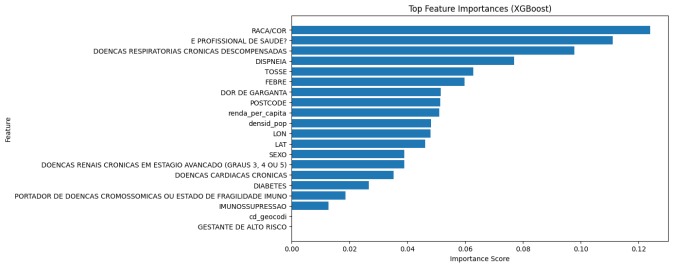

Figura 24. Importância preditiva de variáveis com o XGBoost

Para continuar a análise comparativa, foi feito um quadro para examinar a relação entre renda, sexo, raça e cep. Na matriz de dispersão, vemos que não há uma correlação forte entre as variáveis latitude, longitude, renda e cep. Por vezes, variáveis com correlações muito fortes são eliminadas do conjunto de dados para testar o comportamento das predições e evitar overfitting, mas neste estudo foi escolhido não retirar as variáveis latitude, longitude, renda e cep, pois elas trazem a dimensão de recorte de classe para o conjunto de dados estudado.

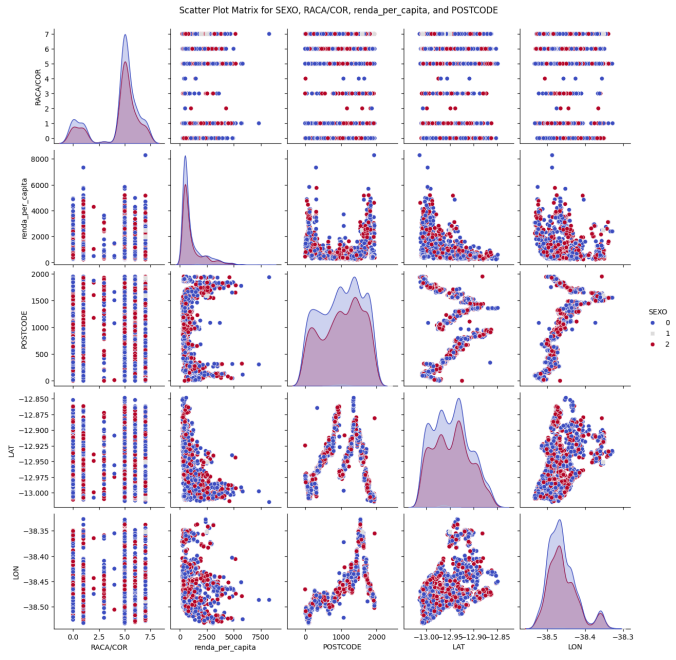

Figura 25. Grpafico de dispersão

Quanto aos gráficos de densidade, eles fornecem informações sobre as distribuições individuais das variáveis, enquanto a distribuição aleatória de sexo nos gráficos sugere que ele pode não ter uma interação significativa com raça/cor, renda ou cep. É necessário realizar outras validações com outros tipos de análise, como a análise SHAP e/ou uma análise com LIME Explainer, para garantir tais afirmações ou identificar essas combinações e contribuições de forma mais concreta. Os itens E e F apresentam essas análises.

E. Análise SHAP

Para aprofundar a análise dos resultados, foi gerado um gráfico que mostra uma análise SHAP, feita a partir de uma amostragem de 1000 registros. Ela evidencia a interação entre as variáveis no modelo feito com XGBoost, revelando como esses fatores influenciam conjuntamente a predição.

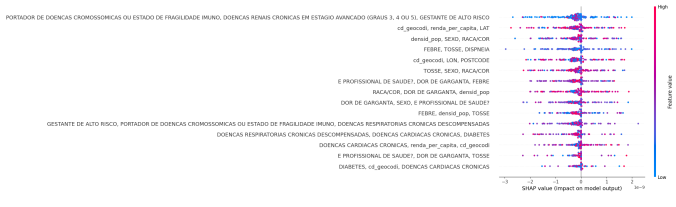

Figura 26. Análise SHAP

Com a análise SHAP, é possível vermos as combinações de variáveis que mais influenciam os resultados do modelo de mortalidade.

Com base no gráfico SHAP, observa-se que as variáveis com maior impacto nas previsões do modelo incluem condições de saúde graves, como doenças cromossômicas, imunossupressão, doenças renais crônicas em estágio avançado e gestação de alto risco, evidenciando a forte influência de comorbidades na mortalidade. Além disso, características sociodemográficas como "sexo" e "raça/cor" também são relevantes, sugerindo vulnerabilidades específicas dentro da população. Variáveis territoriais e socioeconômicas, como densidade populacional, "renda per capita", coordenadas geográficas e códigos de localização como o CEP, também apresentam impacto significativo, indicando desigualdades socioeconômicas de acordo com a região da cidade. Sintomas clínicos como febre, dor de garganta, tosse e dispneia complementam o conjunto de fatores críticos. Dessa forma, o modelo capta tanto elementos clínicos quanto determinantes sociais da saúde, reforçando a importância de uma abordagem integrada na predição dos desfechos.

F. Análise com LIME Explainer

Com o LIME Explainer, observa-se que a variável "RAÇA/COR <= 5.00" exerce o maior impacto negativo nas predições, indicando que determinados grupos raciais apresentam maior vulnerabilidade à mortalidade por COVID-19. A variável "É PROFISSIONAL DE SAÚDE? <= 0.00" também aparece como um importante fator de risco, sugerindo que, para o modelo, não ser profissional de saúde está associado a um aumento na probabilidade de óbito, mas isso só faz sentido se levarmos em conta que o conjuto de dados utilizado composto em sua maioria por pessoas que não são profissionais de saúde. Quanto à localização geográfica, representada pela longitude "LON > -38.44", ela também está associada a um aumento na chance de morte. Isso sugere que o modelo identifica aspectos socioeconômicos e geográficos como fatores relevantes para a mortalidade, conforme havia sido constatado na análise SHAP.

