UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA – PECE

RAFAEL HUMMEL SANTOS

ANÁLISE DE DESEMPENHO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE PROPENSÃO À CONTRATAÇÃO DE CRÉDITO PESSOAL

São Paulo

2024

RAFAEL HUMMEL SANTOS

ANÁLISE DE DESEMPENHO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE PROPENSÃO À CONTRATAÇÃO DE CRÉDITO PESSOAL

Trabalho de Conclusão de Curso da Universidade de São Paulo como requisito para a certificação do curso de Especialização em Engenharia de Dados e Big Data

Orientadora: Dra. Marina Jeaneth Machicao Justo

São Paulo

2024

RESUMO

O mercado financeiro tem evoluído rapidamente, impulsionado pelo surgimento de novos bancos, *fintechs* e instituições de crédito, o que aumenta significativamente a concorrência. Nesse contexto, identificar clientes com maior propensão à contratação de produtos financeiros é uma estratégia crucial para reduzir custos, aumentar a eficiência e direcionar campanhas de marketing com maior precisão. Este trabalho teve como objetivo avaliar e comparar modelos supervisionados de aprendizado de máquina na predição dessa propensão, utilizando algoritmos como Regressão Logística, Árvores de Decisão, *Random Forest*, XGBoost e CatBoost. Os modelos foram aplicados a um conjunto de dados de campanhas de prospecção ativa, e técnicas como SMOTE e *undersampling* foram implementadas para lidar com o desbalanceamento das classes. As métricas utilizadas para avaliação incluíram *recall*, acurácia e F1-score. Entre os modelos analisados, o CatBoost ajustado se destacou por apresentar o melhor equilíbrio entre *recall* e acurácia, permitindo identificar mais clientes propensos sem prejudicar significativamente a precisão. O estudo demonstra que o aprendizado de máquina pode ser uma ferramenta valiosa para o setor financeiro.

Palavras-chave: Aprendizado de Máquina, Mercado Financeiro, Crédito Pessoal, Modelo de Propensão

ABSTRACT

The financial market has been rapidly evolving, driven by the emergence of new banks, fintechs, and credit institutions, which significantly increases competition. In this context, identifying customers with a higher propensity to adopt financial products is a crucial strategy to reduce costs, enhance efficiency, and target marketing campaigns more precisely. This study aimed to evaluate and compare supervised machine learning models for predicting this propensity, utilizing algorithms such as Logistic Regression, Decision Trees, Random Forest, XGBoost, and CatBoost. The models were applied to a dataset of active prospecting campaigns, and techniques such as SMOTE and undersampling were implemented to address class imbalance. The evaluation metrics included recall, accuracy, and F1-score. Among the analyzed models, the adjusted CatBoost stood out for achieving the best balance between recall and accuracy, allowing for the identification of more potential customers without significantly compromising precision. The study demonstrates that machine learning can be a valuable tool for the financial sector.

Keywords: Machine Learning, Financial Market, Personal Loan, Propensity Model

Sumário

1. INTRODUÇÃO

1.1 CONTEXTO

A teoria bancária tradicional defende que os bancos devem diversificar seus portfólios de crédito para reduzir a probabilidade de prejuízos (DIAMOND, 1984 apud TABAK et al., 2011). Essa diversificação se torna essencial em um cenário de reestruturação do setor financeiro devido à entrada de novas empresas de tecnologia financeira, como bancos digitais, *fintechs* e grandes companhias (AZEVÊDO; GARTNER, 2020). Essas mudanças intensificam a concorrência no mercado de crédito, obrigando os bancos tradicionais a adotar estratégias inovadoras.

Além disso, os avanços recentes em ciência de dados aprimoraram a capacidade de transformar dados em informações relevantes (COŞER et al., 2020). A análise de consumidores tem um papel crucial para as instituições financeiras, permitindo que produtos e serviços sejam mais bem direcionados, o que melhora a experiência e a satisfação do cliente. Nesse contexto, tecnologias financeiras inovadoras e análises robustas são fundamentais para enfrentar a concorrência e atender às expectativas dos consumidores.

1.2 PROBLEMA DE PESQUISA

Diante do aumento da concorrência no setor financeiro e da evolução tecnológica, surge a necessidade de avaliar quais modelos de aprendizado de máquina podem ser mais eficientes na predição de comportamentos dos consumidores, como a propensão à contratação de crédito pessoal. A escolha do algoritmo adequado torna-se um desafio, já que diferentes modelos apresentam vantagens e limitações dependendo do tipo de problema (MAHESH, 2020).

1.3 JUSTIFICATIVA

A análise de propensão é um diferencial essencial para instituições financeiras e equipes de Marketing e CRM (*Customer Relationship Management*), pois permite aumentar o retorno das campanhas, melhorar a satisfação do cliente e reduzir *opt-outs* (solicitações de clientes para não receber comunicações) de consumidores não interessados. Diante da crescente competitividade no setor bancário e do avanço tecnológico, torna-se fundamental compreender o comportamento dos clientes para gerar ofertas personalizadas. Embora exista uma vasta literatura sobre modelos preditivos em diversos setores (MAIA et al., 2024; KOHNEHSHAHRI et al., 2024), ainda há espaço para aprofundar o tema, especialmente em sua aplicação em marketing e CRM.

1.4 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

Objetivo Geral:

Avaliar e comparar o desempenho de diferentes modelos de aprendizado de máquina na predição da propensão à contratação de crédito pessoal.

Objetivos Específicos:

- Implementar e treinar cinco modelos distintos de aprendizado de máquina.
- Comparar a performance dos modelos com base em métricas específicas.

- Identificar o modelo mais eficiente e adequado ao problema.

1.5 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos principais, conforme detalhado a seguir:

Capítulo 1 - Introdução: Apresenta o contexto do estudo, o problema de pesquisa, as justificativas para sua realização e os objetivos gerais e específicos. Finaliza com a descrição da estrutura do trabalho, orientando o leitor sobre o conteúdo dos capítulos subsequentes.

Capítulo 2 - Marco Teórico: Aborda os conceitos fundamentais para a compreensão do tema. Inicialmente, apresenta a ciência e engenharia de dados, destacando sua definição, importância e etapas do processo. Em seguida, discute os fundamentos do aprendizado de máquina, incluindo amostragem de dados e técnicas de avaliação de modelos. Por fim, descreve os principais métodos de aprendizado de máquina utilizados neste estudo, como Regressão Logística Binária, Árvore de Decisão, XGBoost e CatBoost.

Capítulo 3 - Proposta: Detalha a metodologia adotada para o estudo, começando pela descrição do conjunto de dados utilizado, seguida pelos processos de ingestão, carregamento, análise exploratória e pré-processamento, incluindo limpeza de dados e transformações. Apresenta ainda os procedimentos de modelagem de dados, métodos de avaliação, visualização e comunicação dos resultados, além da arquitetura utilizada.

Capítulo 4 - Resultados e Discussões: Apresenta os resultados obtidos com a implementação e treinamento dos modelos, realizando uma análise comparativa detalhada entre eles. Discute o ajuste dos modelos pré-selecionados, a seleção do melhor modelo e as implicações práticas dos resultados.

Capítulo 5 - Conclusões: Conclui o trabalho com um resumo dos principais insights obtidos, as limitações encontradas durante o estudo e sugestões para trabalhos futuros que possam expandir ou aprofundar as análises realizadas.

2. MARCO TEÓRICO

O aprendizado de máquina engloba um conjunto de técnicas amplamente utilizadas para resolver diversos problemas do mundo real com o auxílio de sistemas computacionais que aprendem a solucionar desafios sem a necessidade de serem explicitamente programados para isso (KÜHL et al., 2019). Os autores afirmam que essas técnicas buscam, em essência, construir um modelo aplicando algoritmos sobre um conjunto de dados conhecido, com o objetivo de obter insights ou realizar previsões em novos conjuntos de dados. Embora o processo de criação de um modelo de aprendizado de máquina possa variar em termos de fases, ele geralmente abrange três etapas principais: inicialização do modelo, estimativa de desempenho e implantação.

2.1 CIÊNCIA E ENGENHARIA DE DADOS

2.1.1 DEFINIÇÃO E IMPORTÂNCIA

A ciência de dados é um campo interdisciplinar que envolve a extração e a apresentação de insights a partir de dados, utilizando processos de coleta, armazenamento, acesso, análise e comunicação (CADY, 2017). De acordo com os autores, a área abrange competências diagnósticas, descritivas e preditivas, possibilitando que gestores e usuários compreendam os acontecimentos, suas causas e como lidar com as consequências previstas.

Nesse contexto, a engenharia de dados envolve o desenvolvimento, implementação e manutenção de sistemas que transformam dados brutos em informações de alta qualidade e confiabilidade, aplicáveis em diversos contextos, como aprendizado de máquina, análise em tempo real e previsões (REIS; HOUSLEY, 2022). Essa prática é fundamental para garantir que *stakeholders*, como gestores, analistas e desenvolvedores, tenham acesso a informações de forma segura e eficiente. Por meio do desenvolvimento de pipelines e algoritmos, a engenharia de dados organiza os dados para uso imediato, promovendo decisões estratégicas e a integração de tecnologias avançadas, além de assegurar a gestão eficaz do ciclo de vida dos dados (REIS; HOUSLEY, 2022).

2.1.2 ETAPAS DO PROCESSO DE CIÊNCIA DE DADOS

Conforme Cady (2017), as etapas do processo de ciência de dados consistem em formular o problema, compreender os dados, selecionar as características (*features*), treinar o modelo e realizar a análise, apresentar os resultados e implementar o código em produção. Esse processo é cíclico, ou seja, após a avaliação do modelo, pode ser necessário retornar às etapas iniciais, reformular o problema e repetir o ciclo. A Figura 1 apresenta uma imagem adaptada do processo.

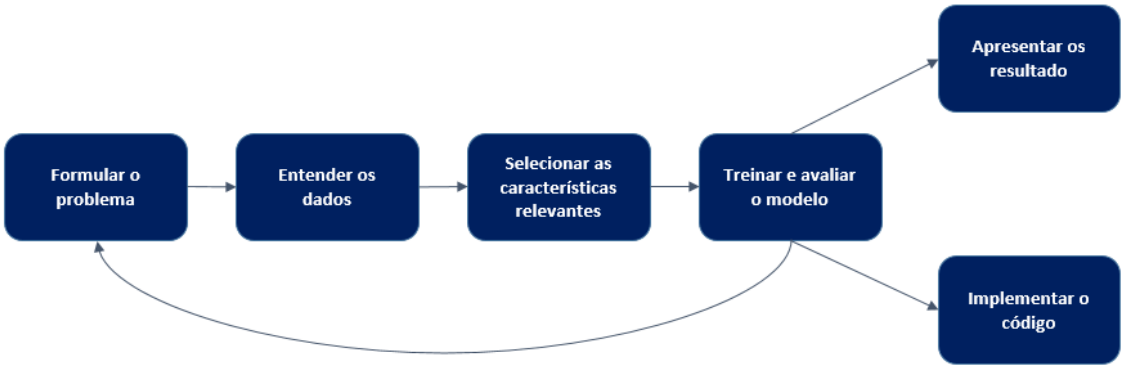

Figura 1 – Etapas do processo de Ciência de Dados Fonte: Adaptado de CADY (2017).

Entre as etapas, a preparação dos dados pode ser uma das etapas mais difíceis de qualquer projeto de aprendizado de máquina, devido às especificidades de cada conjunto de dados em relação ao projeto. Em modelos preditivos, dados crus tipicamente não podem ser utilizados diretamente. Algumas razões para isso incluem: a necessidade de que todos os dados sejam numéricos para algoritmos de aprendizado de máquina; a presença de erros estatísticos, como *outliers* e valores inválidos; e erros relacionados a valores nulos ou categorias inválidas. Por esse motivo, torna-se indispensável a realização de uma etapa de preparação dos dados, que pode englobar limpeza de dados, pré-processamento e engenharia de atributos (BROWNIEE, 2020).

O autor lista algumas das etapas de preparação de dados e suas definições:

- Limpeza de dados: consiste em identificar e corrigir inconsistências e erros nos dados.
- Seleção de atributos: processo de identificar as variáveis independentes mais relevantes para o modelo.
- Transformação de dados: alteração da escala ou distribuição das variáveis.
- Engenharia de atributos: criação de novas variáveis a partir de variáveis pré-existentes no conjunto de dados.
- Redução de dimensionalidade: redução do número de atributos utilizados no modelo.

Essas técnicas são fundamentais para garantir que os dados estejam adequados para os algoritmos de aprendizado, que dependem de padrões extraídos do conjunto para realizar previsões e classificações. No entanto, em alguns casos, o desempenho dos classificadores depende não apenas do aprendizado do algoritmo, mas também da composição das classes no *dataset* (PRADIPTA et al., 2021). Segundo os autores, conjuntos de dados desbalanceados impactam negativamente o aprendizado do modelo, pois a classe majoritária possui uma quantidade significativamente maior de exemplos em comparação à classe minoritária. Quando uma das classes apresenta uma frequência menor de ocorrência, é mais provável que seja tratada como ruído, *outlier* ou que sofra uma classificação inadequada em relação à classe majoritária.

Nesse contexto, técnicas de balanceamento que utilizam *undersampling* realizam a redução de amostras da classe majoritária com o objetivo de equilibrar a distribuição entre as classes do *dataset*. O método mais comum para essa abordagem é o *Random Undersampling* (RUS), que remove de forma aleatória os registros da classe majoritária com base em uma taxa de amostragem previamente definida. Contudo, essa técnica pode levar à perda de informações importantes, uma vez que os registros eliminados não consideram sua relevância no conjunto de dados (DEVI et al., 2020).

Já o SMOTE (*Synthetic Minority Oversampling Technique*) é uma técnica de *oversampling* utilizada para balancear conjuntos de dados desbalanceados por meio da criação de exemplos sintéticos da classe minoritária. Em vez de simplesmente replicar os registros existentes, como ocorre em algumas abordagens de *oversampling*, o SMOTE gera novos exemplos interpolando os dados da classe minoritária (PRADIPTA et al., 2021).

2.2 FUNDAMENTOS DO APRENDIZADO DE MÁQUINA

Técnicas de aprendizado de máquina são amplamente utilizadas para construir modelos que utilizam informações extraídas de dados brutos para prever padrões ou dados desconhecidos (RAMACHANDRAN et al., 2024).

Os algoritmos de aprendizado de máquina podem ser classificados em duas categorias principais de resolução de problemas: preditivos e descritivos. A abordagem preditiva utiliza algoritmos treinados em conjuntos de dados rotulados para gerar modelos capazes de estimar valores da variável alvo com base nos atributos preditores dos dados (FACELI et al., 2021). Por outro lado, na abordagem descritiva, o foco é identificar padrões ou estruturas nos dados, sem a necessidade de variáveis alvo previamente definidas. Esse tipo de tarefa utiliza o paradigma de aprendizado não supervisionado, em que o algoritmo opera sem conhecimento prévio sobre os dados (FACELI et al., 2021).

Além dessas perspectivas, os métodos de aprendizado de máquina podem ser classificados em quatro categorias principais: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço, conforme descrito por Rincy e Gupta (2020) e ilustrado na Figura 2. Essas categorias abrangem diferentes paradigmas e estratégias para abordar os desafios de análise e modelagem de dados.

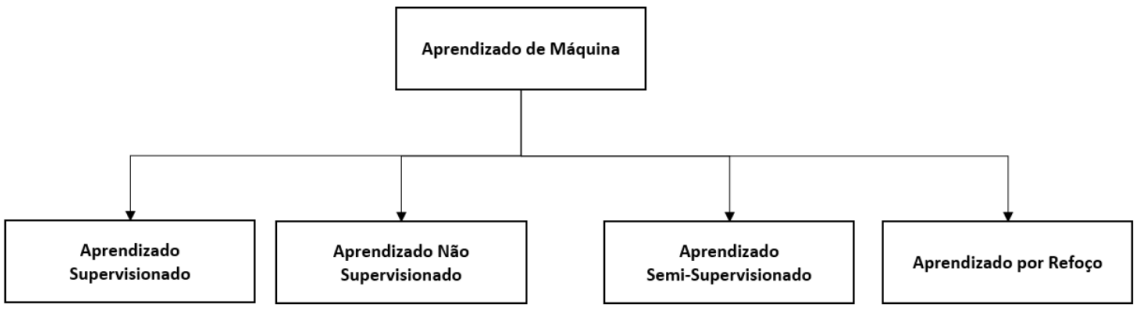

Figura 2 – Tipos de aprendizado de máquina Fonte: Adaptado de RINCY & GUPTA (2020).

