Universidade Federal de Uberlândia
Instituto de Matemática e Estatística

Bacharelado em Estatística

ALGORITMOS DE APRENDIZADO DE MÁQUINA NO ESTUDO DA INADIMPLÊNCIA EM UMA INSTITUIÇÃO FINANCEIRA

Tatiane Moreira Caetano

Uberlândia-MG
2024

Tatiane Moreira Caetano

ALGORITMOS DE APRENDIZADO DE MÁQUINA NO ESTUDO DA INADIMPLÊNCIA EM UMA INSTITUIÇÃO FINANCEIRA

Trabalho de conclusão de curso apresentado à Coordenação do Curso de Bacharelado em Estatística como requisito parcial para obtenção do grau de Bacharel em Estatística.

Orientador: Prof. Dr. José Waldemar da Silva

Uberlândia-MG

2024

Universidade Federal de Uberlândia
Instituto de Matemática e Estatística

Coordenação do Curso de Bacharelado em Estatística

A banca examinadora, conforme abaixo assinado, certifica a adequação deste trabalho de conclusão de curso para obtenção do grau de Bacharel em Estatística.

Uberlândia, ____ de _____ de 20____

BANCA EXAMINADORA

_____

Prof. Dr. José Waldemar da Silva

_____

Prof[a]. Dra. Maria Imaculada de Sousa Silva

_____

Prof[a]. Dra. Vânia de Fátima Lemes de Miranda

Uberlândia-MG

2024

RESUMO

A inadimplência é um desafio significativo para as instituições financeiras, impactando diretamente na sua saúde financeira e operacional. Identificar clientes propensos à inadimplência, com precisão e antecedência, pode contribuir para amenizar prejuízos ou para a maximização dos lucros. Os algoritmos de aprendizado supervisionado de máquina são úteis para a construção de modelos preditivos e neste caso em especial para classificação. Este estudo propõe uma análise comparativa entre três algoritmos de aprendizado de máquina para previsão de inadimplência, sendo eles a regressão logística, a árvore de decisão e a floresta aleatória aplicados aos dados originais, não normalizados e normalizados. Estas seis análises foram realizados com os dados não balanceados e balanceados pelas técnicas de SMOTE e ADASYN. O balanceamento consiste na geração de dados sintéticos na(s) classe(s) minoritárias, neste caso, na classe dos inadimplentes, para que o modelo possa ser bem treinado para realizar predições também nestas classes. Portanto, foram realizadas dezoito análises diferentes. A validação dos modelos é um ponto importante e consiste em aplicar o modelo treinado a uma parte dos dados, não utilizados no treinamento, e avaliar a sua performance por meio de alguma métrica de desempenho. Neste trabalho foram utilizadas as métricas acurácia, recall e F1-score. A metodologia, por meio da linguagem de programação Python, foi aplicada a um conjunto de dados do banco Stone composto por 7.081 clientes e 14 variáveis previsoras. Uma análise exploratória inicial foi realizada e não foram detectadas inconsistências ou dados faltantes. Identificou-se também, graficamente, que a distribuição das variáveis, em geral, são similares nas duas categorias, adimplentes e inadimplentes. Os resultados obtidos a partir da aplicação dos algoritmos, num total de dezoito análises, revelam que o melhor modelo foi obtido a partir do algoritmo floresta aleatória, nos casos com balanceamento, independente do tipo. A normalização dos dados não produziu alterações importantes nas métricas e portanto, para dados como estes considerados neste trabalho, bem como para os algoritmos utilizados, a normalização não se faz necessária. Haja vista a similaridade do algoritmo árvore de decisão com o algoritmo floresta aleatória, o primeiro pode ser indicado devido a sua simplicidade quando comparado ao segundo e, além disso, o algoritmo árvore de decisão demanda menor esforço computacional.

Palavras-chave: Árvore de Decisão, Aprendizado de Máquina, Floresta Aleatória, Inadimplência, Regressão Logística.

ABSTRACT

Default is a significant challenge for financial institutions, directly impacting their financial and operational health. Identifying customers prone to default, accurately and in advance, can help to mitigate losses or maximize profits. Supervised machine learning algorithms are useful for building predictive models and in this case in particular for classification. This study proposes a comparative analysis between three machine learning algorithms for default prediction, namely logistic regression, decision tree and random forest applied to original, non-normalized and normalized data. These six analyzes were performed with unbalanced and balanced data using the SMOTE and ADASYN techniques. Balancing consists of generating synthetic data in the minority class(es), in this case, in the class of defaulters, so that the model can be well trained to make predictions also in these classes. Therefore, eighteen different analyzes were carried out. Model validation is an important point and consists of applying the trained model to a part of the data, not used in training, and evaluating its performance using some performance metric. In this work, the metrics accuracy, recall and F1-score were used. The methodology, using the Python programming language, was applied to a Stone bank data set consisting of 7,081 customers and 14 predictor variables. An initial exploratory analysis was performed and no inconsistencies or missing data were detected. It was also identified, graphically, that the distribution of variables, in general, are similar in the two categories, compliant and non-compliant. The results obtained from the application of the algorithms, in a total of eighteen analyses, reveal that the best model was obtained from the random forest algorithm, in balanced cases, regardless of the type. Data normalization did not produce important changes in the metrics and therefore, for data like those considered in this work, as well as for the algorithms used, normalization is not necessary. Given the similarity of the decision tree algorithm with the random forest algorithm, the first can be indicated due to its simplicity when compared to the second and, in addition, the decision tree algorithm requires less computational effort.