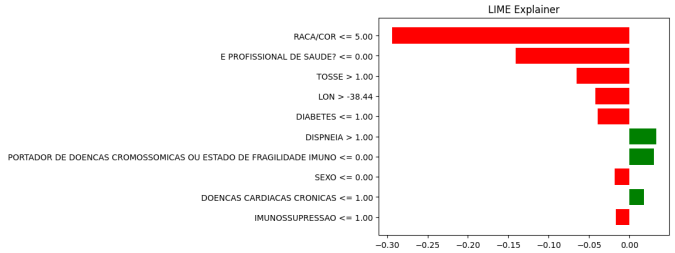

Figura 27. Análise com LIME

Além disso, a variável "DIABETES <= 1.00" contribui negativamente para a predição, indicando que a ausência da doença está associada a menor risco, reforçando a ideia de que o diabetes é uma comorbidade relevante que eleva a probabilidade de mortalidade em pacientes infectados pela COVID-19.

Em resumo, a análise LIME reforça as conclusões obtidas na análise SHAP, destacando a importância combinada de fatores demográficos, clínicos, geográficos e socioeconômicos na modelagem do risco de mortalidade.

G. Comparação dos resultados com outros trabalhos semelhantes

Ao comparar os resultados do estudo com outros três artigos sobre predição de mortalidade por COVID-19 que também

usaram algoritmos de aprendizado de máquina, várias semelhanças e algumas diferenças metodológicas surgem, revelando diferentes abordagens para lidar com desafios específicos.

Em todos os três artigos escolhidos, foram pré-selecionadas apenas pessoas que tinham sido testadas positivo para COVID-19. Primeiramente, no artigo "Comparing machine learning algorithms for predicting COVID-19 mortality" [5], os autores usaram um conjunto de dados com 1500 pacientes adultos, aplicando a técnica SMOTE para balancear classes e melhorar a robustez do modelo Random Forest, que apresentou a melhor acurácia (95,03%) e uma alta AUC (99,02%). Esse estudo incluiu variáveis como dispneia e terapia com oxigênio, destacadas como fatores significativos na predição do óbito, apesar de também terem usado variáveis demográficas. A principal limitação identificada foi a origem monocêntrica dos dados, sugerindo a necessidade de validações em múltiplos centros para uma melhor generalização dos resultados.

Em contrapartida, o presente estudo utilizou dados de diversos estabelecimentos de saúde de Salvador - Bahia e uma comparação entre sete algoritmos. Dentre eles, XGBoost, Random Forest e Gradient Boosting, onde o XGBoost obteve o melhor desempenho geral em termos de acurácia e generalização. A utilização de uma amostragem mais ampla e variada (8000 registros ou cerca de 0.1% do número total) com mais variáveis demográficas, econômicas e geográficas, além dos dados clínicos do dataset principal, foi uma decisão prática para garantir uma boa consistência nas previsões, ainda que o impacto de algumas variáveis tenha se mostrado limitado isoladamente.

O segundo estudo, "A comparison of machine learning algorithms in predicting COVID-19 prognostics" [25], abordou um conjunto de dados maior que a amostragem deste trabalho (11.712 pacientes), com variáveis laboratoriais adicionais, que provaram ser importantes para predições acuradas, além da eliminação de variáveis com correlações muito fortes para evitar overfitting. Nesse estudo, os modelos baseados em árvores, como Extra Trees e CatBoost, apresentaram AUC ROC superior a 0,94, refletindo o impacto positivo das variáveis laboratoriais nos resultados.

Em comparação, o presente estudo revelou uma abordagem focada em dados clínicos, demográficos e econômicos, incorporando variáveis como densidade demográfica e renda, que se mostraram influentes no contexto socioeconômico da cidade de Salvador. O dataset principal deste trabalho tem dados clínicos e laboratoriais focados na triagem de pacientes assim como os trabalhos usados para comparação.

Apesar dos ótimos resultados das métricas com o XGBoost, a ausência de uma gama maior de dados laboratoriais pode limitar a capacidade do presente estudo de aprofundar o impacto de variáveis clínicas adicionais e específicas na recomendação de outros tratamentos além de cuidado intensivo.

Por fim, o artigo "Comparative Analysis of Machine Learning Algorithms for Predicting Covid-19 Mortality in Children and Adolescents Using a Large Public Dataset in Brazil" [63] focou em uma população pediátrica com 37 variáveis, destacando a Regressão Logística como o modelo mais eficaz com acurácia de 92,5%. A redução de saturação de oxigênio e presença de comorbidades foram variáveis críticas para predizer mortalidade em jovens. Em comparação, o presente estudo se beneficiou de um aporte de variáveis preditoras mais abrangentes e contextuais, enquanto que as limitações na coleta mais detalhada de dados laboratoriais não impediram um bom desempenho e generalização do modelo escolhido.

Essas diferenças metodológicas entre os estudos sugerem que, enquanto técnicas como o SMOTE e variáveis laboratoriais adicionais e clínicas melhoram a acurácia de alguns modelos em cenários específicos, a escolha por variáveis mais gerais e o uso de validações mais amplas conferem ao presente estudo uma maior confiança nos resultados para análises em contextos geograficamente variados.

H. Explicabilidade além dos datasets estudados

As análises realizadas (SHAP e LIME Explainer) corroboram com outros estudos, mais voltados para ciências sociais que apontam, além do recorte de classe, haver uma maior taxa de mortalidade entre pessoas negras. Na tabela abaixo, considerando o total combinado de pardos e pretos (pessoas negras), pode-se afirmar que há uma maior propensão proporcional à mortalidade (0,84%) nesse grupo.

Tabela I. DISTRIBUIÇÃO DE ÓBITOS POR RAÇA/COR

Raça/Cor	Total de Pessoas	Total de Óbitos	Percentual de Óbitos (%)
IGNORADA	6.723	1.919	28,54
PARDA	2.716.288	23.268	0,86
AMARELA	550.861	4.389	0,80
PRETA	739.596	5.770	0,78
BRANCA	525.630	3.935	0,75
INDÍGENA	5.922	42	0,71
NULL	548.829	388	0,07
IGNORADO	68.500	23	0,03

Nesse recorte demográfico, as mulheres negras, em particular, apresentaram um índice de mortalidade proporcionalmente maior em comparação a outros grupos, conforme estudos publicados em Ciência & Saúde Coletiva [64], no Saúde Debate [65]e no Cadernos de Saúde Pública [66].