O ciclo de vida dos modelos de aprendizado de máquina pode ser dividido em quatro etapas principais: gerenciamento de dados, aprendizado do modelo, verificação do modelo e *deployment*. As três primeiras estão diretamente relacionadas à criação do modelo (ASHMORE et al., 2021). O gerenciamento de dados, por sua vez, envolve dois processos fundamentais. O primeiro é a coleta de dados, que consiste na identificação da população, do fenômeno ou do evento-alvo, bem como das características e rótulos associados. Como geralmente não é viável incluir toda a população, utiliza-se uma amostra representativa (SURESH; GUTTAG, 2021). O segundo é o pré-processamento, que, segundo Suresh e Guttag (2021), depende do tipo de problema a ser resolvido. Exemplos de atividades incluem lidar com dados ausentes, simplificar variáveis, normalizar dados contínuos e aplicar técnicas como *one-hot encoding* e *label encoding*.

Na etapa de aprendizado, inicia-se o processo de escolha do modelo, considerando fatores como o tipo de problema (regressão ou classificação), o volume de dados disponíveis e a estrutura do treinamento (ASHMORE et al., 2021). Por sua vez, a etapa de verificação tem como principal desafio garantir que o modelo treinado tenha boa capacidade de generalização, ou seja, que performe bem em novos dados. Nessa fase, o modelo é testado em um conjunto de dados de verificação, previamente separado na etapa de gerenciamento de dados e mantido independente do conjunto de treinamento.

2.2.1 AMOSTRAGEM DE DADOS

A amostragem de dados é um conjunto de métodos que facilitam a seleção de informações relevantes em grandes volumes de dados, permitindo o processamento de subconjuntos menores sem comprometer a representatividade das informações essenciais (BAYRAKTAR; ERDEM, 2023). De acordo com os autores, trata-se de uma área de pesquisa que varia conforme o problema estudado, mas que ocorre após a definição adequada do problema. Para que a resolução seja eficaz, é fundamental que o conjunto de dados utilizado seja coletado corretamente. Ademais, as amostras extraídas devem representar fielmente o todo. A etapa mais importante no processo de amostragem é a determinação da população-alvo, que inclui todos os registros que afetam diretamente os resultados do sistema e, ao mesmo tempo, são influenciados por eles (SURESH; GUTTAG, 2021).

2.2.2 TÉCNICAS DE AVALIAÇÃO DE MODELOS

As técnicas de avaliação de modelos de aprendizado de máquina são essenciais para verificar a eficácia, a capacidade de generalização e a adequação dos modelos aos problemas específicos em que são aplicados, uma vez que, de maneira geral, não existe técnica universal capaz de garantir o melhor desempenho em qualquer tipo de problema (FACELI et al., 2021). Assim, são amplamente utilizadas práticas como a divisão dos dados em conjuntos de treino, validação e teste, além da validação cruzada para garantir maior consistência nos resultados.

2.2.2.1 MÉTODOS DE AVALIAÇÃO

As métricas avaliadas ao longo do trabalho foram acurácia, precisão, sensibilidade, f1-score, Curva ROC, em conjunto com a técnica de validação cruzada e o método *hold-out*. A validação cruzada avalia a performance de modelos de aprendizado de máquina, dividindo o conjunto de dados original em subconjuntos de treinamento e

teste. Pode ser exaustiva, considerando todas as possíveis combinações de divisão dos dados, ou não exaustiva, como no método *hold-out*, que realiza uma divisão única, geralmente alocando 70% dos dados para o treinamento e 30% para o teste (SERAJ et al., 2023). Métodos mais avançados de validação cruzada incluem múltiplos subconjuntos de dados, oferecendo avaliações mais robustas do desempenho do modelo.

Métricas de Avaliação de Modelos

Acurácia (Accuracy): Avalia a proporção de previsões corretas em relação ao total de previsões. O numerador corresponde a soma dos valores verdadeiros positivos (VP) e verdadeiros negativos (VN), enquanto o denominador é a soma VP, VN, falsos positivos (FP) e falsos negativos (FN).

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \tag{3}$$

Precisão (Precision): Avalia a proporção de verdadeiros positivos em relação a todas as observações previstas como positivas.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{4}$$

Sensibilidade (*Recall*): Mede a proporção de verdadeiros positivos em relação a todos os registros que são realmente positivos.

$$\text{Sensibilidade} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{5}$$

F1-Score: Obtido por meio da média harmônica entre precisão e sensibilidade, é útil para balancear as métricas em casos de classes desbalanceadas.

$$\text{F1} - \text{Score} = 2 \text{ x } \frac{\text{Precisão x Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \tag{6}$$

A Curva ROC (*Receiver Operating Characteristic*) é amplamente utilizada para avaliar o desempenho de um modelo preditivo, comparando suas predições com os resultados reais. Um dos principais benefícios da análise ROC é sua independência em relação a um limiar específico, permitindo avaliar o desempenho do modelo em diferentes *thresholds* e auxiliar na escolha do limiar ideal com base em uma função de custo ou objetivo (MUSCHELLI, 2020). A Curva ROC representa a relação entre a sensibilidade (taxa de verdadeiros positivos) e a especificidade (1 – taxa de falsos positivos) em diferentes limiares. A área sob a curva (AUC – *Area Under the Curve*) resume a capacidade preditiva geral do modelo.

2.3 MÉTODOS DE APRENDIZADO DE MÁQUINA

Nesta seção, serão apresentados e discutidos alguns dos principais modelos de aprendizado de máquina utilizados para análises preditivas, com ênfase em técnicas amplamente aplicadas em diferentes áreas, como finanças, marketing e medicina. Para a

criação dos modelos de Regressão Logistica, Árvore de decisão e *Random Forest*, foram utilizadas classes da biblioteca Scikit-learn. Os parâmetros de cada modelo estão detalhados no Apêndice A.

2.3.1 REGRESSÃO LOGÍSTICA BINÁRIA

A análise de regressão é uma técnica estatística amplamente utilizada para investigar e modelar a relação entre variáveis. Ela é particularmente relevante na área de mineração de dados, sendo uma ferramenta fundamental na ciência de dados (MONTGOMERY et al., 2021).

Este método visa determinar se existe uma relação causal entre a variável dependente e uma ou mais variáveis independentes, que podem ser contínuas, discretas ou mistas. A Regressão Logística é especialmente eficaz quando a variável dependente é categórica e binária, como nos casos em que se deseja classificar dados em duas categorias, por exemplo, cliente propenso ou não propenso para tomar crédito, ou produção de milho alta e baixa (SALAM et al., 2024). A aplicação deste modelo permite examinar o impacto de diversas variáveis independentes sobre a variável dependente de natureza binária, proporcionando uma análise robusta em contextos de previsão.

O modelo estatístico baseia-se na função logística, também conhecida como função logit (Equação 1), para estabelecer a relação entre as variáveis independentes e a variável dependente. Essa função resulta em uma curva sigmoide (em forma de S que mapeia a probabilidade da variável dependente assumir o valor 1 em um intervalo contínuo entre 0 e 1. Esse modelo considera uma relação linear entre as variáveis independentes por meio de seus coeficientes, permitindo estimar as probabilidades associadas à variável dependente (MONTGOMERY; RUNGER, 2018).

$$f(x) = \frac{1}{1 + \exp[-(\beta_0 + \beta_1 x)]} \quad (1)$$

A curva sigmoide ilustrada na Figura 3 evidencia a natureza não linear do modelo, característica essencial para abordar a restrição de que as probabilidades estejam limitadas ao intervalo [0, 1]. A regressão logística é amplamente utilizada em problemas com variáveis binárias, como na medicina para prever a probabilidade de uma condição clínica, no marketing para segmentar clientes, e nas finanças para estimar o risco de crédito (MONTGOMERY; RUNGER, 2018). Esses exemplos reforçam sua versatilidade como uma técnica robusta para modelagem de eventos binários influenciados por múltiplos fatores.

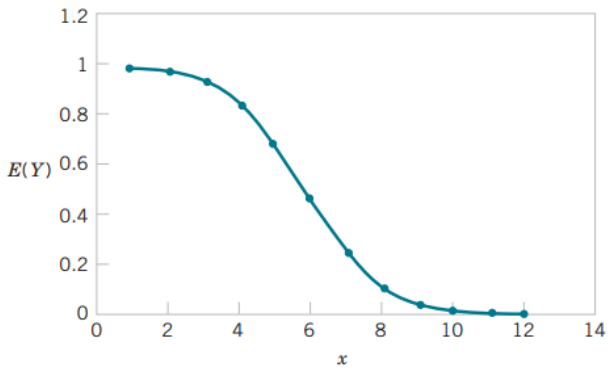

Figura 3 – Função logística Fonte: MONTGOMERY; RUNGER, 2018, (p. 306).

2.3.2 ÁRVORE DE DECISÃO

Árvores de decisão são estruturas hierárquicas que empregam a estratégia de dividir para conquistar, dividindo problemas complexos em partes menores para resolver questões de classificação e regressão. Durante esse processo, o espaço de instâncias é segmentado em subespaços por meio de regras definidas pelos atributos dos dados. Nos nós de divisão, são realizados testes condicionais, enquanto as folhas da árvore representam funções que minimizam os custos associados, como a moda em problemas de classificação ou a média em casos de regressão (FACELI et al., 2021).

A utilização de árvores de decisão apresenta as seguintes vantagens: não exige nenhuma distribuição específica dos dados; os atributos podem ser qualitativos ou quantitativos; é possível construir modelos para qualquer função, desde que o número de exemplos de treinamento seja suficiente; e oferece um elevado grau de interpretabilidade (LEMOS et al., 2005).

A Figura 4 apresenta as primeiras camadas de uma das árvores de decisão utilizadas neste trabalho.

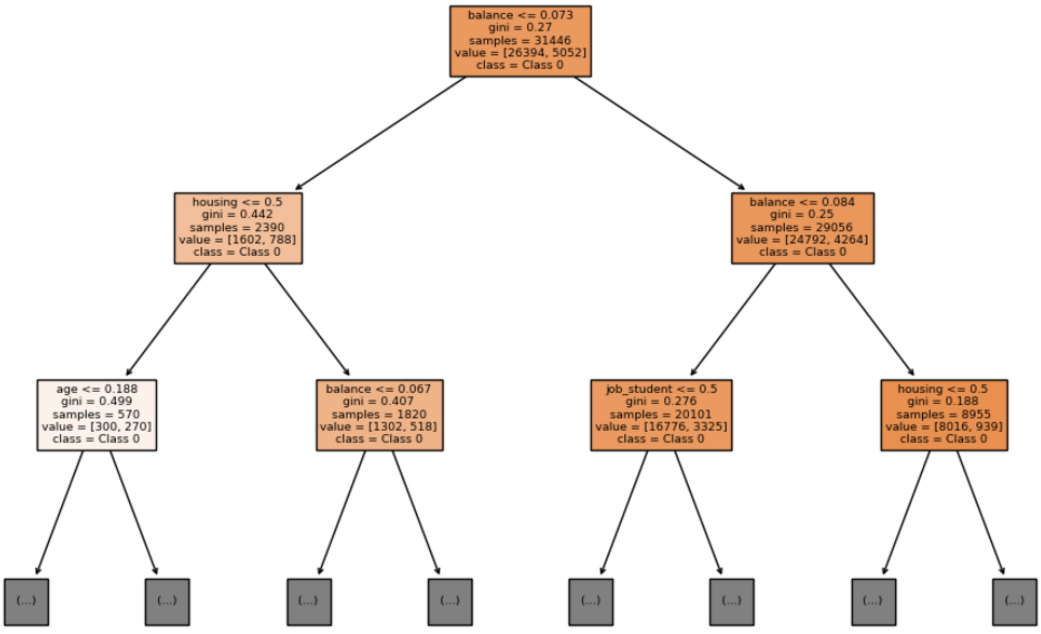

Figura 4 – Ilustração de uma árvore de decisão e seus 2 primeiros níveis Fonte: Elaborado pelo Autor.

Nesta figura, cada retângulo contém 5 estatísticas do nó, o atributo selecionado, gini, números de amostras (*samples*), distribuição dos dados (*value*) e a probabilidade da classe predominante (*class*). Por exemplo, a primeira linha no retângulo indica o atributo utilizado para dividir os dados. Neste caso, a variável *balance* foi utilizada inicialmente para separar os dados com base no critério *balance* <= 0.073. A linha subsequente, *gini*, representa a impureza do nó, indicando uma mistura moderada das duas classes (Classe 0 e Classe 1). A terceira linha, *samples*, mostra o número total de amostras presentes neste nó, que neste caso é 31.446. A linha *value* indica a quantidade de amostras pertencentes a cada classe, sendo 26.394 amostras da Classe 0 e 5.052 da Classe 1. Por fim, a linha *class* mostra a classe predominante no nó, que é a Classe 0, ou seja, a classe que o modelo prevê para este nó.

2.3.3 RANDOM FOREST

O princípio geral da Random Forest é combinar um conjunto de árvores de decisão geradas de forma aleatória para formar um modelo robusto. Diferentemente do método CART, que constrói uma única árvore determinística utilizando todo o conjunto de dados, o algoritmo da Random Forest gera múltiplos preditores, que não precisam ser individualmente ótimos. Essa abordagem explora de maneira mais ampla o espaço de possibilidades dos modelos preditivos, resultando, na prática, em um desempenho preditivo superior (GENUER; POGGI, 2020).

O processo de construção dessas árvores utiliza a técnica de *bagging* (*bootstrap aggregating*), que consiste em treinar cada árvore com um subconjunto aleatório de dados gerado por amostragem com reposição (*bootstrap sampling*). Além disso, em cada divisão de uma árvore, apenas um subconjunto aleatório de variáveis é considerado, o que reduz a correlação entre as árvores e amplia a diversidade do modelo. Essa estratégia de aleatoriedade permite que as previsões individuais das árvores sejam menos dependentes umas das outras (BREIMAN, 2001). O resultado final é obtido por meio de um *ensemble*, onde as árvores combinam suas previsões (por votação no caso de classificação ou média no caso de regressão), garantindo maior precisão e menor risco de *overfitting*. Adicionalmente, o desempenho do modelo pode ser avaliado utilizando os exemplos não selecionados em cada amostragem (dados *out-of-bag*), permitindo o cálculo do erro geral sem a necessidade de um conjunto de validação separado. Essa característica torna a Random Forest eficiente tanto na construção quanto na avaliação de modelos preditivos (BREIMAN, 2001).

2.3.4 XGBOOST

O XGBoost é uma ferramenta que se baseia no *gradient boosting tree* (GBT), um método que constrói uma sequência de árvores de decisão. Em cada etapa, o algoritmo adiciona uma nova árvore ao modelo para tentar corrigir os erros das previsões anteriores. Para isso, utiliza-se uma função de perda, que mede a diferença entre as previsões feitas pelo modelo e os valores reais. Uma das vantagens do XGBoost é o uso de uma técnica que considera tanto os gradientes (que indicam a direção do ajuste) quanto a curvatura (que refina esse ajuste), tornando o treinamento mais rápido e o modelo mais preciso (CHEN; GUESTRIN, 2016).

Além disso, o XGBoost se destaca por integrar várias otimizações que melhoram sua eficiência e capacidade de lidar com grandes volumes de dados. Ele foi projetado para trabalhar de forma rápida e escalável, utilizando computação paralela e técnicas como a regularização L1 e L2, que ajudam a evitar *overfitting*. Outro diferencial importante é sua habilidade de lidar com dados incompletos ou esparsos, garantindo bom desempenho mesmo em cenários onde os dados estão longe de ser perfeitos. Essas características tornam o XGBoost uma escolha popular em competições e aplicações reais de aprendizado de máquina (CHEN; GUESTRIN, 2016).