Keywords: Decision Tree, Default, Logistic Regression, Machine Learning, Random Forest.

SUMÁRIO

1. INTRODUÇÃO

A inadimplência no Brasil vem crescendo consideravelmente há alguns anos. Desde de 2016, o país alcançou o maior número de inadimplentes, em maio de 2022, desde o início da série histórica, feita pela Serasa (órgão de Centralização de Serviços dos Bancos) [5]. É um fator de grande relevância no contexto econômico e financeiro do país. Este tema desperta o interesse de acadêmicos, profissionais do setor financeiro e gestores públicos, pois tem impactos significativos na estabilidade do sistema financeiro, na economia como um todo e na vida das pessoas. A inadimplência bancária não está fora deste contexto e se caracteriza pelo não pagamento de dívidas e obrigações financeiras por parte dos clientes. Este tipo de inadimplência também representa um desafio complexo que envolve questões econômicas, sociais, políticas e regulatórias.

O não cumprimento das obrigações financeiras gera impactos profundos para empresas bancárias e por isso oferecem métodos e formas de pagamentos facilitados afim de auxiliar o devedor a quitar suas dívidas. Fatores como crises financeiras, que historicamente acontecem no país, a corrupção, a falta de confiança no mercado financeiro brasileiro, por parte dos investidores, instabilidade política e a pandemia do COVID-19 que teve um impacto significativo na economia brasileira, podem gerar o aumento do desemprego e por consequência ao aumento da inadimplência, tornando-a mais comum na vida do cidadão brasileiro. Este fato faz, por um lado, com que as empresas sofram com a falta ou atraso de pagamento das dividas e, por outro, com que os clientes tenham que arcar com altas taxa de juros e com a falta de crédito. Para se ter uma visão do quão crítico é a situação, em 2022, o número de famílias com dívidas em atraso atingiu o maior patamar desde março de 2010 (Confederação Nacional do Comércio de Bens, Serviços e Turismo, 2022) [9].

As empresas bancárias não devem enxergar seus clientes inadimplentes como vilões. Mesmo com atraso no pagamento das dívidas o objetivo das empresas continua sendo receber o valor devido de maneira a maximizar os lucros. Evitar ou minimizar a inadimplência é uma forma de maximizar lucros e por isso, as empresas buscam formas mais seguras na escolha do cliente para qual será concedido o crédito. A identificação do perfil de cliente mais ou menos propício à inadimplência auxilia na decisão sobre a concessão do crédito. Portanto, as empresas buscam compreender padrões para evitar ou minimizar a inadimplência [11].

O aprendizado de máquina, também conhecido como machine learning, é uma subárea de inteligência artificial que se concentra em desenvolver algoritmos e modelos capazes de aprender e melhorar seu desempenho a partir de dados. Essa técnica permite que sistemas computacionais adquiram conhecimento a partir de exemplos e experiências, ao invés de serem programados de forma lógica.

O aprendizado de máquina desempenha um papel crucial nas instituições financeiras, permitindo a

análise de dados de clientes para prever com soluções a inadimplência. Essa ferramenta possibilita a criação de modelos estatísticos sofisticados que podem examinar o perfil do cliente, incluindo histórico de crédito, comportamento de compra e variáveis pessoais, para identificar padrões de risco de inadimplência. Ao analisar grandes conjuntos de dados, o aprendizado de máquina pode fornecer *insights* valiosos para a tomada de decisões financeiras, auxiliando na concessão de crédito responsável e estabelecimento de limites de crédito adequados ao perfil do cliente. Isso não apenas protege as instituições financeiras contra perdas, mas também ajuda a oferecer uma experiência mais personalizada aos clientes, promovendo a confiança e a fidelização.

O uso de algoritmos de aprendizado de máquina na avaliação de inadimplência, tem-se tornando cada vez mais comum com, por exemplo, o uso desta técnica para identificar o melhor o modelo na avaliação do risco de crédito em bancos [28]. Outro exemplo de pesquisa neste sentido, identifica a alta precisão preditiva trazida por tecnologias sofisticadas em modelos de aprendizado de máquina [17].

Portanto trazer esse tema à tona é crucial, pois a aplicação de aprendizado de máquina no setor financeiro representa uma revolução na forma como as instituições gerenciam riscos e tomam decisões. Com a crescente complexidade do mercado financeiro e a necessidade de maior precisão na avaliação de crédito, essas tecnologias oferecem uma solução robusta para tentar prever inadimplências.