De acordo com essas pesquisas, a fatalidade por COVID-19 foi mais elevada entre pessoas negras de baixa renda e/ou em situação de rua. As(os) autoras(es) desses estudos discorrem sobre o fato da população negra no Brasil geralmente residir em áreas com maior densidade populacional e infraestrutura precária de saneamento básico e serviços públicos, fatores que elevam os índices de infecção e aumentam o índice de mortalidade dessa parcela da população, desmontando assim tese de que a pandemia foi "democrática" e afetou igualmente os diferentes grupos demográficos [64] [65] [66].

Portanto, apesar da variável de gênero não ter apresentado ainda mais relevância nas predições do XGBoost, conforme evidenciado pelas análises com SHAP e LIME Explainer, é importante destacar que esses métodos não revelam relações causais, mas sim padrões associativos que o modelo identificou nos dados, com base em correlações.

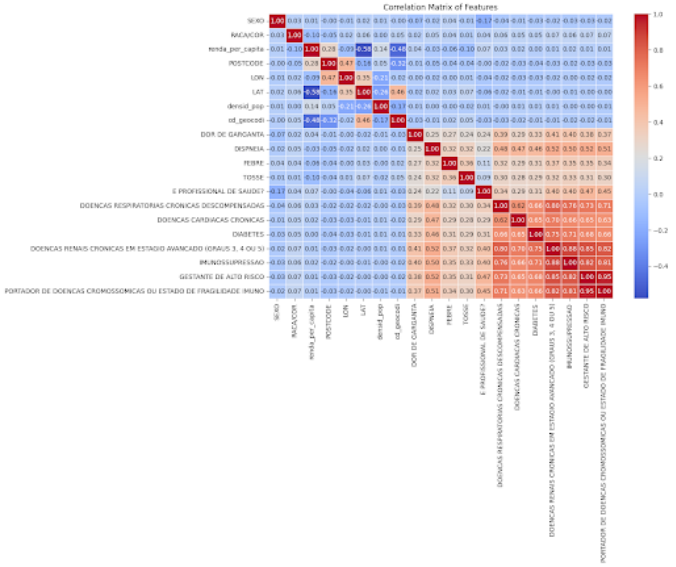

Figura 28. Matriz de correlação

Para realizar uma análise causal seriam necessárias outras ferramentas que exigem um poder computacional maior e recursos (infraestrutura de nuvem paga) que estão fora do escopo deste trabalho.

Nos trabalhos de ciências sociais citados anteriormente, raça e gênero, além de indicadores que refletem recorte de classe, são fatores correlatos determinantes segundo as análises quantitativas e qualitativas das autoras (es).

Neste trabalho, a partir do modelo preditivo desenvolvido com XGBoost, assim como nos estudos sociais mencionados, as características indicativas de uma maior chance de fatalidade que emergem com mais evidência estão relacionadas aos recortes de classe e raça:

- Fatores geográficos (latitude, longitude e CEP);
- Fatores demográficos (raça e densidade populacional);
- Fatores econômicos (renda per capita média por CEP).

Correlação não reflete necessariamente causalidade. Analisando a matriz da figura 34, não é possível enxergar claramente uma correlação entre raça/cor, sexo e fatores de risco, tampouco uma causalidade direta, ao contrário dos estudos sociais citados que utilizaram uma visão mais ampla e qualitativa para analisar as mortes causadas por COVID-19.

No entanto, no gráfico de permutação, vemos uma análise de importância das variáveis e é notório o peso que as variáveis de recorte de classe e raça têm, mas a figura também reafirma a importância dos recortes de gênero nas inferências preditivas do modelo.

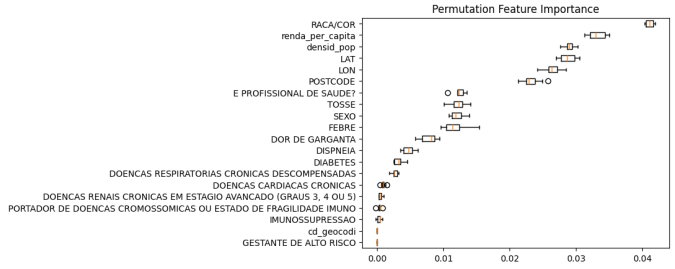

Figura 29. Permutation feature importance

Desta forma, observando a emergência de padrões semelhantes quanto à inferência preditiva, podemos dizer que o modelo consegue ter um melhor desempenho graças à captura de aspectos mais abrangentes da realidade dos pacientes, que não necessariamente são clínicos ou laboratoriais.

Apesar de algumas limitações dos dados do *dataset*, visto que dados de raça/cor, nem sempre são corretamente preenchidos ou sequer o são (10% dos registros não possuem informação alguma para essa variável), o modelo desenvolvido ajuda a interpretar a influência de dados demográficos e geográficos utilizando uma gama mais ampla de fontes.

Os resultados obtidos confirmam a necessidade de adotar abordagens interseccionais que revelem nuances frequentemente ignoradas pelo pragmatismo dos modelos e ofereçam uma perspectiva interdisciplinar com um alcance mais amplo para estudar e resolver problemas complexos em saúde coletiva, como a alta mortalidade por COVID-19 em diferentes grupos demográficos e regiões geográficas. Considerando o contexto socioeconômico do estudo de forma mais acurada, essas abordagens podem contribuir não apenas para a alocação de recursos hospitalares e a tomada de decisões clínicas em momentos de crise, mas também para avanços na epidemiologia, garantindo uma saúde pública efetiva e de qualidade para as parcelas da população que mais precisam dela.

VI. Conclusão

Este trabalho teve como objetivo principal aplicar técnicas e conceitos de engenharia de dados e aprendizado de máquina para comparar modelos preditivos no contexto de mortalidade de pacientes por COVID-19. O estudo utilizou um conjunto de dados que integra informações socioeconômicas e de saúde, empregando ferramentas como Apache Spark para lidar com o grande volume de dados, além de bibliotecas como Pandas, NumPy e Scikit-learn para manipulação, análise dos dados e predição. Foram exploradas técnicas de aprendizado supervisionado binário e algoritmos como Random Forest, Gradient Boosting e XGBoost, buscando desenvolver modelos preditivos robustos e confiáveis.