2.3.5 CATBOOST

O CatBoost é um algoritmo baseado em GBT projetado para lidar com variáveis categóricas com mínima perda de informação. Ele se diferencia de outros algoritmos baseados na mesma técnica ao utilizar o *ordered boosting* para evitar vazamento de informação, sendo eficaz inclusive em conjuntos de dados pequenos. Sua abordagem para variáveis categóricas consiste na substituição dessas variáveis por valores numéricos durante o pré-processamento. Além disso, o CatBoost realiza permutações aleatórias para estimar os valores das folhas durante a construção das árvores, mitigando o *overfitting* (PROKHORENKOVA et al., 2021).

O *ordered boosting* é uma técnica do CatBoost que calcula os resíduos de cada exemplo usando um modelo treinado sem incluir esse exemplo. Isso evita que o modelo use informações futuras durante o treinamento, garantindo previsões mais confiáveis e sem vazamento de informação (PROKHORENKOVA et al., 2021).

3. PROPOSTA

O ciclo de vida dos dados inicia-se com a ingestão, onde os dados brutos são baixados de uma fonte externa, extraídos e armazenados na camada *raw*. Em seguida, os dados passam pela etapa de exploração e limpeza, onde são analisadas dimensões, tipos, valores ausentes e distribuições, realizando-se a remoção de valores desconhecidos e imputações baseadas em frequências relativas para garantir consistência. Após a limpeza, ocorre a transformação e pré-processamento, com a codificação de variáveis categóricas, padronização ou normalização de variáveis numéricas e armazenamento dos dados processados na camada *refined*. Na sequência, é realizada a análise estatística e preparação para modelagem, incluindo a geração de histogramas, mapas de calor para avaliação de correlações, ajustes para mitigar multicolinearidade e divisão dos dados em conjuntos de treinamento e teste. Por fim, na etapa de modelagem e avaliação, diferentes algoritmos são aplicados, as métricas de desempenho são calculadas e os melhores modelos são selecionados. Essas etapas estão detalhadas no Apêndice E, que apresenta as implementações realizadas para o ciclo de vida descrito. A Figura 5 ilustra o ciclo de vida proposto.

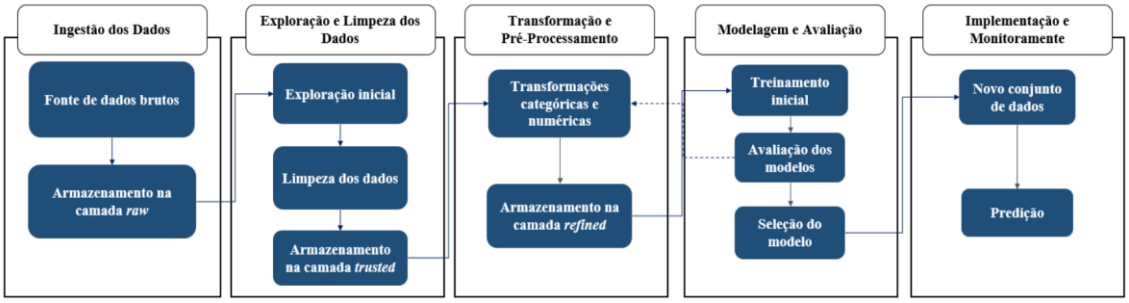

Figura 5 – Ciclo de vida do projeto de Ciência e Engenharia de Dados. Fonte: Elaborado pelo Autor.

3.1 ARQUITETURA E METODOLOGIA

O projeto utilizou duas principais arquiteturas: uma baseada em Data Warehouse, para o armazenamento estruturado dos dados em diferentes etapas de processamento, e outra em camadas semânticas, voltada para a análise e visualização dos dados. A metodologia aplicada seguiu um processo de ETL (*Extract, Transform, Load*), no qual os dados foram extraídos de uma fonte externa, transformados em múltiplas etapas (incluindo limpeza, padronização e *encoding*) e, por fim, carregados em diferentes camadas para armazenamento e análise (REIS; HOUSLEY, 2022). A Figura 6 apresenta a arquitetura e a metodologia propostas, destacando o fluxo desde a ingestão inicial no banco SQLite até a aplicação de um modelo de *machine learning*, com os resultados sendo utilizados para análise no Power BI.

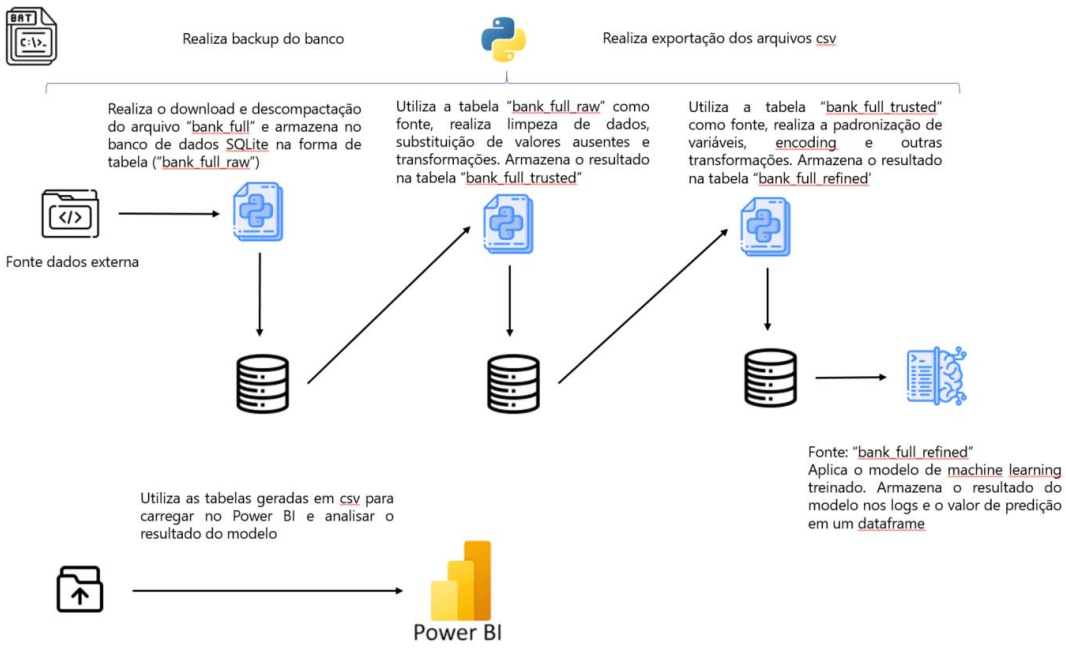

Figura 6 – Arquitetura e metodologia proposta. Fonte: Elaborado pelo Autor.

3.2 DESCRIÇÃO DO CONJUNTO DE DADOS

O conjunto de dados utilizado neste trabalho foi obtido no repositório *UC Irvine Machine Learning Repository* e contém informações sobre campanhas de prospecção ativa, realizadas por meio de ligações telefônicas, de uma instituição bancária. O objetivo inicial do *dataset* é analisar a predisposição dos clientes em contratar o produto ofertado, um depósito a prazo, no qual o cliente deposita uma quantia de dinheiro por um período determinado, sem possibilidade de resgate antecipado, e recebe o montante inicial acrescido de juros ao final do prazo.

Abaixo segue uma representação das variáveis encontradas na tabela, descrição, tipo e exemplos:

Tabela 1 – Dicionário de dados contendo todas as colunas do conjunto de dados. Fonte: Elaborado pelo Autor.

#Dados dos clientes do banco			
Nome da variável	**Descrição**	**Tipo de Dado**	**Valores Possíveis**
age	Idade	Numérico	Entre 18 e 95 anos
job	Emprego	Categórico	"admin.","blue-collar","entrepreneur","housemaid", "management","retired","self-employed","services", "student","technician","unemployed","unknown"
marital	Estado Civil	Categórico	"divorced","married","single","unknown"; Nota: "divorced" pode ser divorced ou widowed
education	Nível de Escolaridade	Categórico	"unknown","secondary","primary","tertiary"
default	Indica se o cliente está inadimplente	Binário	0 e 1
balance	Saldo médio da conta em euros	Numérico	Valores entre € -8.019 e € 102.127
housing	Indica se o cliente possuí empréstimo imobiliário	Binário	0 e 1
loan	Indica se o cliente possuí empréstimo pessoal	Binário	0 e 1
# Dados de contato com o cliente			
contact	Canal de comunicação utilizado	Categórico	"unknown","telephone","cellular"
day	Último dia em que o cliente foi comunicado no mês	Numérico	1 a 31
month	Último mês do ano em que o cliente foi contatado	Categórico	"jan", "feb", "mar", "apr","may","jun", "jul", "aug", "sep", "opt", "nov", "dec"
duration	Duração do último contato com o cliente (em segundos)	Numérico	Entre 0 e 4.918 segundos
# Outros atribudos			
campaign	Número de contatos realizados durante essa campanha	Numérico	Entre 1 e 63 contatos
pdays	Número de dias passado entre a última vez que o cliente foi contatado na campanha anterior	Numérico	Entre -1 e 871; Nota: -1 indica que o cliente não foi contatado anteriormente
previous	Número de contatos realizados para o cliente antes dessa campanha	Numérico	Entre 0 e 275
poutcome	Indica o resultado da última campanha para esse cliente	Categórico	"unknown","other","failure","success"
y	Indica se o cliente submeteu seu dinheiro em investimento de prazo	Binário	0 e 1

3.3 INGESTÃO E CARREGAMENTO DE DADOS

A ingestão dos dados foi realizada a partir de uma fonte externa, por meio do *download* de um arquivo compactado contendo os dados brutos, utilizando a biblioteca *requests*. O arquivo foi obtido através de uma URL fornecida pelo repositório *UCI Machine Learning Repository* (https://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank.zip). Após o *download*, o conteúdo do arquivo ZIP foi extraído diretamente na memória, e o arquivo bank-full.csv foi carregado em um *DataFrame* da biblioteca Pandas. O conjunto de dados, composto por 45.211 linhas e 17 colunas, foi então armazenado no *DataFrame* denominado *df_raw*, ficando disponível para manipulação e análise no ambiente Jupyter Notebook. Essa abordagem facilitou o prosseguimento para as etapas subsequentes de limpeza, transformação e modelagem dos dados.

3.4 ANÁLISE EXPLORATÓRIA

O processo de análise exploratória foi iniciado com a verificação da dimensão do *dataset*, incluindo o número de registros e colunas. Em seguida, foi analisado o tipo de dado de cada coluna e a existência de valores nulos. Realizada essa primeira análise, observou-se a distribuição de cada variável categórica no *dataset*, obtendo-se os valores absolutos dentro de cada categoria, bem como os valores presentes em cada coluna.

Após essa análise inicial, foram removidas colunas consideradas desnecessárias para o desenvolvimento dos modelos. Dados relacionados às métricas de campanha, bem como a variável alvo original do *dataset*, foram descartados, pois o objetivo do trabalho é aplicar modelos de aprendizado de máquina para identificar quais clientes são propensos a contratar um empréstimo pessoal. Após essa filtragem preliminar, o número de colunas foi reduzido de 17 para 8, conforme apresentado na Tabela 2.

Tabela 2 – Dicionário de dados contendo as variáveis necessárias para o estudo. Fonte: Elaborado pelo Autor.

Nome da variável	Descrição	Tipo de Dado	Valores Possíveis
age	Idade	Numérico	Entre 18 e 95 anos
job	Emprego	Categórico	"admin.","bluecollar","entrepreneur","housemaid","management","retired","self-employed","services","student","technician","unemployed","unknown"
marital	Estado Civil	Categórico	"divorced","married","single","unknown"
education	Nível de Escolaridade	Categórico	"unknown","secondary","primary","tertiary"
default	Indica se o cliente está inadimplente	Binário	0 e 1
balance	Saldo médio da conta em euros	Numérico	Valores entre € -8.019 e € 102.127
housing	Indica se o cliente possuí empréstimo imobiliário	Binário	0 e 1
loan	Indica se o cliente possuí empréstimo pessoal	Binário	0 e 1

Com essa alteração, o novo *dataset* foi transferido para um *DataFrame* denominado *df_trusted*. Após o carregamento dos dados no *df_trusted*, realizou-se uma

série de estatísticas e análises gráficas para entender melhor a distribuição e a relação entre as variáveis, utilizando as bibliotecas Seaborn e Matplotlib.

Essa análise foi dividida em duas etapas: a primeira consistiu na avaliação das categorias e distribuições das variáveis qualitativas, por meio de gráficos de barras, (apresentados no Apêndice B). Para as variáveis quantitativas, foram construídos histogramas com o objetivo de examinar a distribuição de cada variável e verificar possíveis concentrações em faixas específicas. Além disso, *boxplots* foram utilizados para identificar potenciais *outliers*, que poderiam indicar valores discrepantes ou irreais, demandando tratamento antes da aplicação dos modelos preditivos. Os histogramas estão apresentados no Apêndice C e os *boxplots* no Apêndice D.

Nessa etapa, foram prontamente identificados *outliers* nas variáveis numéricas, assim como valores desconhecidos em duas categorias de dados qualitativos. Outro fator relevante observado foi a distribuição da variável alvo no conjunto de dados, que revelou um desbalanceamento significativo: 37.967 (84%) registros pertenciam à classe de clientes que não contrataram crédito, enquanto 7.244 (16%) registros correspondiam à classe de clientes que optaram pela contratação. Esse desbalanceamento pode afetar o desempenho de modelos preditivos, sendo necessário adotar técnicas de balanceamento para garantir resultados mais robustos.

3.5 PRÉ-PROCESSAMENTO

3.5.1 LIMPEZA DE DADOS

Após as avaliações iniciais, tornou-se necessário manipular os registros na tabela *df_trusted*, a fim de organizar os dados de maneira adequada para o treinamento dos modelos. Inicialmente, foram identificadas duas variáveis independentes com valores nulos: a variável *education* e a variável *job*. Tentou-se identificar uma correlação entre essas duas variáveis, de modo que uma pudesse auxiliar na imputação da outra, mas, devido à baixa correlação entre elas, essa abordagem mostrou-se inviável.

Posteriormente, realizou-se o teste do qui-quadrado para avaliar se as variáveis categóricas *job*, *education* e *marital* apresentavam alguma associação significativa com a variável alvo, ou se poderiam ser descartadas. O teste do qui-quadrado compara as frequências observadas em uma tabela de contingência com as frequências esperadas, sob a hipótese de independência entre as variáveis, para determinar a existência de uma relação estatisticamente significativa entre elas. Como o valor-p para essas variáveis foi inferior a 0,05, rejeitou-se a hipótese nula de que elas são independentes da variável alvo, indicando que possuem associação significativa e, portanto, devem ser mantidas no modelo.

Dado que os valores categorizados como "*unknown*" na categoria *job* eram muito pequenos, aproximadamente 0.6%, removeu-se esses registros, passando de 45.211 para 44.923 registros.

3.5.2 TRANSFORMAÇÕES E CODIFICAÇÃO

Após a remoção, realizou-se a codificação binária das categorias *default*, *housing* e *loan*, substituindo os valores "*yes*" e "*no*" por 1 e 0, respectivamente. Nessa mesma etapa, o nome da coluna *loan* foi alterado para y, indicando que esta variável passou a

representar o alvo do problema estudado. Essa transformação é essencial para a preparação dos dados, facilitando a interpretação pelo modelo e alinhando o conjunto de dados aos requisitos das técnicas de modelagem supervisionada.

O processo de *one-hot encoding* foi aplicado às variáveis *job* e marital, removendo-se uma das categorias para evitar a ocorrência de multicolinearidade e, consequentemente, redundância nos dados. Essa transformação foi realizada com o objetivo de adequar os dados para modelos que requerem variáveis numéricas, assegurando que as categorias sejam representadas de maneira compatível com as técnicas de modelagem utilizadas. A remoção de uma categoria, conhecida como *drop-first*, é uma prática essencial para evitar a multicolinearidade, especialmente em modelos lineares. A inclusão de todas as categorias de uma variável categórica como *dummies* cria uma dependência linear entre elas, dificultando a interpretação dos coeficientes e reduzindo a eficiência do modelo. Por exemplo, ao manter as três categorias possíveis de uma variável como *married* em forma de *dummies*, sabemos que, sempre que uma delas ocorre, as outras duas não ocorrem. Nesse caso, é possível prever o valor de uma das variáveis com base nas outras duas, o que gera multicolinearidade. Esse problema pode confundir o modelo, tornando difícil identificar qual variável está, de fato, influenciando o resultado.

No entanto, essa técnica também elevou o número de colunas no *dataset*, aumentando de 8 para 18 variáveis. Embora isso melhore a representação das categorias e facilite a interpretação do modelo, o aumento no número de variáveis também adiciona complexidade, podendo demandar maior capacidade computacional e potencialmente contribuir para o *overfitting* do modelo.

Referente aos valores preenchidos como "*unknown*" na variável *education*, optou-se por criar um *DataFrame* contendo a frequência relativa de cada valor único presente na coluna. Esse procedimento permitiu compreender a distribuição dos dados e facilitou a imputação. Em seguida, os registros com valores "*unknown*" foram preenchidos de forma aleatória, respeitando as proporções observadas nos dados existentes. Essa abordagem foi realizada para preservar as características originais da amostra, garantindo que os valores imputados fossem representativos da distribuição dos dados. Como resultado, o problema de completude na variável *education* foi solucionado, assegurando que não restassem valores ausentes no conjunto de dados.

Após a imputação, realizou-se a codificação da variável *education* por meio do *Label Encoding*, de modo a transformar os níveis de escolaridade em valores numéricos representativos. Os níveis foram ajustados para -1, 0 e 1, correspondendo, respectivamente, ao ensino primário, secundário e terciário, buscando manter a escala das demais variáveis. Essa codificação facilita a interpretação pelo modelo e assegura a coerência com as técnicas de análise utilizadas, uma vez que transforma variáveis categóricas ordinais em valores numéricos que mantém a ordem natural dos níveis de escolaridade.

Para as variáveis quantitativas *age* e *balance*, optou-se por manter seus *outliers*, pois, embora sejam discrepantes em relação aos demais valores, são dados reais e plausíveis de ocorrer. Dessa forma, restou a necessidade de transformar esses valores para uma escala mais próxima da utilizada nas demais variáveis. Foram aplicadas duas

transformações distintas, a primeira por meio da função *StandardScaler*, e a segunda pela função *MinMaxScaler*, ambas pertencentes a biblioteca Scikit-Learn.

O *StandardScaler* transforma os dados para uma distribuição com média 0 e desvio padrão 1, centralizando-os e normalizando com base nos valores médios e na variabilidade dos dados. Esse método é ideal para dados que seguem uma distribuição aproximadamente normal, mas pode ser influenciado por *outliers*, pois estes afetam o desvio padrão. Por outro lado, o *MinMaxScaler* transforma os dados para um intervalo definido, no caso entre 0 e 1, escalando cada valor proporcionalmente entre o mínimo e o máximo observado. Essa técnica preserva a distribuição original dos dados e é menos sensível a *outliers*.

Após testar ambos os métodos, optou-se pelo uso do *MinMaxScaler* (Equação 2), pois apresentou uma leve melhora em relação ao *StandardScaler*, sendo mais adequado pela menor sensibilidade aos *outliers*, o que proporciona uma escala mais uniforme entre as variáveis.

$$x_{novo} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2)$$

De posse de todas as variáveis, foi elaborado um mapa de calor para analisar a correlação entre elas, com o objetivo de identificar quais variáveis independentes possuem maior influência sobre a variável dependente. O mapa de calor (Figura 7) é uma ferramenta visual que facilita a interpretação das correlações, permitindo observar relações positivas e negativas entre as variáveis. Essa análise auxilia na seleção de variáveis que possam contribuir significativamente para a eficácia do modelo, além de identificar possíveis problemas de multicolinearidade, que podem comprometer a robustez do modelo. A correlação varia de -1 a 1: valores próximos de -1 indicam uma forte correlação negativa (quando uma variável aumenta, a outra diminui), enquanto valores próximos de 1 indicam uma forte correlação positiva (ambas aumentam ou diminuem simultaneamente). Já valores próximos de 0 indicam ausência de correlação entre as variáveis.

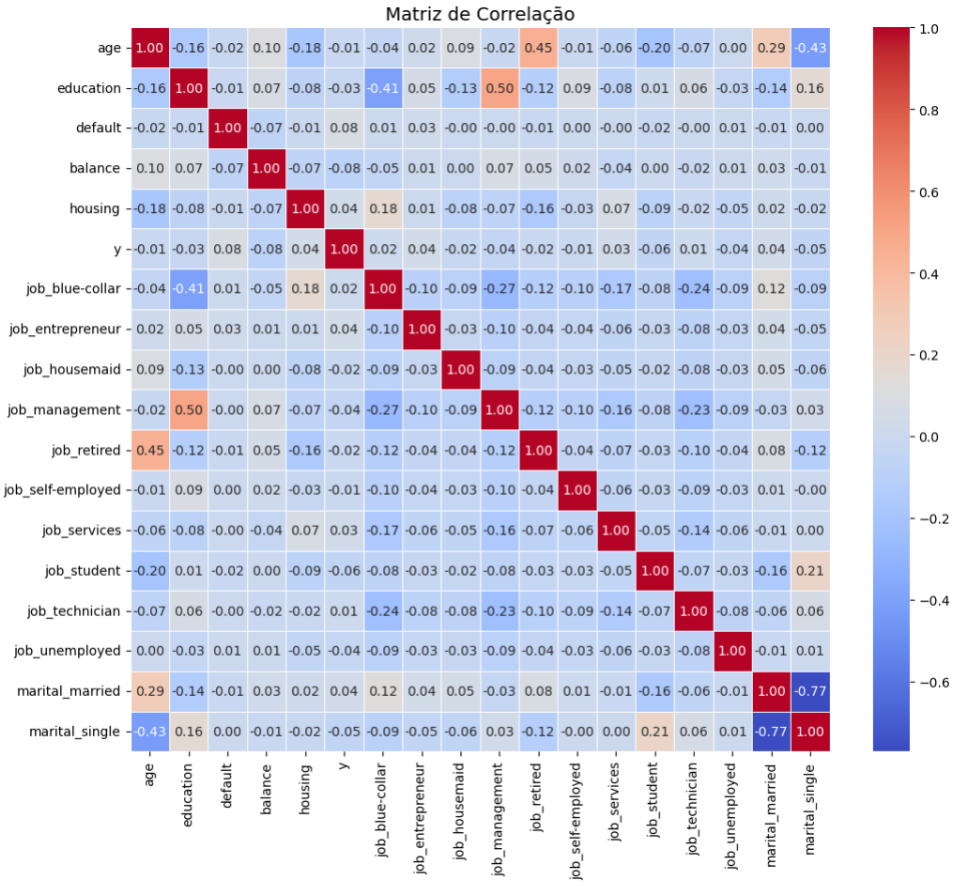

Figura 7 – Mapa de calor contendo a correlação entre as 18 variáveis contidas no *dataset*. Fonte: Elaborado pelo Autor

Ao avaliar o mapa de calor apresentado na Figura 7, foi identificada uma alta correlação entre duas variáveis *dummies* derivadas da variável *marital*. Essa correlação pode causar problemas de multicolinearidade no modelo, impactando negativamente a interpretação dos resultados. Para mitigar esse problema, optou-se por ajustar a categorização da variável *marital*. Originalmente dividida em *single*, *married* e *divorced*, foi consolidada em uma nova categorização binária: 1 para indivíduos que já foram casados (incluindo *married* e *divorced*) e 0 para solteiros (*single*). Além disso, observou-se que não existe variável aparente que tenha uma forte correlação com a variável dependente y (*loan*).

Após esse ajuste, foi elaborado um novo mapa de calor (Figura 8) para verificar se o problema de alta correlação entre as variáveis foi resolvido. Essa abordagem não apenas elimina a redundância, mas reduz a complexidade do modelo.

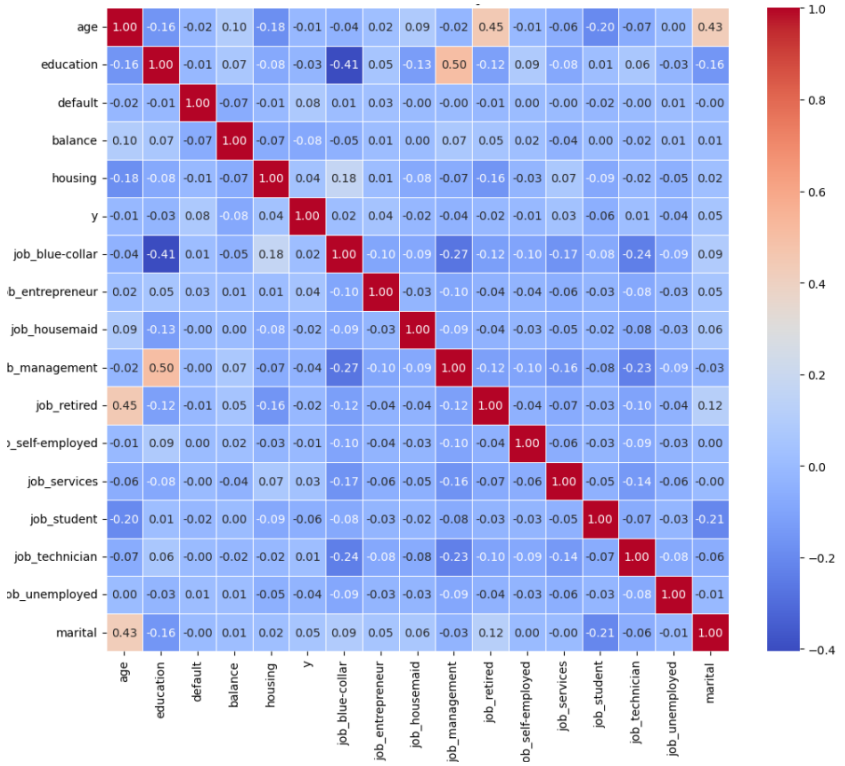

Figura 8 – Mapa de calor com alteração da variável *marital*. Fonte: Elaborado pelo Autor

Para medir o grau de multicolinearidade entre as variáveis numéricas do *dataset*, foi calculado o *Variance Inflation Factor* (VIF), com o auxílio da biblioteca *statsmodels*. O VIF é uma medida que avalia o aumento da variância das estimativas dos parâmetros de um modelo linear quando uma variável adicional é incluída. Ele é utilizado para detectar multicolinearidade em uma matriz de design, identificando variáveis que estão altamente correlacionadas. De acordo com a documentação oficial, valores de VIF acima de 5 indicam alta colinearidade, o que pode resultar em grandes erros padrão nas estimativas dos parâmetros. Observa-se, a partir da Tabela 3, que as variáveis avaliadas apresentaram valores de VIF baixos, indicando a ausência de multicolinearidade significativa entre elas.

Tabela 3 – *Variance Inflation Factor* do conjunto de dados analisados. Fonte: Elaborado pelo Autor.

Variável	VIF
age	1,65
education	1,67
default	1,01
balance	1,03
housing	1,11
y	1,02
job_blue-collar	2,39
job_entrepreneur	1,26
job_housemaid	1,25
job_management	2,58
job_retired	1,67
job_self-employed	1,28
job_services	1,64
job_student	1,23
job_technician	2,07
job_unemployed	1,22
marital	1,29
job_technician	2,07
job_unemployed	1,22
marital	1,29

3.5.3 DIVISÃO EM CONJUNTOS DE TREINAMENTO E TESTE

Após o tratamento dos dados, o conjunto final foi dividido em dois subconjuntos: um contendo as variáveis independentes (x) e outro contendo a variável dependente ou alvo (y). Essa separação foi realizada utilizando a biblioteca Scikit-learn, por meio da função train_test_split, que dividiu os dados em conjuntos de treinamento e teste. Para este trabalho, 30% dos dados foram reservados para o conjunto de teste, enquanto os outros 70% foram utilizados para o treinamento do modelo. A utilização do parâmetro random_state garantiu a reprodutibilidade da divisão dos dados, permitindo que os mesmos subconjuntos fossem gerados em execuções futuras.

Essa abordagem é fundamental para avaliar o desempenho do modelo, pois permite testar sua capacidade de generalização em dados não vistos durante o treinamento. A separação dos dados em treinamento e teste é uma prática essencial para garantir que o modelo não esteja superajustado aos dados de treinamento e para validar sua eficácia de maneira robusta.

3.6 MODELAGEM DE DADOS

A etapa de modelagem de dados consiste em aplicar técnicas de aprendizado de máquina para desenvolver modelos preditivos, utilizando conjuntos de dados previamente processados. Nessa etapa, foram selecionados os algoritmos, definidas as métricas de avaliação e ajustados os hiperparâmetros, com o objetivo de obter um modelo eficiente, capaz de resolver o problema proposto.

3.6.2 AVALIAÇÃO DOS MODELOS

Foram avaliadas cinco técnicas distintas de *machine learning*: Regressão Logística (MONTGOMERY; RUNGER, 2018), árvores de decisão (FACELI et al., 2021), Random Forest (GENUER; POGGI, 2020), XGBoost (CHEN; GUESTRIN, 2016) e CatBoost (PROKHORENKOVA et al., 2021). Cada algoritmo foi testado inicialmente sem qualquer tipo de balanceamento de dados ou ajuste de hiperparâmetros. Posteriormente foram aplicadas as técnicas de *undersampling* (DEVI et al., 2020) e SMOTE (PRADIPTA et al., 2021) para balancear o conjunto de dados e avaliar como essas estratégias impactaram o desempenho de cada modelo.

3.6.3 SELEÇÃO DO MODELO

Após a aplicação dos cinco modelos de *machine learning* nos conjuntos de treino e teste, juntamente com as duas técnicas de balanceamento em cada modelo, obteve-se a Tabela 4, que apresenta os resultados de cada um dos quinze cenários avaliados.

Tabela 4 – Métricas por modelo avaliado. Fonte: Elaborado pelo Autor.

Modelo	Balanceamento	Acurácia	AUC	Precisão Classe 0	Sensibilidade Classe 0	f1-score Classe 1	Precisão Classe 1	Sensibilidade Classe 1	f1-score Classe 1
Regressão Logística	Sem balanceamento	0.55	0.63	0.89	0.53	0.66	0.21	0.65	0.32
Regressão Logística	SMOTE	0.54	0.63	0.89	0.51	0.65	0.21	0.68	0.32
Regressão Logística	Undersampling	0.55	0.62	0.88	0.54	0.67	0.21	0.63	0.31
Árvore de decisão	Sem balanceamento	0.78	0.6	0.87	0.86	0.87	0.32	0.33	0.32
Árvore de decisão	SMOTE	0.67	0.6	0.88	0.71	0.79	0.24	0.47	0.32
Árvore de decisão	Undersampling	0.59	0.6	0.88	0.59	0.71	0.22	0.6	0.32
Random Forest	Sem balanceamento	0.82	0.68	0.86	0.94	0.9	0.39	0.21	0.27
Random Forest	SMOTE	0.7	0.66	0.88	0.75	0.81	0.26	0.46	0.33
Random Forest	Undersampling	0.6	0.66	0.89	0.59	0.71	0.23	0.64	0.34
XGBoost	Sem balanceamento	0.84	0.67	0.84	0.99	0.91	0.48	0.05	0.09
XGBoost	SMOTE	0.65	0.65	0.88	0.68	0.77	0.24	0.51	0.32
XGBoost	Undersampling	0.59	0.66	0.89	0.58	0.71	0.23	0.64	0.34
CatBoost	Sem balanceamento	0.84	0.68	0.84	0.99	0.91	0.52	0.04	0.08
CatBoost	Undersampling	0.59	0.67	0.9	0.57	0.7	0.23	0.67	0.35
CatBoost	SMOTE	0.71	0.65	0.87	0.77	0.82	0.25	0.39	0.3

Os três algoritmos selecionados para a realização do *fine-tuning* foram o CatBoost com *undersampling*, XGBoost com *undersampling* e Random Forest com *undersampling*, por apresentarem um desempenho melhor em relação a métrica F1-Score da Classe 1, demonstrando melhor balanceamento entre precisão e sensibilidade para a Classe 1.

4. RESULTADOS E DISCUSSÕES

4.1 CONFIGURAÇÃO E TREINAMENTO DOS MODELOS

Os dados utilizados nos treinamentos e testes dos modelos foram obtidos no site *UCI Machine Learning Repository*, mais especificamente no conjunto de dados denominado bank_full.csv, fornecido por Moro et al. (2014). O conjunto de dados original contém informações de uma instituição financeira que realizou uma campanha de marketing ativo, utilizando ligações telefônicas para promover um produto de investimento a prazo. O objetivo principal da campanha era identificar, a partir das informações de contato e características pessoais dos clientes, quais deles contratariam o produto financeiro ofertado.