Os resultados demonstraram que as técnicas de balanceamento dos dados e validação cruzada randomizada foram cruciais para garantir a capacidade de generalização dos modelos. Apesar dos desafios enfrentados, como dados incompletos ou inconsistentes e a ausência de uma chave comum para integrar *datasets* de pacientes, de vacinação e de postos de saúde, foi possível alcançar bons níveis de desempenho preditivo. Em particular, os modelos apresentaram métricas satisfatórias, com o XGBoost destacando-se pela maior precisão e capacidade

de identificar padrões em dados complexos. Esses resultados reforçam o potencial do aprendizado de máquina para apoiar decisões críticas no contexto da saúde coletiva.

Entretanto, algumas limitações reduziram o escopo do estudo. A qualidade dos dados disponíveis, incluindo CEPs incompletos e valores faltantes, causou uma perda de dados após as tabelas terem sido unidas (5% dos registros não tinham CEP preenchido). A integração de dados socioeconômicos, embora enriquecedora, foi prejudicada pela ausência de campos comuns com outros *datasets* públicos, o que reduziu a abrangência das análises. Além disso, o processo de otimização dos modelos demandou múltiplos ajustes para mitigar o *overfitting*, evidenciando a importância do pré-processamento para obter dados mais consistentes e representativos e melhorar a capacidade de generalização dos modelos.

Como desdobramento, futuras melhorias podem incluir a incorporação de novos dados, como registros mais completos de vacinação e informações detalhadas de unidades de saúde, caso contenham campos de referência para integração. Outras possibilidades envolvem a aplicação do estudo a diferentes contextos geográficos e a evolução deste experimento em um sistema de recomendação voltado para a vigilância epidemiológica de doenças respiratórias agudas, permitindo a detecção precoce de padrões e a melhor alocação de recursos em cenários críticos. Além disso, o sistema poderá ser expandido para auxiliar na identificação e monitoramento de doenças crônicas, como diabetes e hipertensão, fornecendo suporte à tomada de decisões clínicas e à formulação de políticas públicas voltadas para a infraestrutura da saúde. Espera-se que essas iniciativas aproveitem o potencial da inteligência artificial aplicada à saúde coletiva e contribuam para o aprimoramento das políticas do sistema público de saúde, aumentando a eficiência das estratégias de prevenção e controle de doenças.

REFERÊNCIAS

[1] Organização Pan-Americana da Saúde, "Histórico da pandemia covid-19," https://www.paho.org/pt/covid19/historico-da-pandemia-covid-19, acesso em: 03 out. 2024.

[2] Fundação Oswaldo Cruz, "A gestão de riscos e governança na pandemia por covid-19 no brasil: análise dos decretos estaduais no primeiro mês," Escola Nacional de Saúde Pública Sergio Arouca, Tech. Rep., 2020.

[3] World Economic Forum, "How ai and machine learning are helping to fight covid-19," https://www.weforum.org/agenda/2020/05/how-ai-and-machine-learning-are-helping-to-fight-covid-19/, acesso em: 03 out. 2024.

[4] M. H. R. Galvão and A. G. Roncalli, "Fatores associados a maior risco de ocorrência de óbito por covid-19: análise de sobrevivência com base em casos confirmados," Revista Brasileira de Epidemiologia, vol. 23, 2020.

[5] K. Moulaei, M. Shanbehzadeh, Z. Mohammadi-Taghiabad, and H. Kazemi-Arpanahi, "Comparing machine learning algorithms for predicting covid-19 mortality," Journal of Big Data, 2023.

[6] Bahia, "Painel covid-19 bahia," https://infovis.sei.ba.gov.br/covid19/, acesso em: 03 out. 2024.

[7] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2012.

[8] J. Reis and M. Housley, Fundamentals of Data Engineering: Plan and Build Robust Data Systems, 1st ed. O'Reilly Media, 2022.

[9] J. Densmore, Data Pipelines Pocket Reference: Moving and Processing Data for Analytics, 1st ed. O'Reilly Media, 2021.

[10] J. Jo and S. Lee, Data Engineering: Concepts and Techniques. Berlin: Springer, 2019.

[11] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

[12] C. M. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006.

[13] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Upper Saddle River: Pearson, 2009.

[14] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[15] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York: Springer, 2009.

[16] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R. New York: Springer, 2013.

[17] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Boca Raton: Chapman & Hall/CRC, 1984.

[18] C. Molnar, Interpretable Machine Learning. Springer, 2019.

[19] A. Taly, P. Deshpande, S. Singh et al., Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, 2020.

[20] M. Mitchell, Artificial Intelligence: A Guide for Thinking Humans. Penguin Books, 2019.

[21] L. E. P. F. de Souza, "Saúde pública ou saúde coletiva?" Revista Espaço para a Saúde, vol. 15, no. 4, pp. 1–21, 2014. [Online]. Available: https://www.escoladesaude.pr.gov.br/arquivos/File/saude$_{p}ublica_{4}.pdf$

[22] J. S. Paim, Saúde Coletiva: Teoria e Prática. Rio de Janeiro, Brasil: Medbook, 1999.

[23] C. Teixeira, N. de Almeida Filho, and J. Paim, Epidemiologia e Saúde: Fundamentos, Métodos e Aplicações. Rio de Janeiro, Brasil: Editora Fiocruz, 1998.

[24] M. I. N. de Albuquerque, E. M. F. de Carvalho, and L. P. Lima, "Vigilância epidemiológica: conceitos e institucionalização," Revista Brasileira de Saúde Materno Infantil, vol. 9, no. 2, pp. 203–212, 2009. [Online]. Available: https://www.scielo.br/j/rbsmi/a/6L4R958YLyJywqtG9WcRRCv/

[25] S. Ustebay, A. Sarmis, G. K. Kaya, and M. Sujan, "A comparison of machine learning algorithms in predicting covid-19 prognostics," Internal and Emergency Medicine, vol. 18, no. 1, pp. 229–239, Jan. 2023, disponível em: https://pubmed.ncbi.nlm.nih.gov/36116079/. Acesso em: 23 out. 2024.