O presente projeto teve como objetivo verificar se um cliente é ou não propenso a tomar crédito, utilizando as variáveis disponíveis no conjunto de dados. O *dataset* inclui informações categóricas e numéricas, como idade, profissão, estado civil, nível educacional e saldo bancário, além de indicadores sobre a existência de empréstimos e financiamentos (Seção 3.1). Essas variáveis foram analisadas com o intuito de prever o comportamento dos clientes e identificar sua propensão à contratação de um empréstimo pessoal.

O arquivo original possui 45.211 registros e 17 colunas, das quais nove foram excluídas por conterem informações relacionadas a campanhas de marketing e contratação de produtos de investimento a prazo, que não eram relevantes para o objetivo deste projeto (Seção 3.4). As variáveis restantes foram classificadas nas seguintes categorias:

- Quantitativas: idade e saldo médio;
- Categóricas: nível educacional, estado civil e profissão;
- Binárias: empréstimo imobiliário, empréstimo pessoal e inadimplência.

A variável *loan* foi definida como a nova variável-alvo do conjunto de dados, uma vez que, identificando os clientes que contratam empréstimos, é possível treinar o modelo para prever novos potenciais clientes propensos a adquirir esse produto.

Durante o pré-processamento (Seção 3.4), foi verificada a existência de valores ausentes e extremos nas variáveis quantitativas (*age* e *balance*). Apesar de identificados *outliers* em ambas as variáveis, verificou-se que os mesmos se tratavam de valores possíveis e reais, condizentes com a natureza dos dados. Sendo assim, o único tratamento realizado foi a aplicação do método *MinMaxScaler*, com o objetivo de padronizar as variáveis e garantir que ficassem em escalas próximas, facilitando o treinamento dos modelos.

Valores ausentes foram identificados nas variáveis categóricas *job* e *education*, sendo que registros com valores ausentes em *job* (0,64% do *dataset*) foram excluídos, enquanto os de *education* foram imputados com base na frequência dos valores existentes. A variável *marital* não apresentou valores ausentes (Seção 3.5.1). Após a limpeza, as variáveis categóricas foram transformadas em numéricas: *education* recebeu *label*

encoding por possuir uma ordem definida, enquanto *marital* e *job* foram convertidas em variáveis *dummies*, aumentando o número de colunas. Em seguida, um mapa de calor foi utilizado para analisar correlações, mas nenhuma variável apresentou impacto significativo sobre a variável-alvo. Por fim, decidiu-se utilizar todas as variáveis no modelo preditivo para preservar informações relevantes e explorar plenamente as relações do *dataset* (Seção 3.5.2).

A escolha por incluir todas as variáveis também foi sustentada pela baixa complexidade computacional, visto que o número de variáveis é pequeno. Essa abordagem permite um modelo robusto sem impacto significativo no tempo de processamento ou risco de *overfitting*, considerando que o número de amostras no *dataset* é adequado para suportar essa quantidade de variáveis.

Em todo o processo descrito, os dados foram organizados em diferentes camadas, garantindo a estruturação e o controle durante o fluxo de processamento. Inicialmente, na camada *raw*, o conjunto de dados armazenado exatamente como foi obtido da fonte, sem qualquer modificação, preservando sua integridade e garantindo rastreabilidade.

Na camada *trusted*, foram realizadas etapas de limpeza, incluindo a exclusão de colunas consideradas irrelevantes para o problema e o tratamento de valores ausentes, como a imputação ou exclusão de registros conforme necessário. Essa camada assegura que os dados estejam confiáveis e consistentes para o início do processo analítico.

Por fim, a camada *refined* incluiu todas as etapas de pré-processamento necessárias para tornar os dados utilizáveis nos modelos de predição. Isso abrangeu técnicas como a padronização de variáveis numéricas (por meio de métodos como Z-score ou *MinMaxScaler*), *label encoding* para variáveis categóricas ordinais e *one-hot encoding* para variáveis categóricas nominais. Essas transformações garantiram que os dados estivessem adequadamente preparados para modelos de aprendizado de máquina, que frequentemente exigiam formatos numéricos e escalas consistentes.

Essa abordagem em camadas não apenas organiza o pipeline de dados de maneira eficiente, mas também facilita o monitoramento e a reprodutibilidade, permitindo revisões ou ajustes em qualquer etapa do processo sem comprometer as demais.

Foram testados cinco métodos distintos no *dataset* tratado: Regressão Logística**,** árvore de decisão**,** Random Forest**,** XGBoost **e** CatBoost**.** Inicialmente, o modelo de Regressão Logística foi utilizado como *baseline**,*** servindo de referência para comparar o desempenho dos modelos mais complexos, como o Random Forest e o XGBoost. Durante o desenvolvimento do trabalho, optou-se por incluir o modelo de árvore de decisão, devido à sua relação direta com o Random Forest, e o CatBoost, por ser uma metodologia especialmente eficiente no tratamento de variáveis categóricas. Todos os modelos foram desenvolvidos com auxílio da biblioteca Sickit-learn

4.2 ANÁLISE COMPARATIVA DOS MODELOS

Devido à grande quantidade de modelos testados (Regressão Logística, Árvore de Decisão, Random Forest, XGBoost e CatBoost) e métodos de balanceamento avaliados (sem balanceamento, *undersampling* e SMOTE), o número de possibilidades seria amplificado significativamente caso fossem considerados os ajustes de hiperparâmetros

para cada modelo. Por isso, nesta etapa inicial, optou-se por utilizar configurações padrão, com o menor número possível de ajustes nos hiperparâmetros. Essa abordagem buscou evitar o consumo excessivo de tempo, o aumento do risco de *overfitting* e o viés na seleção dos modelos, garantindo robustez e comparabilidade inicial entre as diferentes técnicas. Além disso, para assegurar a reprodutibilidade dos ensaios realizados, foi configurada uma semente (*seed*) fixa, padronizando os resultados e minimizando a variabilidade aleatória do processo. A Tabela 4 (Seção 3.6.3) apresentou as métricas de cada modelo avaliado, e a Figura 9 apresenta o resultado gráfico da Curva ROC para cada modelo utilizado no ensaio

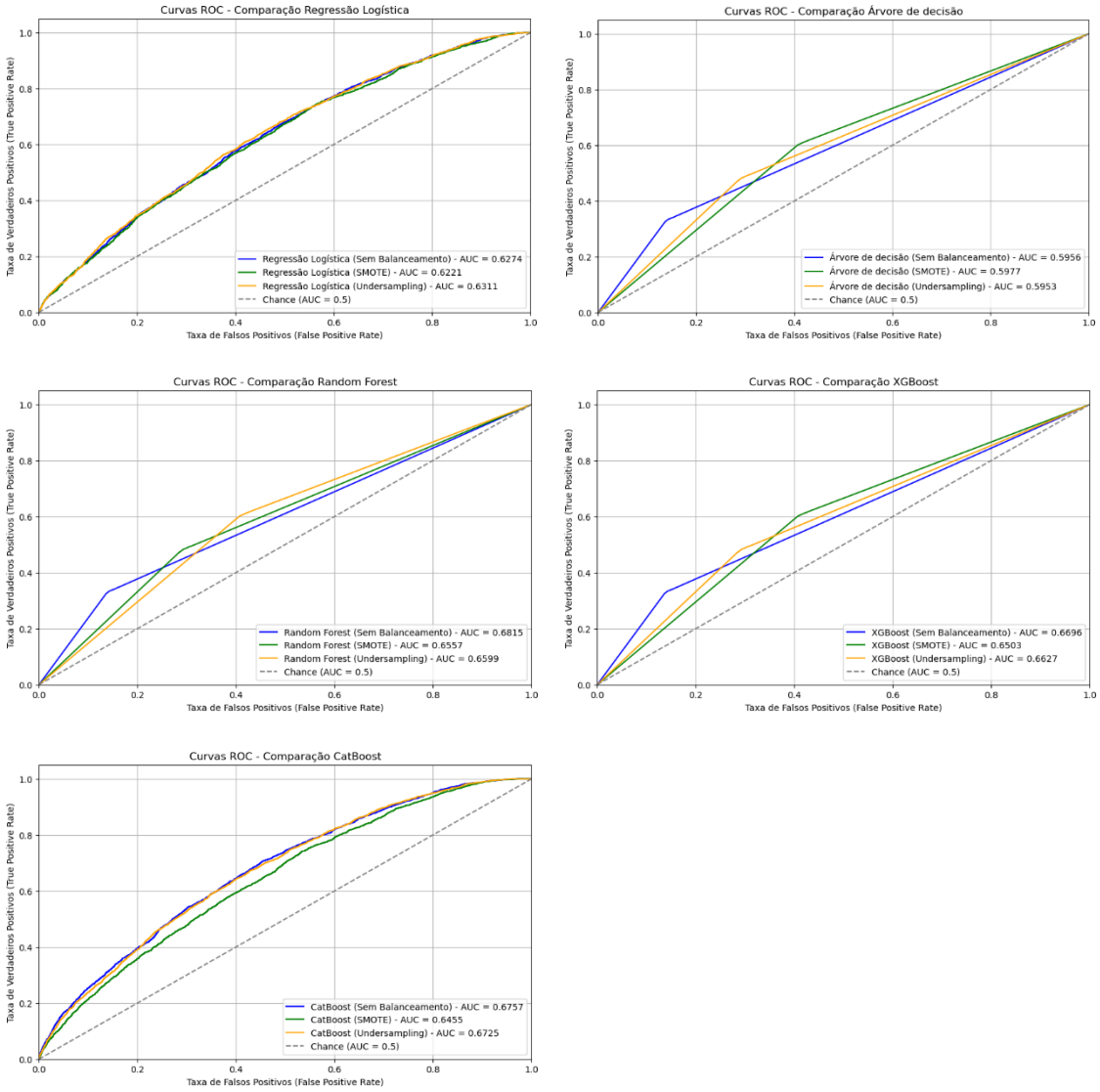

Figura 9 – Curvas ROC para os modelos de Regressão Logística, Árvore de decisão, Random Forest, XGBoost e CatBoost com diferentes métodos de balanceamento. Fonte: Elaborado pelo Autor.

O primeiro critério avaliado foi a Curva ROC. Os modelos de Regressão Logística e Árvore de Decisão apresentaram desempenho abaixo ou próximo a 0.6, indicando que, caso o *threshold* fosse escolhido aleatoriamente, o modelo teria um desempenho apenas ligeiramente melhor que uma escolha ao acaso. Por essa razão, seis das quinze estratégias avaliadas (Regressão Logística e Árvore de Decisão em todos os cenários de

balanceamento) foram eliminadas devido à sua baixa flexibilidade e performance limitada.

A segunda métrica avaliada foi o *recall* da classe 1, considerada a principal métrica neste problema, já que o objetivo é identificar clientes propensos a tomar crédito. Nesse contexto, a capacidade de prever corretamente os casos positivos é essencial para aumentar a eficiência das campanhas e evitar perdas de oportunidade. Dentre os modelos testados, três se destacaram: CatBoost com *undersampling*, XGBoost com *undersampling* e Random Forest com *undersampling*, devido ao equilíbrio apresentado entre sensibilidade da classe 1 (*recall*), precisão e F1-score. Esses modelos obtiveram recall acima de 0.60 para a classe minoritária, além de AUCs entre 0.65 e 0.68, demonstrando bom potencial para futuras etapas de *fine-tuning*.

O fato de os três modelos selecionados utilizarem a mesma estratégia de balanceamento reforça que a redução no volume de registros da classe majoritária proporcionou ao modelo uma melhor oportunidade de identificar os registros da classe minoritária. Além disso, destaca-se o impacto da penalização em modelos desbalanceados: nesses casos, errar a previsão de um cliente propenso como não propenso (falso negativo) gera uma penalização muito baixa em relação ao erro de prever um cliente não propenso como propenso (falso positivo). O excesso de registros da classe majoritária também contribui para enviesar o aprendizado do modelo em direção aos padrões dessa classe, em vez de promover uma melhor generalização para ambas as classes (DAS et al, 2022).

Em relação ao método SMOTE, este pode ser inadequado em alguns casos, pois a geração de múltiplas cópias do mesmo padrão pode tornar o modelo excessivamente específico, levando ao *overfitting*. Além disso, SMOTE aumenta a variância devido à sobreposição causada pela criação de dados sintéticos que não consideram adequadamente a vizinhança entre as classes (DAS et al, 2022). A vizinhança entre as classes refere-se à proximidade de pontos de dados da classe minoritária em relação aos pontos da classe majoritária no espaço de características. Quando essa vizinhança não é levada em conta, SMOTE pode gerar exemplos sintéticos em regiões próximas ou até mesmo dentro da fronteira da classe majoritária, dificultando a separação clara entre as classes e comprometendo o desempenho do modelo.

4.3 AJUSTE DOS MODELOS PRÉ-SELECIONADOS

Após aplicar os 15 algoritmos ao conjunto de dados, selecionaram-se os três modelos com maior F1-Score: CatBoost com *Undersampling*, Random Forest com *Undersampling* e XGBoost com *Undersampling*. Apesar de a principal métrica de avaliação para o problema ser a sensibilidade da classe 1 (*recall*), buscou-se maximizar este indicador sem comprometer completamente a acurácia. O objetivo principal é identificar clientes propensos à contratação de crédito pessoal, considerando que os meios mais comuns de abordagem para essas ofertas incluem e-mails, notificações, *pushs* e outras formas de comunicação digital. Embora falsos positivos não representem um grande problema nesse contexto, em escalas muito grandes, a redução do envio desnecessário de comunicações pode contribuir para a economia de recursos financeiros e a preservação da relação com o cliente, evitando experiências negativas ou incômodas.

Os modelos CatBoost otimizado, Random Forest otimizado e XGBoost otimizado foram ajustados utilizando o método RandomizedSearchCV, da biblioteca Scikit-learn, com o objetivo de maximizar a sensibilidade (*recall*). A escolha pelo RandomizedSearchCV deveu-se à sua eficiência em explorar um grande espaço de hiperparâmetros de forma ágil, possibilitando identificar configurações promissoras em menos tempo em comparação ao GridSearchCV.

Após a identificação dos melhores hiperparâmetros sugeridos para cada modelo, realizou-se o treinamento utilizando essas configurações otimizadas, conforme apresentado na Tabela 5.

Tabela 5 – Valores dos hiperparâmetros selecionados pelo RandomizedSearch. Fonte: Elaborado pelo Autor

Sugestões de valores - Randomized Search					
Hiperparâmetros	CatBoost Undersampling otimizado	Hiperparâmetros	Random Forest Undersampling otimizado	Hiperparâmetros	XGBoost Undersampling otimizado
scale_pos_weight	5	n_estimators	200	subsample	0.6
learning_rate	0.01	max_depth	10	reg_lamda	5
l2_leaf_reg	1	min_samples_split	4	reg_alpha	0
iterations	500	min_samples_lead	5	n_estimators	200
grow_policy	SymmetricTree	class_weight	{0: 1, 1: 5}	max_depth	5
depth	6			learning_rate	0.01
border_count	254			gamma	1
bagging_temperature	0			colsample_bytree	1.0

É importante destacar que, embora esses valores tenham sido testados e resultassem em uma sensibilidade igual a 1, a acurácia apresentou resultados muito baixos. Por esse motivo, apesar das sugestões do algoritmo, os hiperparâmetros relacionados ao peso das classes foram ajustados manualmente para estabelecer uma relação de 1 para 1, buscando um valor razoável entre sensibilidade e acurácia.

Em seguida, aplicou-se a validação cruzada (*cross-validation*) para avaliar o desempenho dos modelos em diferentes subconjuntos de dados, garantindo maior robustez na análise. As métricas avaliadas foram as mesmas para todos os experimentos.

Além das etapas de ajuste fino (*fine-tuning*) e validação, foi aplicado o método SHAP (*Shapley Additive Explanations*) para interpretar os modelos. O SHAP permitiu analisar a contribuição individual de cada variável para a predição final, proporcionando maior entendimento sobre como cada modelo toma decisões. Essa análise não apenas aumentou a interpretabilidade dos modelos, mas também ajudou a identificar variáveis-chave, contribuindo para a melhoria contínua do processo preditivo

Conforme a Tabela 6, os modelos que apresentaram melhor sensibilidade da Classe 1 foram o CatBoost e o Random Forest, ambos com desempenho destacado após o ajuste de threshold. O modelo CatBoost alcançou um *recall* de 0.78, enquanto o Random Forest obteve 0.79. Apesar de o Random Forest apresentar uma ligeira vantagem no *recall*, o CatBoost superou nos indicadores de acurácia (0.48 vs. 0.46) e na média da

sensibilidade na validação cruzada (0.64 vs. 0.63). Esses fatores evidenciam que o CatBoost mantém maior estabilidade e capacidade de generalização nos diferentes subconjuntos de dados.

Tabela 6 – Métricas do modelo CatBoost com *fine-tuning*. Fonte: Elaborado pelo Autor.

Modelo	Balanceamento	Ajuste de threshold	Acurácia	Cuva ROC	Precisão Classe 0	Sensibilidade Classe 0	F1-Score Clasee 0	Precisão Classe 1	Sensibilida de Classe 1	F1-Score Classe 1	Média da sensibilidade validação cruzada	Desvio padrão da sensibilidade
CatBoost	**Undesampling**	**N/A**	0.56	0.66	0.89	0.55	0.68	0.22	0.67	0.33	0.64	0.019
CatBoost	**Undesampling**	**0.47**	0.48		0.91	0.43	0.58	0.21	0.78	0.33		
Random Forest	**Undesampling**	**N/A**	0.59	0.65	0.88	0.58	0.70	0.22	0.61	0.32	0.63	0.027
Random Forest	**Undesampling**	**0.47**	0.46		0.91	0.40	0.56	0.20	0.79	0.32		
XGBoost	**Undesampling**	**N/A**	0.54	0.65	0.89	0.52	0.66	0.21	0.67	0.32	0.67	0.011
XGBoost	**Undesampling**	**0.47**	0.50		0.90	0.46	0.61	0.21	0.72	0.32		

A Figura 10 apresentam as matrizes de confusão dos três modelos que sofreram *fine-tunning*.

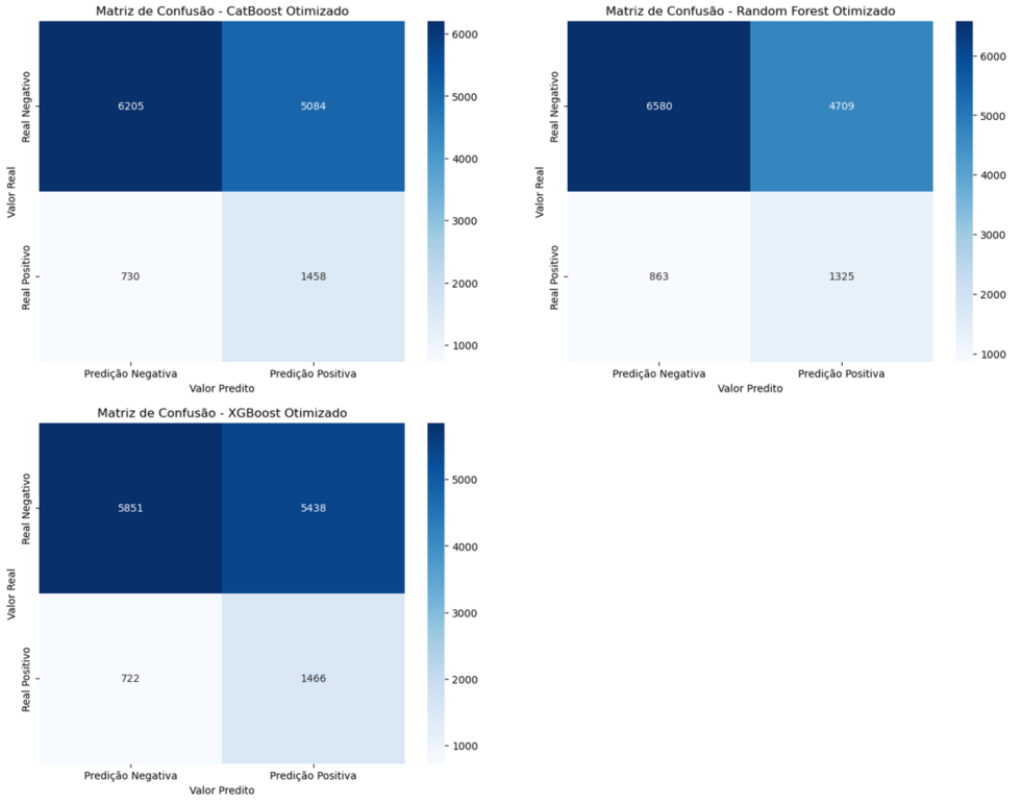

Figura 10 – Matrizes de confusão dos modelos otimizados. Fonte: Elaborado pelo Autor.

A Figura 11 mostra as curvas ROC dos modelos de CatBoost, Random Forest e XGBoost otimizado.

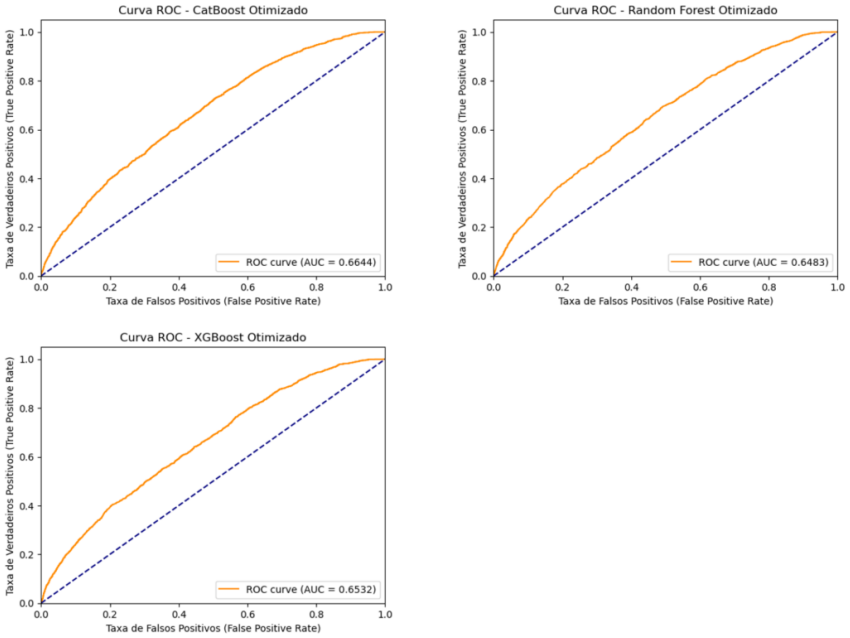

Figura 11 – Curva ROC dos modelos otimizados. Fonte: Elaborado pelo Autor.

Por fim, a figura 12 apresenta o método SHAP aplicado a cada um dos três modelos.

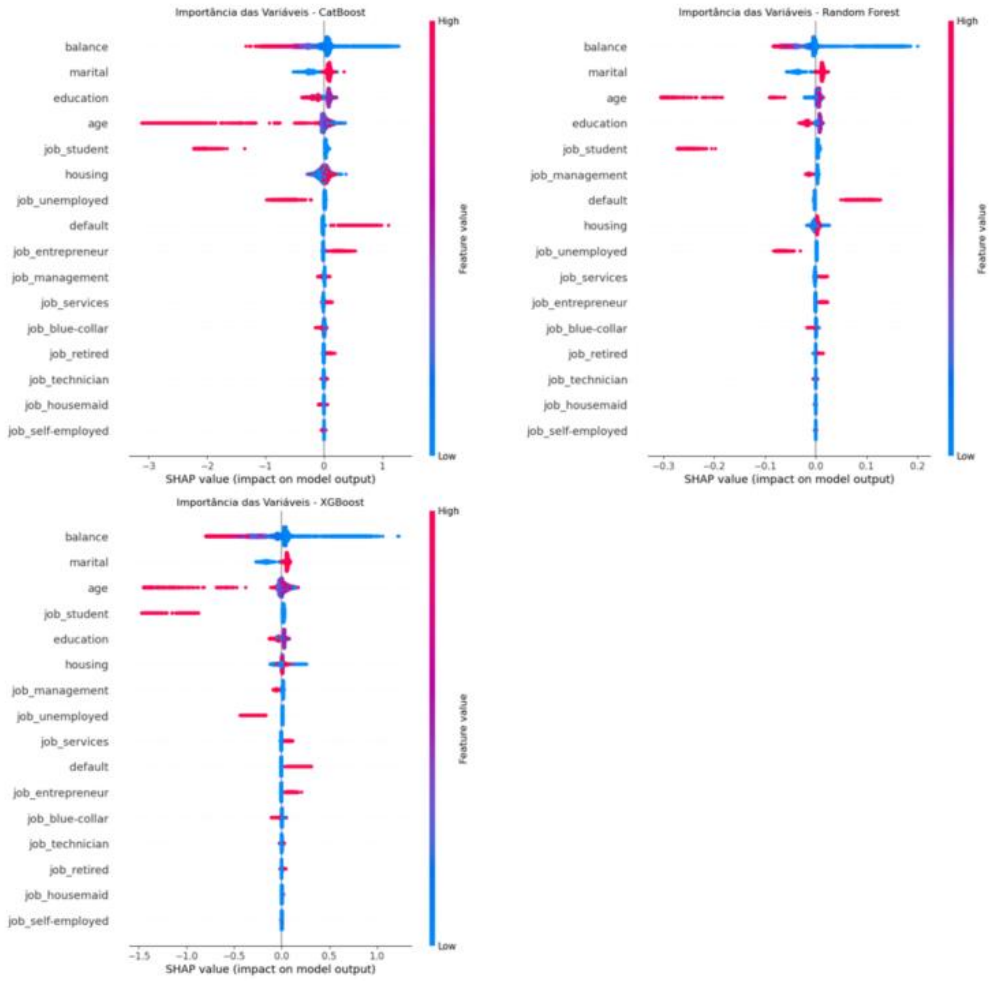

Figura 12 – Importância das variáveis em cada um dos modelos otimizados. Fonte: Elaborado pelo Autor.

A Figura 12 apresenta as variáveis com maior impacto no modelo, com base na explicação fornecida pelo método SHAP (*Shapley Additive Explanations*). O eixo horizontal representa a magnitude da influência de cada variável sobre a predição do modelo, com valores mais à esquerda indicando maior contribuição negativa para a variável alvo e valores mais à direita indicando maior contribuição positiva. No eixo vertical, observam-se os registros ordenados de acordo com o impacto de cada variável. As cores dos pontos indicam os valores dos registros: azul para valores baixos e rosa para valores altos.

Nos três modelos avaliados, observa-se que a variável *balance* apresenta um padrão consistente: valores mais altos (em rosa) têm maior impacto negativo sobre a variável alvo, indicando menor propensão do cliente a tomar crédito. Em contraste, valores baixos (em azul) contribuem positivamente para a probabilidade de o cliente contratar crédito. Essa análise reforça o papel preditivo da variável *balance* na modelagem.

4.4 SELEÇÃO DO MELHOR MODELO

Após a avaliação de todos os modelos, foi realizada uma última etapa de validação, na qual cada um dos modelos otimizados foi aplicado a um novo conjunto de dados. O novo *dataset*, denominado df_novo, passou pelos mesmos processos de tratamento aplicados ao *dataset* originalmente utilizado para o teste dos modelos. Após os testes, foi analisada a performance de cada modelo, tanto sem ajuste de *threshold* quanto com ajuste de *threshold*. A Tabela 7 contempla os resultados obtidos.

Tabela 7 – Métricas dos modelos aplicados a um novo *dataset*. Fonte: Elaborado pelo Autor.

Modelo	Balanceamento	Ajuste de threshold	Acurácia	Precisão Classe 0	Sensibilidade Classe 0	F1-Score Clasee 0	Precisão Classe 1	Sensibilidade Classe 1	F1-Score Classe 1
CatBoost	**Undesampling**	**N/A**	0.56	0.90	0.55	0.68	0.21	0.66	0.32
CatBoost	**Undesampling**	**0.47**	0.48	0.92	0.43	0.58	0.20	0.78	0.32
Random Forest	**Undesampling**	**N/A**	0.58	0.89	0.58	0.70	0.21	0.60	0.31
Random Forest	**Undesampling**	**0.47**	0.40	0.92	0.32	0.47	0.18	0.85	0.30
XGBoost	**Undesampling**	**N/A**	0.54	0.90	0.52	0.65	0.20	0.67	0.31
XGBoost	**Undesampling**	**0.47**	0.43	0.92	0.36	0.52	0.19	0.82	0.31

A escolha do modelo mais adequado para identificar clientes propensos à contratação de empréstimos depende do objetivo principal: maximizar o *recall* (sensibilidade da Classe 1). Esse objetivo é crucial, pois o foco está em identificar o maior número possível de clientes propensos, mesmo que isso resulte em um aumento no número de falsos positivos.

Ao considerar a Tabela 7, percebe-se que o Random Forest apresentou desempenho inferior no quesito F1-Score (Classe 1) (0.30) em comparação ao CatBoost (0.32), o que reforça a escolha do CatBoost como o modelo final. Além disso, o XGBoost, embora tenha demonstrado resultados consistentes, apresentou menor *recall* ajustado (0.72) em relação aos outros dois modelos conforme a Tabela 6, sendo uma opção menos favorável para o problema em questão.

O modelo resolve o problema de identificar clientes propensos à contratação de empréstimos, priorizando a redução de falsos negativos (ou seja, minimizar a quantidade de clientes propensos que não foram identificados). A métrica determinante foi o *recall* da Classe 1, já que o objetivo é garantir que o maior número possível de clientes propensos seja identificado. Adicionalmente, a acurácia e a validação cruzada foram consideradas para avaliar a estabilidade do modelo.

4.5 DISCUSSÃO SOBRE OS RESULTADOS

Os modelos CatBoost, Random Forest e XGBoost otimizados foram avaliados com base em múltiplas métricas para determinar qual deles seria mais eficaz na identificação de clientes propensos à contratação de empréstimos. O objetivo principal foi maximizar o *recall* da Classe 1 (sensibilidade), garantindo que o maior número possível de clientes propensos fosse corretamente identificado. Essa abordagem é crucial, pois, no contexto de campanhas de marketing, é preferível priorizar potenciais clientes mesmo ao custo de alguns falsos positivos, já que a perda de clientes propensos (falsos negativos) pode representar uma oportunidade perdida.

Conforme a Tabela 6 os modelos apresentaram comportamentos distintos:

- O CatBoost destacou-se pela maior média de sensibilidade na validação cruzada (0,64), mantendo consistência nos diferentes subconjuntos de dados.
- O Random Forest apresentou ligeiramente maior *recall* ajustado na Classe 1 (0,79 contra 0,78 do CatBoost), mas com pior desempenho em métricas complementares, como acurácia (0,46 vs. 0,48) e F1-Score (0,32 vs. 0,33).
- O XGBoost apresentou resultados consistentes, mas ficou aquém nos indicadores de *recall* ajustado (0,72) e sensibilidade da Classe 1, o que o torna uma escolha menos favorável para o objetivo proposto.

Portanto, o CatBoost otimizado foi selecionado como o modelo mais adequado, por equilibrar *recall*, acurácia e estabilidade, demonstrando capacidade de generalização e menor variabilidade.

O aumento do recall (sensibilidade da Classe 1) muitas vezes resulta em uma redução na precisão, especialmente em problemas de classes desbalanceadas. Esse *trade-off* é evidente nos modelos avaliados em um novo conjunto de dados, conforme Tabela 7, que inclui os impactos do ajuste de *threshold*:

- Para o CatBoost, o ajuste de *threshold* aumentou o *recall* para 0,78, mas reduziu a precisão da Classe 1 para 0,20. Isso significa que, ao identificar mais clientes propensos, o modelo gerou mais falsos positivos.

- O Random Forest, embora tenha alcançado o maior *recall* ajustado, apresentou a menor precisão (0,18 na Classe 1), o que indica maior número de falsos positivos, reduzindo sua eficácia prática em campanhas de marketing.