[26] Scikit-Learn, "Tree," https://scikit-learn.org/1.5/modules/tree.html, acesso em: 10 out. 2024.

[27] E. Alpaydin, Introduction to Machine Learning, 3rd ed. Cambridge: MIT Press, 2014.

[28] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[29] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000.

[30] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[31] Scikit-Learn, "Ensemble," https://scikit-learn.org/1.5/modules/ensemble.html, acesso em: 15 out. 2024.

[32] ——, "Naive bayes," https://scikit-learn.org/1.5/modules/naive$_{b}ayes.html, acessoem: 10out.2024$.

[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[34] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," Advances in Neural Information Processing Systems, vol. 31, 2018.

[35] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," arXiv preprint arXiv:1810.11363, 2018.

[36] K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA: MIT Press, 2012.

[37] T. M. Cover and J. A. Thomas, Elements of Information Theory, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2006.

[38] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[39] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in IJCAI, vol. 14, no. 2, 1995, pp. 1137–1145.

[40] M. Kuhn and K. Johnson, Applied Predictive Modeling. Springer, 2013.

[41] D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, 5th ed. New York: Wiley, 2011.

[42] I. T. Jolliffe, Principal Component Analysis, 2nd ed. New York: Springer, 2002.

[43] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," Chemometrics and Intelligent Laboratory Systems, vol. 2, pp. 37–52, 1987.

[44] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[45] V. García, J. Saez, J. Luengo, and F. Herrera, "A survey on the use of over-sampling and undersampling for class imbalance learning," ACM Computing Surveys, vol. 45, no. 3, pp. 1–39, 2012.

[46] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 20–29, 2004.

[47] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in International Conference on Intelligent Computing. Springer, 2005, pp. 878–887.

[48] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, Learning from Imbalanced Data Sets. Springer, 2018.

[49] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

[50] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, vol. 13, pp. 281–305, 2012.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[52] L. S. Shapley, "A value for n-person games," in Contributions to the Theory of Games. Princeton University Press, 1953, vol. 2, pp. 307–317.

[53] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.

[54] S. M. Lundberg, G. G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "Explainable ai for trees: From local explanations to global understanding," Nature Machine Intelligence, vol. 2, pp. 252–262, 2020.

[55] E. Documentation, "Permutation importance," n.d., accessed: 2024-12-11. [Online]. Available: https://eli5.readthedocs.io/en/latest/blackbox/permutation$_importance.html$

[56] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," Journal of Statistical Software, vol. 36, no. 11, pp. 1–13, 2010, disponível em: https://www.jstatsoft.org/article/view/v036i11. Acesso em: 20 out. 2024.

[57] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?"explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016, pp. 1135–1144.

[58] D. Garreau and U. von Luxburg, "Explaining the explainer: A first theoretical analysis of lime," in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS). PMLR, 2020, pp. 1287–1296.

[59] T. Fawcett, "An introduction to roc analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[60] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," Pattern Recognition, vol. 30, no. 7, pp. 1145–1159, 1997.

[61] E. Mathieu, H. Ritchie, L. Rodés-Guirao, C. Appel, C. Giattino, J. Hasell, B. Macdonald, S. Dattani, D. Beltekian, E. Ortiz-Ospina, and M. Roser, "Coronavirus pandemic (covid-19)," Our World in Data, 2020, https://ourworldindata.org/coronavirus.

[62] E. Mathieu, H. Ritchie, L. Rodés-Guirao, C. Appel, D. Gavrilov, C. Giattino, J. Hasell, B. Macdonald, S. Dattani, D. Beltekian, E. Ortiz-Ospina, and M. Roser, "Mortality risk of covid-19," Our World in Data, 2020, https://ourworldindata.org/mortality-risk-covid.

[63] A. Lages dos Santos, M. C. L. L. Oliveira, E. A. Colosimo, C. Pinhati, S. C. Galante, H. Martelli-Júnior, A. C. Simões e Silva, and E. Oliveira, "Comparative analysis of machine learning algorithms for predicting covid-19 mortality in children and adolescents using a large public dataset in brazil," https://ssrn.com/abstract=4740297, 2024, acesso em: 18 out. 2024.

[64] F. M. Estrela, C. F. Soares, M. A. d. Cruz, A. F. d. Silva, J. R. L. Santos, T. M. d. O. Moreira, A. B. Lima, and M. G. Silva, "Pandemia da covid-19: Refletindo as vulnerabilidades à luz do gênero, raça e classe," Ciência & Saúde Coletiva, vol. 25, no. 9, pp. 3431–3436, 2020.

[65] A. P. d. Bishreis, E. F. Góes, F. B. Pilecco, M. d. C. C. d. Almeida, L. M. Diele-Viegas, G. M. d. S. Menezes, and E. M. L. Aquino, "Desigualdades de gênero e raça na pandemia de covid-19: Implicações para o controle no brasil," Saúde Debate, 2020.

[66] R. G. d. Oliveira, C. R. Rocha, L. M. d. Souza, M. F. Bezerra, L. N. B. Porto, I. d. A. Siqueira, and D. A. Santos, "Desigualdades raciais e a morte como horizonte: Considerações sobre a covid-19 e o racismo estrutural," Cadernos de Saúde Pública, vol. 36, no. 9, pp. 1–14, 2020.

[67] L. Ramalho, Fluent Python, 2nd ed. Sebastopol: O'Reilly Media, 2022.

[68] T. E. Oliphant, A Guide to NumPy. USA: Trelgol Publishing, 2006.

[69] W. McKinney, "Data structures for statistical computing in python," in Proceedings of the 9th Python in Science Conference, 2010, pp. 51–56.

[70] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10). Berkeley, CA, USA: USENIX Association, 2016, pp. 10–10.