- O XGBoost, com *recall* ajustado de 0,82 e precisão de 0,19, conseguiu um equilíbrio intermediário, mas não superou os outros dois modelos.

Esses resultados ilustram a necessidade de priorizar métricas que reflitam diretamente os objetivos do problema. No caso analisado, o *recall* foi escolhido como métrica central, pois é mais importante identificar clientes propensos do que evitar falsos positivos.

Os resultados mostram que o modelo selecionado, o CatBoost otimizado, é o mais adequado para resolver o problema de identificação de clientes propensos à contratação de empréstimos. Ele maximiza o *recall* com perda controlada de precisão, sendo ideal para campanhas de marketing direcionadas. A escolha reflete um compromisso entre identificar o maior número de clientes potenciais e minimizar impactos negativos decorrentes de falsos positivos. Além disso, a estabilidade do CatBoost ao longo de diferentes rodadas de validação reforça sua confiabilidade como solução preditiva.

5. CONCLUSÕES

5.1 RESUMO DOS PRINCIPAIS INSIGHTS

O trabalho teve como foco principal a comparação de cinco técnicas de aprendizado de máquina para definição de um modelo de propensão, além da análise de seus resultados. Foi desenvolvida uma metodologia que envolveu a aplicação inicial dos modelos preditivos sem ajustes nos hiperparâmetros, buscando garantir uma comparação justa e objetiva entre as abordagens testadas. Essa análise inicial permitiu identificar o desempenho básico dos modelos utilizando três técnicas distintas de balanceamento de classes (SMOTE, *undersampling* e sem balanceamento), visando identificar o método mais adequado para o problema. As etapas completas dessa metodologia estão detalhadas no Apêndice E, que descreve as implementações realizadas.

Com base nos resultados dessa primeira etapa, foi realizada uma segunda análise com o ajuste fino dos hiperparâmetros, utilizando validação em um novo conjunto de dados para confirmar a robustez dos modelos. Essa abordagem permitiu explorar o potencial máximo de cada técnica, considerando as características específicas do problema e as métricas de avaliação.

O estudo demonstrou que a métrica de *recall*, que mede a proporção de verdadeiros positivos sobre a soma de verdadeiros positivos e falsos negativos, é a mais eficaz para este tipo de problema de classificação. Essa escolha é justificada pela menor relevância dos falsos positivos neste contexto, dado que eles não geram impactos significativos nos negócios. A ênfase no *recall* permitiu avaliar com maior precisão a capacidade dos modelos em identificar clientes propensos, atendendo ao objetivo principal da análise.

Por fim, o trabalho destacou a eficiência do modelo CatBoost, que apresentou o melhor desempenho geral após o *fine-tuning*, especialmente em cenários com dados desbalanceados e uma grande quantidade de variáveis categóricas. Sua superioridade foi confirmada tanto na análise inicial quanto após os ajustes, evidenciando seu potencial como solução robusta e eficiente para problemas similares. Os resultados obtidos reforçam a aplicabilidade prática da abordagem proposta, com impactos diretos para o setor bancário, onde a otimização de campanhas de marketing baseadas em dados reais pode reduzir custos, aumentar a eficiência e maximizar o retorno sobre o investimento.

5.2 LIMITAÇÕES DO ESTUDO

Apesar do uso de técnicas de balanceamento de dados, como o SMOTE e o *undersampling***,** o estudo demonstrou que o desempenho dos modelos ainda é afetado pelo *trade-off* entre *recall* e precisão. Embora o *recall* tenha sido priorizado para maximizar a identificação de clientes propensos à contratação de empréstimos, a redução na precisão implica um aumento no número de falsos positivos. Essa característica pode ser aceitável no contexto de comunicações massivas, mas ainda é uma limitação na análise de métricas complementares, como o F1-Score.

Outra limitação foi a seleção de apenas três modelos (CatBoost, Random Forest e XGBoost) para o ajuste de hiperparâmetros e validação final. Apesar desses modelos apresentarem bons resultados e serem amplamente utilizados no mercado, a exclusão de

alternativas, como LightGBM (KE et al., 2017), pode ter restringido a identificação de soluções potencialmente mais eficazes.

O estudo utilizou dados históricos para treinar e avaliar os modelos, o que pode limitar sua capacidade de adaptação a mudanças no comportamento dos clientes ao longo do tempo. Essa limitação destaca a necessidade de monitoramento contínuo e reavaliação dos modelos para manter sua eficácia em cenários futuros.

Por fim, embora o pipeline utilizado seja funcional, ele carece de maior automação em etapas como ingestão de dados, pré-processamento e aplicação de modelos. Essa limitação reduz a escalabilidade e a eficiência da solução, especialmente em contextos empresariais que demandam maior agilidade e integração.

5.3 SUGESTÕES PARA TRABALHOS FUTUROS

Investigar o impacto de técnicas como ADASYN ou a combinação de *oversampling* e *undersampling* pode contribuir significativamente para a melhoria dos resultados em problemas com classes desbalanceadas. Além disso, a exploração de outros modelos de aprendizado de máquina, como LightGBM (KE et al. 2017), ou métodos híbridos que combinem os melhores resultados obtidos, também se apresenta como uma abordagem promissora para aumentar a precisão e a sensibilidade das predições.

Outro aspecto relevante a ser investigado é o impacto dos falsos positivos nos custos e na eficiência das campanhas de marketing. É fundamental avaliar a viabilidade de incorporar métricas de custo ao processo de otimização, o que pode proporcionar uma análise mais realista e prática no contexto empresarial.

Adicionalmente, o aprimoramento do pipeline automatizado para a ingestão de dados, pré-processamento, treinamento e avaliação dos modelos é essencial para facilitar a aplicação prática em ambientes empresariais. Esse processo automatizado pode reduzir erros, otimizar o tempo e garantir maior consistência nos resultados.

Por fim, este estudo fornece uma base sólida para a aplicação de modelos preditivos em campanhas de marketing direcionadas, destacando a importância da escolha de métricas adequadas e da análise criteriosa de *trade-offs* para atingir os objetivos do problema. Trabalhos futuros podem expandir e aprofundar essas abordagens, aprimorando ainda mais a eficácia e a robustez das soluções propostas.

6. REFRÊNCIAS

ASHMORE, R.; CALINESCU R.; PATERSON C. **Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges**. ACM Computing Surveys, v. 54, 2021. https://doi.org/10.1145/3453444

AZEVÊDO, M.; GARTNER, I. **Concentração e Competição no Mercado de Crédito Doméstico**. Associação Nacional de Pós-Graduação e Pesquisa em Administração (ANPAD). v.24, 2020. https://doi.org/10.1590/1982-7849rac2020190347

BAYRAKTAR, R.; ERDEM M. **Motion Aware Data Sampling Using Sequential Frames for Deep Learning Models**. 2023

https://ieeexplore.ieee.org/document/10297027

BREIMAN, L. **Random Forests**. Machine Learning. v.45, p. 5–32, 2001. https://doi.org/10.1023/A:1010933404324

BROWNIEE, J.; **Data Preparation for Machine Learning: Data Cleaning, Feature Selection and Data Transforms in Python**. 2020.

CADY, Field. **The Data Science Handbook**. 2. ed. Hoboken: Wiley, 2017.

CHEN, T; GUESTRIN, C. **XGBoost: A Scalable Tree Boosting System**. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Minning, San Francisco, CA, USA, p. 785-794, 2016. https://doi.org/10.1145/2939672.2939785

COŞER, A.; ALDEA, A.; MAER-MATEI, M.; BESIR, L. **Propensity to churn in banking: What makes customers close the relationship with a bank?** Economic Computation and Economic Cybernetics Studies and Research. pp. 77-94. v.54, 2020. https://doi.org/10.24818/18423264/54.2.20.05

DAS, S.; MULLICK, S. S.; ZELINKA, I. **On Supervised Class-Imbalanced Learning: An Updated Perspective and Some Key Challenges**. IEEE Transactions on Artificial Intelligence, v. 3, p. 973-993, 2022. https://doi.org/10.1109/TAI.2022.3160658

DEVI, D.; BISWAS, S.; PURKAYASHTHA, B; **A Review on Solution to Class Imbalance Problem: Undersampling Approaches**. International Conference on Computational Performance Evaluation (ComPE), 2020. https://doi.org/10.1109/compe49325.2020.9200087

DIAMOND, D. **Financial Intermediation and Delegated Monitoring**. The Review of Economic Studies, 1984.

FACELI, K.; LORENA, A. C.; GAMA, J.; ALMEIDA, T; CARVALHO, A; Inteligência Artificial: uma abordagem de aprendizado de máquina, 2. ed. Rio de Janeiro: LTC, 2021. p. 148-149.

GENUER, R.; POGGI, J. **Random Forests with R**. 1. ed, Springer, 2020. p. 33-40.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T. **LightGBM: A Highly Efficient Gradient Boosting Decision Tree**. In Advances in Neural Information Processing Systems 31 (NeurIPS 2017).

KOHNEHSHAHRI, F; MERLO, A; MAZZOLI, D; BÒ, M; STAGNI, R. **Machine learning applied to gait analysis data in cerebral palsy and stroke: A systematic review**. Gait & Posture, v. 111, 2024. p. 105-121. https://doi.org/10.1016/j.gaitpost.2024.04.007

KÜHL, N.; GOUTIER, M.; HIRT, R.; SATZGER, G. **Machine Learning in Artificial Intelligence: Towards a Common Understanding**. Proceedings of the 52nd Hawaii International Conference on System Sciences, 5236-5245, 2019.

LEMOS, P.; STEINER, M; NIEVOLA, J. **Análise de crédito bancário por meio de redes neurais e árvores de decisão: uma aplicação simples de data mining**. Revista de Administração RAUSP, v. 40, pp. 225-234, 2005. Disponível em: https://www.redalyc.org/articulo.oa?id=223417392002

MAIA, W; DAVID, S; **Machine learning applied to estate pricing for residential rentals in dynamic urban markets—The case of São Paulo city**. Engineering Analysis with Boundary Elements. v. 169, 2024. https://doi.org/10.1016/j.enganabound.2024.105988

MAHESH, B. **Machine Learning Algorithms - A Review**. International Journal of Science and Research (IJSR), v. 9, n. 1, p. 381-386, 2020. https://doi.org/10.21275/ART20203995

MONTGOMERY, D.; PECK, E.; VINNING, G. **Introdutction to Linear Regression Analysis**. 6. ed. Hoboken, NJ: Wiley, 2021.

MONTGOMERY, D; RUNGER, G. **Applied Statistics and Probability for Engineers**. 7. ed. Hoboken: Wiley, 2018. p. 305-308.

MORO S.; RITA P.; CORTEZ P. **Bank Marketing**. Doado em: 13 de fevereiro de 2012.

Disponível em: https://archive.ics.uci.edu/dataset/222/bank+marketing

MUSCHELLI, J. **ROC and AUC with a Binary Predictor: a Potentially Misleading Metric**. Journal of Classification, v. 37, 2020. p.696-708. https://doi.org/10.1007/s00357-019-09345-1

PRADIPTA, G; WARDOYO, R; MUSDHOLIFAH, A; SANJAYA, I; ISMAIL, M. **SMOTE for Handling Imbalanced Data Problem: A Review**. 2021 Sixth International Conference on Informatics and Computing (ICIC), 2021. https://doi.org/10.1109/icic54025.2021.9632912

PROKHORENKOVA, L; GUSEV, G.; VOROBEV, A; DOROGUSH, A; GULIN, A. **CatBoost: unbiased boosting with categorical features**. In Advances in Neural Information Processing Systems 32 (NeurIPS 2018), Montréal, Canada.

RAMACHANDRAN, S.; JAYALAL, M; VASUDEAN M.; DAS, S.; JEHADEESAN, R. **Combining Machine Learning techniques and Genetic Algorithm for predicting run times of High Performance Computing jobs**, Applied Soft Computing Journal, v 165, 2024.

https://doi.org/10.1016/j.asoc.2024.112053

REIS, J.; HOUSLEY, M. **Fundamentals of Data Engineering: Plan and Build Robust Data Systems**. 1. ed. Sebastopol: O'Reilly, 2022.

RINCY, T.;GUPTA, R. **A Survey on Machine Learning Approaches and Its Techniques**, 2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS). https://doi.org/10.1109/SCEECS48394.2020.190

SALAM, M.; RUKKA, R; SAMMA, M.; TENRIAWARU A.; RAHMADANIH; MUSLIM A; ALI, H, RIDWAN, M. **The causal-effect model of input factor allocation on maize roduction: Using binary logistic regression in search for ways to be more productive**. Journal of Agriculture and Food Research 16. 2024 https://doi.org/10.1016/j.jafr.2024.101094

SERAJ, A.; MOHAMMADI-KHANAPOSHTANI M.; DANESHFAR R.; NASERI M.; ESMAEILI M.; BAGHBAN A.; HABIBZADEH S.; ESLAMIAN S. Handbook of HydroInformatics Volume I: Classic Soft-Computing Techniques. Capitulo 5, p. 89-105. v. 1, 2023.

https://doi.org/10.1016/B978-0-12-821285-1.00021-X

SICKIT-LEARN. **Documentação Oficial**

Disponível em: https://scikit-learn.org/dev/api/sklearn.preprocessing.html

STATS MODELS. **Documentação Oficial**

Disponível em: https://www.statsmodels.org/stable/generated/statsmodels.stats.outliers_influence.variance_inflation_factor.html#statsmodels.stats.outliers_influence.variance_inflation_factor

SURESH, H.; GUTTAG J. **A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle**. Equity and Access in Algorithms Mechanisms and Optimization (EAAMO '21) (pp. 1–9). ACM. 2021.

https://doi.org/10.1145/3465416.3483305

TABAK, B.; FAZIO D.; CAJUEIRO, D. **The effects of loan portfolio concentration on Brazilian banks return and risk**. Journal of Banking & Finance. pp. 3065-3076. v.35, 2011. https://doi.org/10.1016/j.jbankfin.2011.04.006

UC IRVINE MACHINE LEARNING REPOSITORY. **Bank Marketing Data Set**.

Disponível em: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing.

Apêndice A – Hiperparâmetros dos modelos utilizados

Tabela A1 – Hiperparâmetros utilizados nos modelos de Regressão Logística. Fonte: Elaborado pelo Autor.

Hiperparâmetro	Descrição	Valor Utilizado
class_weight	Ajusta o peso das classes com base nas frequências observadas no conjunto de dados.	balanced' (sem balanceamento), None (com SMOTE e undersampling) (explícito)
random_state	Permite resultados reprodutíveis ao fixar a aleatoriedade do experimento.	seed (explícito)
penalty	Define o tipo de regularização, sendo neste caso a regularização de norma quadrada (l2).	'l2' (implícito)
solver	Algoritmo usado para otimizar a função de custo.	'lbfgs' (implícito)
tol	Representa o nível de precisão necessário para declarar que o modelo convergiu.	1e-4 (implícito)
fit_intercept	Indica se um intercepto será incluído no modelo, definindo o ponto de interseção com o eixo y.	True (implícito)
C	Representa a inversão da força de regularização. Valores menores implicam regularização maior.	1.0 (implícito)

Tabela A2 - Hiperparâmetros utilizados nos modelos de Árvore de decisão. Fonte: Elaborado pelo Autor.

Parâmetro	Descrição	Valor Utilizado
random_state	Permite resultados reprodutíveis ao fixar a aleatoriedade do experimento.	seed (explícito)
criterion	Função usada para medir a qualidade da divisão.	'gini' (implícito)
splitter	Estratégia usada para dividir os nós da árvore.	'best' (implícito)
max_depth	Profundidade máxima da árvore. Se None, os nós são expandidos até que todas as folhas sejam puras ou contenham menos de min_samples_split amostras.	None (implícito)
min_samples_split	Número mínimo de amostras necessárias para dividir um nó.	2 (implícito)
min_samples_leaf	Número mínimo de amostras necessário para formar uma folha.	1 (implícito)
min_weight_fraction_leaf	Fração mínima do peso total (amostras ponderadas) requerida em uma folha.	0.0 (implícito)
max_features	Número máximo de recursos considerados para encontrar a melhor divisão.	None (implícito)
random_state	Semente para garantir reprodutibilidade.	seed (explícito)
max_leaf_nodes	Número máximo de folhas permitidas na árvore. Se None, o número de folhas não será limitado.	None (implícito)
min_impurity_decrease	Valor mínimo de redução de impureza necessária para dividir um nó.	0.0 (implícito)
ccp_alpha	Parâmetro de custo-complexidade usado para poda mínima.	0.0 (implícito)

Tabela A3 - Hiperparâmetros utilizados nos modelos de *Random Forest*. Fonte: Elaborado pelo Autor.