APÊNDICE A
TERMOS IMPORTANTES

1) Python - É uma linguagem de programação de alto nível, interpretada, amplamente utilizada em diversas áreas, desde o desenvolvimento web até a análise de dados e ciência de dados [67].
2) NumPy (Numerical Python) - É uma biblioteca fundamental para computação científica em Python, que oferece suporte a arrays multidimensionais e uma ampla coleção de funções matemáticas para operar sobre esses arrays de maneira eficiente [68].
3) Pandas - É uma biblioteca de código aberto em Python voltada para a manipulação e análise de dados, especialmente em formato tabular [69].
4) Scikit-learn - É uma biblioteca de machine learning de código aberto para Python, que oferece ferramentas simples e eficientes para análise preditiva de dados [51].
5) Apache Spark - É um framework de computação distribuída de código aberto, otimizado para processar grandes volumes de dados de forma rápida e eficiente [70].

APÊNDICE B
DICIONÁRIO DE DADOS

Tabela II. FEATURES UTILIZADAS NO ESTUDO

Nome da Feature	Descrição	Tipo de Dados	Categoria	Notas / Possíveis Valores
SEXO	Sexo do indivíduo.	Categórica	Categórica	Masculino/Feminino ou valores binários.
RACA/COR	Raça ou cor do indivíduo.	Categórica	Categórica	Branco, Preto, Pardo, Indígena, etc.
renda_per_capita	Renda per capita do CEP do indivíduo.	Numérica	Numérica	Valor numérico representando a renda per capita.
POSTCODE	Código postal/CEP do endereço.	Categórica	Categórica	Código do endereço.
LON	Longitude do endereço.	Numérica	Numérica	Coordenada decimal.
LAT	Latitude do endereço.	Numérica	Numérica	Coordenada decimal.
densid_pop	Densidade populacional.	Numérica	Numérica	Pessoas por km².
cd_geocodi	Identificador de geocódigo da área.	Numérica	Numérica	Código geográfico único.
DOR DE GARGANTA	Indica se há dor de garganta.	Categórica	Categórica	Sim/Não.
DISPNEIA	Indica dificuldade para respirar.	Categórica	Categórica	Sim/Não.
FEBRE	Indica febre.	Categórica	Categórica	Sim/Não.
TOSSE	Indica tosse.	Categórica	Categórica	Sim/Não.
E PROFISSIONAL DE SAUDE?	Indica se é profissional de saúde.	Categórica	Categórica	Sim/Não.
DOENCAS RESPIRATORIAS CRONICAS DESCOMPENSADAS	Indica doenças respiratórias crônicas.	Categórica	Categórica	Sim/Não.
DOENCAS CARDIACAS CRONICAS	Indica doenças cardíacas crônicas.	Categórica	Categórica	Sim/Não.
DIABETES	Indica presença de diabetes.	Categórica	Categórica	Sim/Não.
DOENCAS RENAIS CRONICAS EM ESTAGIO AVANCADO	Indica doenças renais avançadas.	Categórica	Categórica	Sim/Não.
IMUNOSSUPRESSAO	Indica estado de imunossupressão.	Categórica	Categórica	Sim/Não.
GESTANTE DE ALTO RISCO	Indica gestante de alto risco.	Categórica	Categórica	Sim/Não.
DOENCAS CROMOSSOMICAS OU FRAGILIDADE IMUNOLOGICA	Indica doenças cromossômicas ou fragilidade imunológica.	Categórica	Categórica	Sim/Não.

Tabela III. CÓDIGOS E SIGNIFICADOS DA VARIÁVEL RACA/COR

Código	Descrição
1	Branca
2	Preta
3	Parda
4	Amarela (Asiática)
5	Indígena
9	Ignorado / Não informado