Parâmetro	Descrição	Valor Utilizado
random_state	Permite resultados reprodutíveis ao fixar a aleatoriedade do experimento.	seed (explícito)
n_estimators	Número de árvores na floresta.	100 (implícito)
criterion	Função usada para medir a qualidade da divisão.	'gini' (implícito)
max_depth	Profundidade máxima das árvores. Se None, os nós são expandidos até todas as folhas serem puras ou conterem menos de min_samples_split amostras.	None (implícito)
min_samples_split	Número mínimo de amostras necessárias para dividir um nó.	2 (implícito)
min_samples_leaf	Número mínimo de amostras necessário para formar uma folha.	1 (implícito)
min_weight_fraction_leaf	Fração mínima do peso total (amostras ponderadas) requerida em uma folha.	0.0 (implícito)
max_features	Número máximo de recursos considerados para encontrar a melhor divisão.	'sqrt' (implícito)
max_leaf_nodes	Número máximo de folhas permitidas nas árvores. Se None, o número de folhas não será limitado.	None (implícito)
bootstrap	Se os dados de treino serão amostrados com reposição.	True (implícito)
oob_score	Se deve usar amostras fora da bolsa (out-of-bag) para estimar a precisão.	False (implícito)
n_jobs	Número de jobs (processos paralelos) usados para treinamento.	None (implícito)
verbose	Controla o nível de mensagens de saída durante o ajuste do modelo.	0 (implícito)
warm_start	Se deve reutilizar soluções anteriores para adicionar mais árvores ao modelo atual.	False (implícito)
ccp_alpha	Parâmetro de custo-complexidade usado para poda mínima.	0.0 (implícito)
max_samples	Número máximo de amostras da base para treinar cada árvore, se bootstrap=True.	None (implícito)

Tabela A4 - hiperparâmetros utilizados nos modelos de XGBoost. Fonte: Elaborado pelo Autor.

Parâmetro	Descrição	Valor Utilizado
random_state	Permite resultados reprodutíveis ao fixar a aleatoriedade do experimento.	seed (explícito)
n_estimators	Número de árvores no modelo de boosting.	100 (implícito)
max_depth	Profundidade máxima de cada árvore.	6 (implícito)
learning_rate	Taxa de aprendizado. Controla o peso atribuído a cada árvore no ensemble.	0.3 (implícito)
verbosity	Controla o nível de mensagens de saída durante o treinamento.	1 (implícito)
objective	Função de perda otimizada durante o treinamento.	'binary:logistic' (implícito)
booster	Tipo de booster usado (e.g., 'gbtree', 'gblinear', 'dart').	'gbtree' (implícito)
tree_method	Algoritmo usado para construir as árvores.	'auto' (implícito)
gamma	Mínima redução na função de perda exigida para realizar uma divisão.	0 (implícito)
min_child_weight	Soma mínima do peso das instâncias (amostras ponderadas) em um nó folha.	1 (implícito)
subsample	Fração de amostras utilizadas para treinar cada árvore.	1.0 (implícito)
colsample_bytree	Fração de features usadas por árvore.	1.0 (implícito)
colsample_bylevel	Fração de features usadas em cada nível da árvore.	1.0 (implícito)
colsample_bynode	Fração de features usadas em cada nó da árvore.	1.0 (implícito)
reg_alpha	Força de regularização L1 aplicada aos pesos das folhas.	0 (implícito)
reg_lambda	Força de regularização L2 aplicada aos pesos das folhas.	1 (implícito)
scale_pos_weight	Peso de compensação para lidar com classes desbalanceadas.	1 (implícito)
base_score	Valor inicial da predição antes do treinamento.	0.5 (implícito)
missing	Valor a ser considerado como ausente no dataset.	np.nan (implícito)

Tabela A5 - Hiperparâmetros utilizados nos modelos de CatBoost. Fonte: Elaborado pelo Autor.

Parâmetro	Descrição	Valor Utilizado
random_state	Permite resultados reprodutíveis ao fixar a aleatoriedade do experimento.	seed (explícito)
iterations	Número máximo de iterações (árvores) no modelo.	1000 (implícito)
learning_rate	Taxa de aprendizado usada para ajustar o peso das árvores.	0.03 (implícito)
depth	Profundidade máxima das árvores.	6 (implícito)
loss_function	Função de perda otimizada durante o treinamento.	'Logloss' (implícito)
eval_metric	Métrica usada para avaliação do modelo durante o treinamento.	'Logloss' (implícito)
bootstrap_type	Tipo de método de bootstrap para amostrar os dados durante o treinamento.	'Bayesian' (implícito)
subsample	Fração de amostras utilizadas em cada iteração.	0.66 (implícito, depende do bootstrap_type)
l2_leaf_reg	Coeficiente de regularização L2.	3.0 (implícito)
border_count	Número máximo de divisões (bins) permitidas para variáveis numéricas.	254 (implícito)
thread_count	Número de threads usadas para paralelismo.	(utiliza todas disponíveis, implícito)
verbose	Controla o nível de saída de mensagens durante o treinamento.	True (implícito)
early_stopping_rounds	Número de iterações sem melhora na métrica de validação antes de interromper o treinamento.	None (implícito)
cat_features	Índices ou nomes das colunas categóricas no dataset.	None (implícito)
class_weights	Pesos atribuídos às classes para lidar com desbalanceamento.	None (implícito)
max_ctr_complexity	Complexidade máxima permitida para interações categóricas.	4 (implícito)

Apêndice B – Análise exploratória das variáveis qualitativas do *dataset*

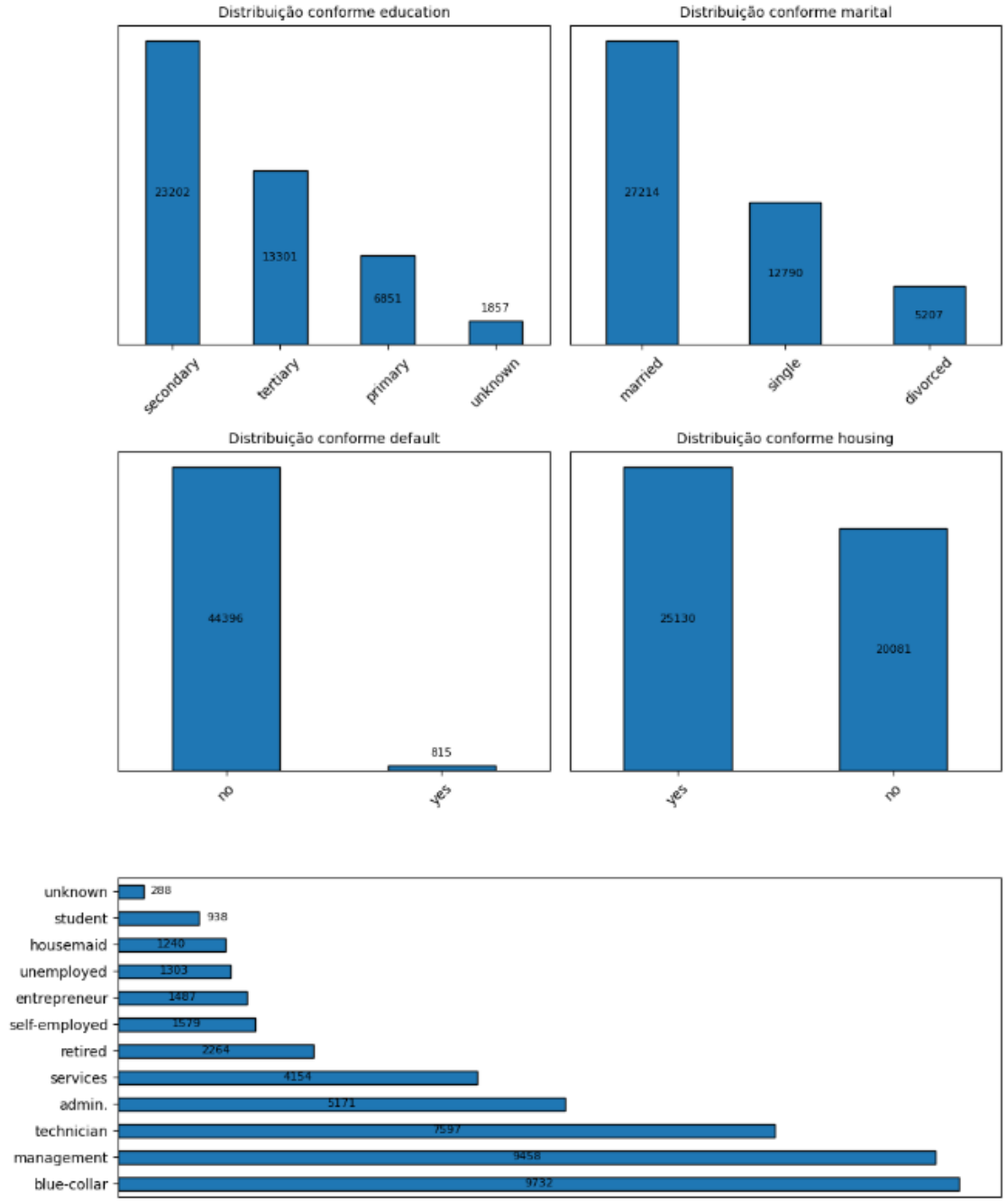

Figura B1 – Processo exploratório das variáveis categóricas.

Fonte: Elaborado pelo Autor.

Apêndice C - Análise exploratória das variáveis quantitativas do *dataset*

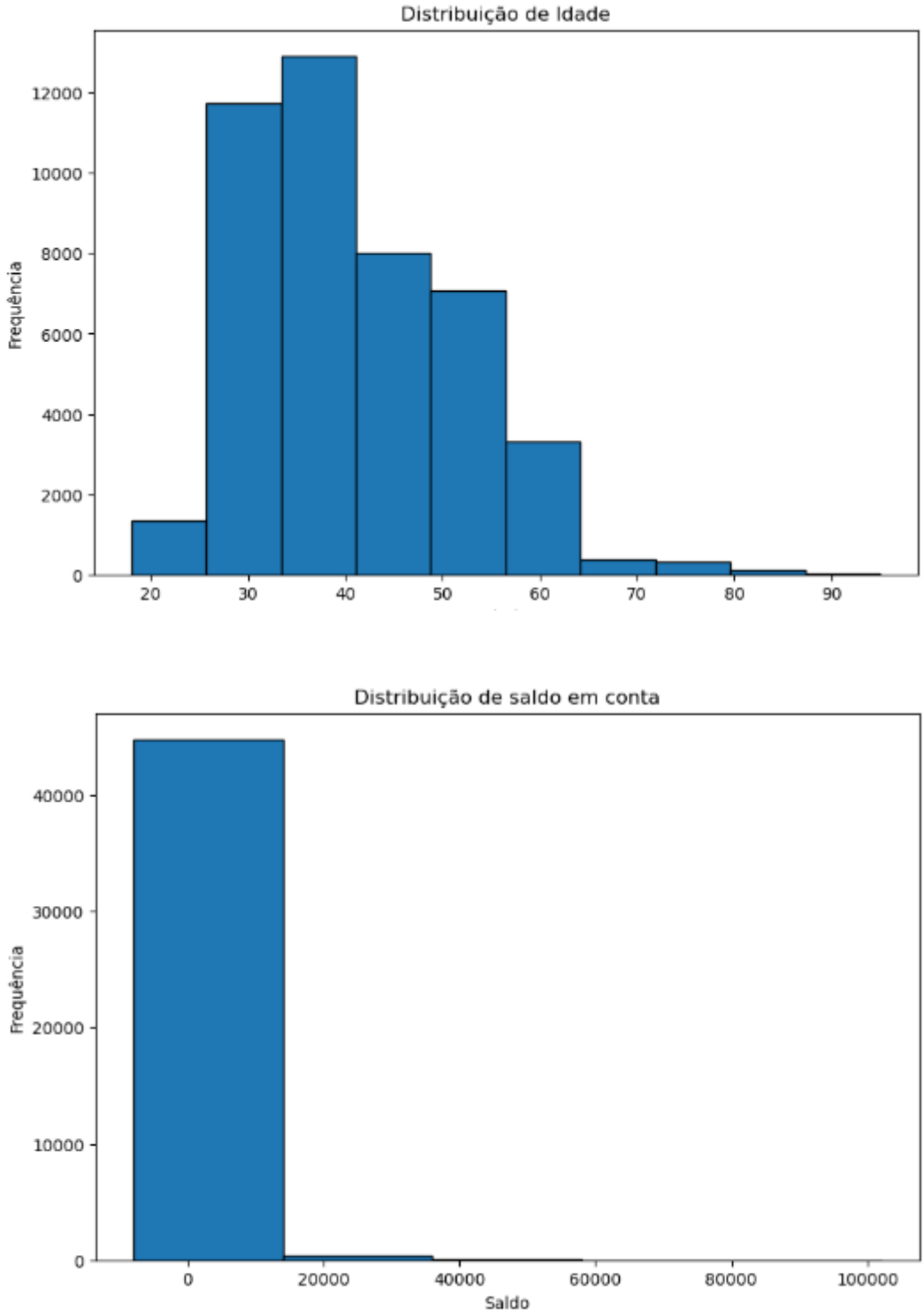

Figura C1 – Análise de distribuição das variáveis continuas

Fonte: Elaborado pelo Autor.

Apêndice D – Análise de *outliers* nas categorias quantitativas

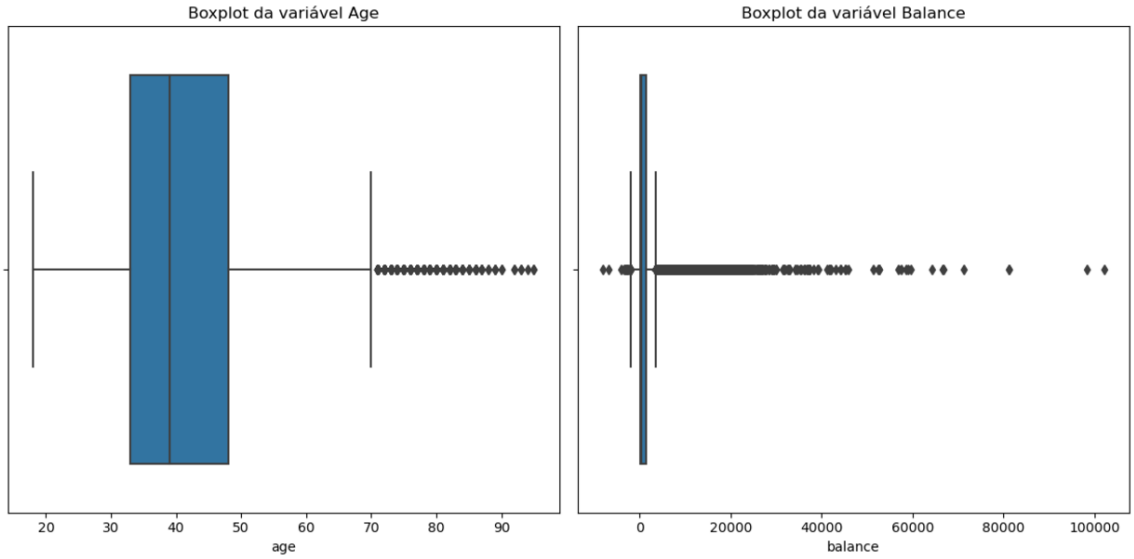

Figura D1 - Análise de *outliers* nas variáveis categóricas.

Fonte: Elaborado pelo Autor.

Apêndice E – Jupyter Notebook utilizado para a criação dos modelos

O link abaixo refere-se ao notebook utilizado para tratamento e análises dos dados, bem como para a avaliação dos modelos apresentados neste trabalho.

Disponível em: https://colab.research.google.com/drive/1s-8Bq1bPo_29LMuc66LaH7oGuDHYmBXb?usp=sharing