APÊNDICE C
IMAGENS REDIMENSIONADAS

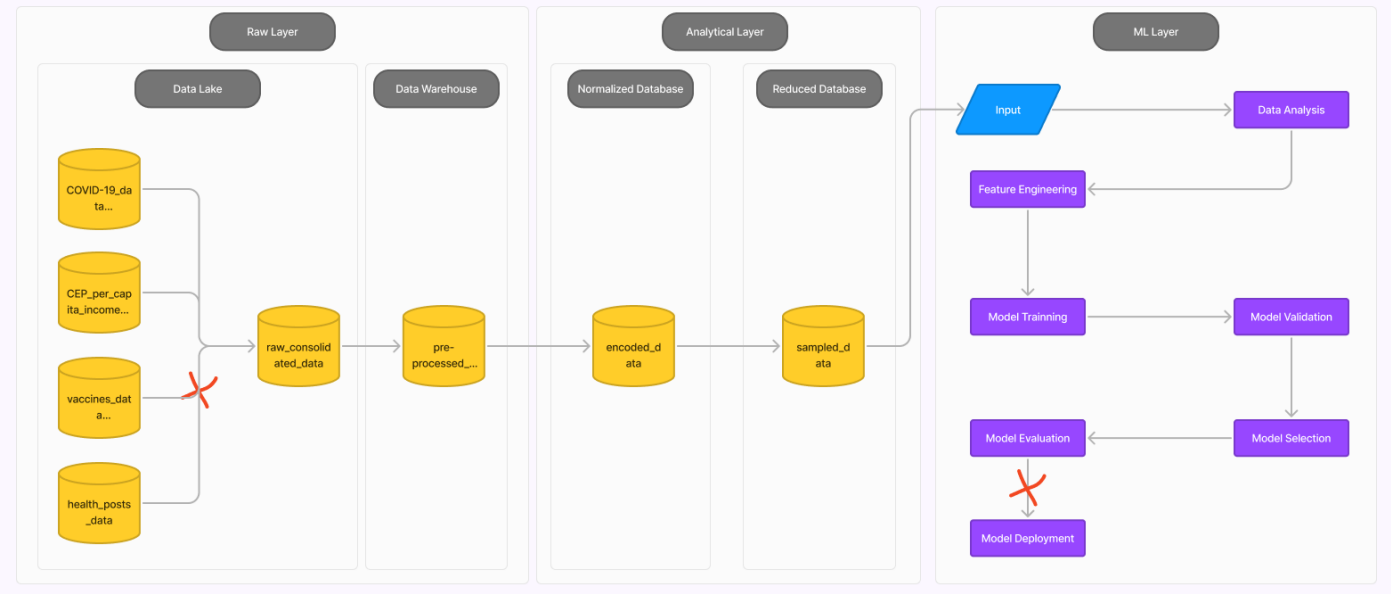

Figura 30. Dataflow do projeto

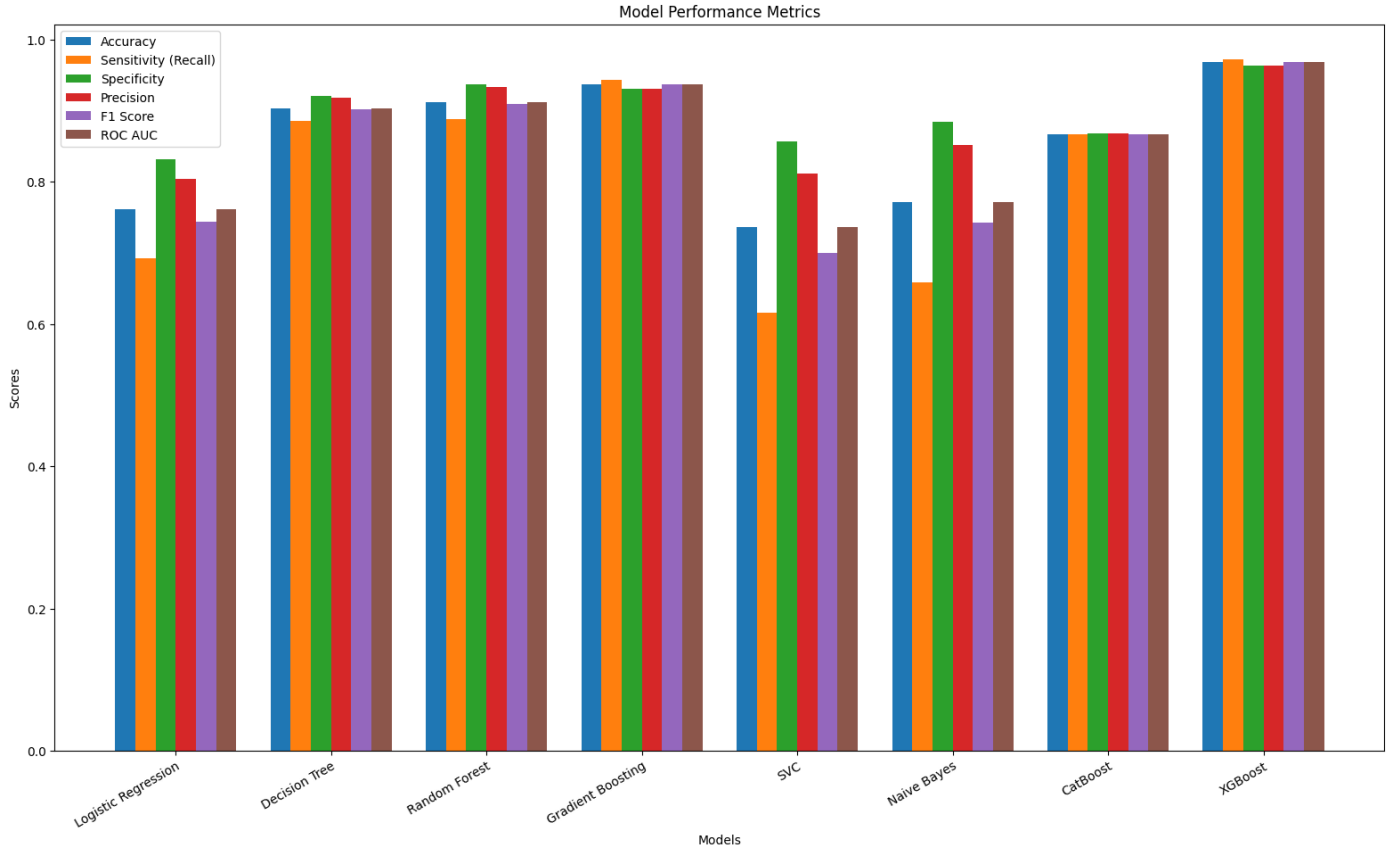

Figura 31. Comparação de métricas por modelo.

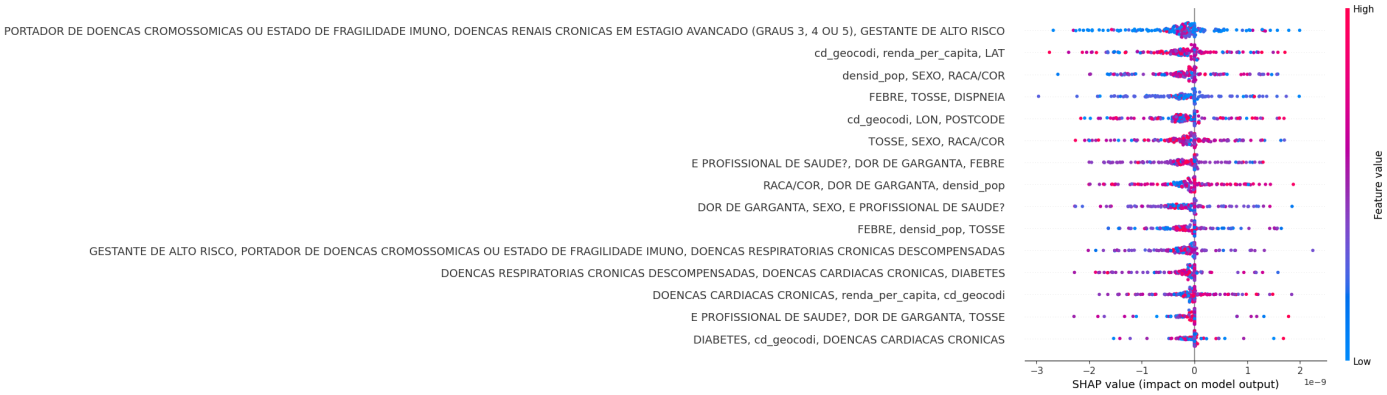

Figura 32. Análise SHAP

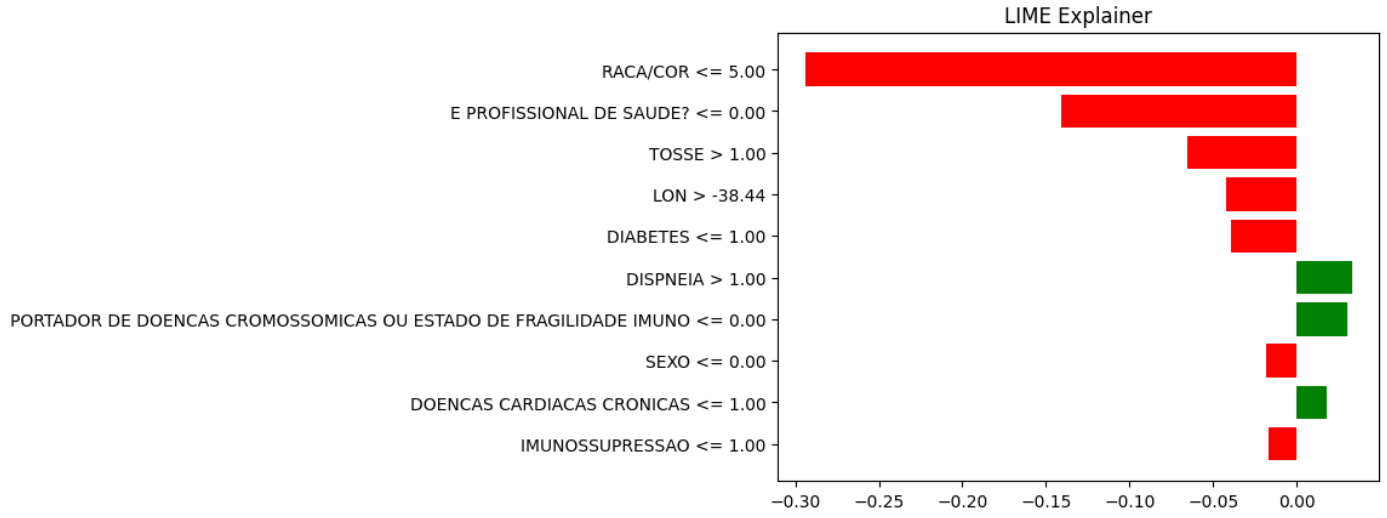

Figura 33. Análise com LIME

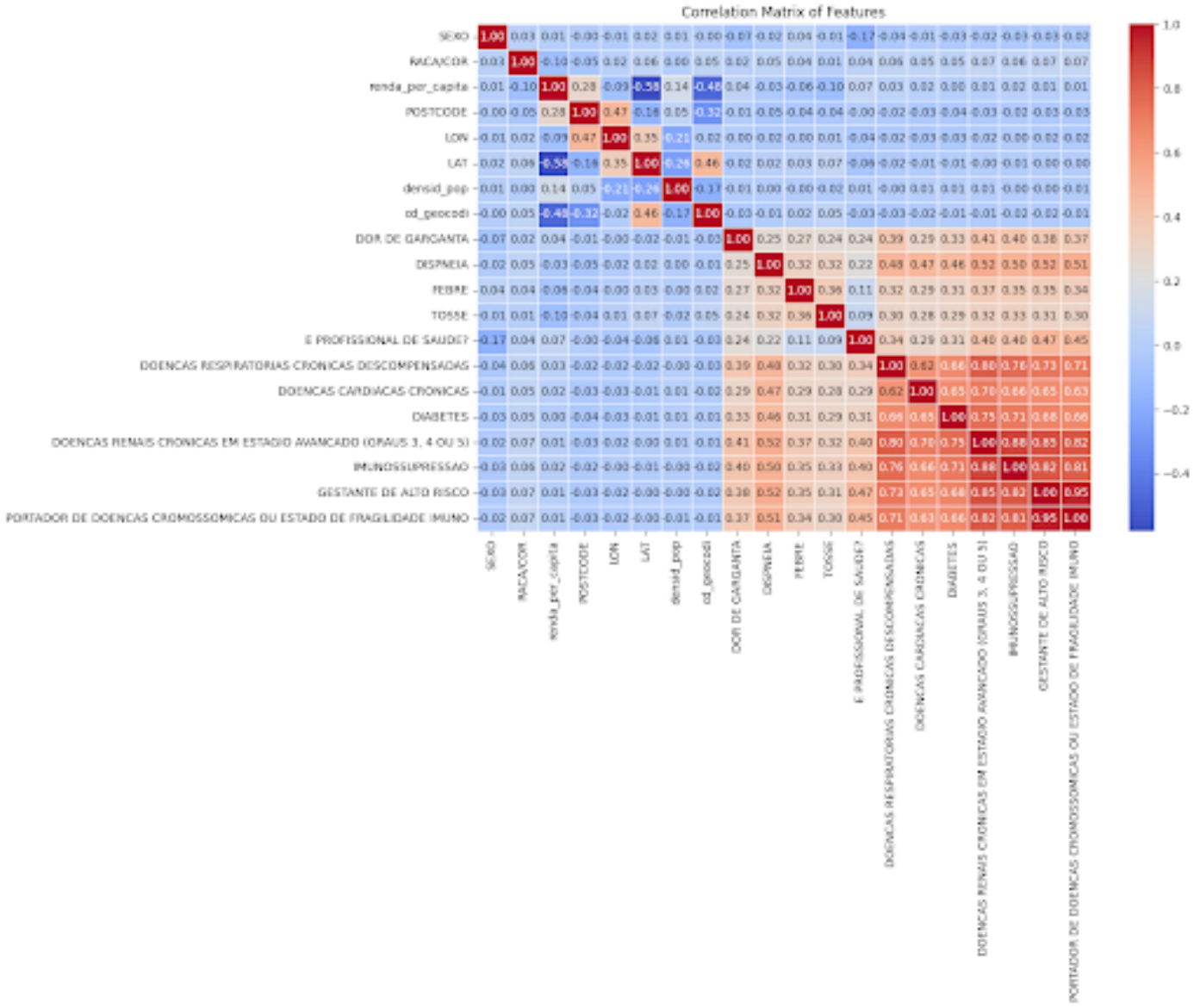

Figura 34. Matriz de correlação