Ao longo deste trabalho, serão apresentados o estudo de caso, incluindo análises exploratórias e resultados obtidos a partir da aplicação dos algoritmos, que permitirão uma visão abrangente sobre o tema. Especificamente objetiva-se obter um modelo acessível para empresas bancárias, em que a partir dos dados do cliente teriam uma resposta rápida para tomada de decisão. O trabalho será dividido nos capítulos a seguir:

- Objetivo
- Metodologia, onde será apresentada a forma de trabalho para chegar em um modelo viável
- Resultados, apresentação do resultado do trabalho afim de descobrir a qualidade do modelo criado
- Conclusões

Estes capítulos serão apresentados a seguir.

1.1 OBJETIVOS

Este trabalho tem como objetivo utilizar algoritmos de aprendizado de máquina supervisionado para treinar modelos de classificação de clientes em adimplentes ou inadimplentes. Para tanto, será realizada uma análise detalhada de uma base de dados da instituição financeira Stone como exemplo.

Especificamente objetiva-se treinar o modelo a partir dos algoritmos, regressão logística, árvore de decisão e floresta aleatória, com dados não normalizados e normalizados, não balanceados e balanceados e com validação cruzada em todos os casos.

As análises, incluindo a implementação dos algoritmos, serão realizadas com auxilio da linguagem Python [27].

2. METODOLOGIA

Algoritmos de aprendizado de máquina para treinamento de modelos de classificação foram utilizados para atender o objetivo deste trabalho que é desenvolver um modelo para classificação [12] de clientes em adimplentes ou inadimplentes em função de algumas variáveis previsoras, as quais serão apresentadas em 2.1. Características como o desbalanceamento dos dados devido ao número de observações expressivamente diferente nas duas categorias da variável resposta e variáveis previsoras mensuradas em escalas diferentes podem comprometer a qualidade do modelo final [13].

A validação cruzada também foi utilizada neste trabalho e é uma técnica interessante na avaliação e/ou validação de modelos pois utiliza a rotatividade entre a parte dos dados utilizada para treinamento e a parte utilizada para validação na obtenção de k modelos e não apenas um. A parte dos dados reservada para validação em um modelo, por exemplo na validação do primeiro modelo, se juntará à parte dos dados para treinamento dos outros $k-1$ modelos. Desta forma, são possíveis k valores da métrica de validação e a magnitude destes valores, resumida na média por exemplo, bem como a dispersão, podem ser indicativos da qualidade destes modelos.

O entendimento ou descrição das variáveis envolvidas em qualquer análise estatística é de suma importância e portanto, análises exploratórias também foram realizadas neste trabalho.

Portanto em seções específicas, neste capítulo, são apresentadas a metodologia de análise exploratória, os algoritmos utilizados, as técnicas de balanceamento, a normalização para amenizar o problema de escala diferente das previsoras e a validação cruzada. Além disso, no Apêndice A é apresentado o código utilizado para a implementação das análises realizadas durante a execução deste trabalho.

2.1 ANÁLISE EXPLORATÓRIA

Os dados utilizados são oriundos da instituição financeira Stone. A base consiste em informações pessoais e bancárias dos clientes da empresa e nela é possível encontrar informações sobre: idade, sexo, dependentes, escolaridade, estado civil, salário anual, tipo de cartão, meses de relacionamento, quantidade de produtos, iterações no período de 1 ano, meses inativos no período de 1 ano, limite de crédito, valor de transações em 1 ano, quantidade de transações em 1 ano, inadimplente (sim ou não). Estas foram as variáveis utilizadas neste trabalho em que inadimplência, foi a variável resposta.

A análise estatística exploratória tem por objetivo resumir ou descrever uma série de dados, permite uma visão geral dos dados, possibilita identificar padrões e portanto descrever o perfil geral dos clientes. A análise descritiva, na fase de pré-processamento dos dados, possibilita identificar inconsistência, como valores negativos quando a variável assume somente valores positivos, idade negativa

por exemplo, valores discrepantes, idade de 300 anos por exemplo e dados ausentes. Neste trabalho foram construídos gráficos paras as variáveis previsoras, por categoria de inadimplência.

Os dados, em se tratando de análise exploratória, geralmente são apresentados em tabelas ou em gráficos. Em tabelas é possível mostrar o resumo numérico dos dados seja por meio de medidas resumo ou por meio de distribuições empíricas. Por outro lado, gráficos, além da apresentação visual permitem rápida extração de informações [21] e também podem conter informações numéricas.

2.2 ALGORITMOS

Nesta seção, serão apresentados os algoritmos empregados na realização deste trabalho, a diversidade de técnicas disponíveis para abordar um mesmo problema ressalta a importância de uma análise criteriosa na seleção dos algoritmos mais adequados. No contexto deste trabalho, aplicou-se os algoritmos de Machine para construir modelos de classificação dos clientes em adimplentes ou inadimplentes. Os selecionados foram, regressão logística, árvore de decisão e floresta aleatória, para construir modelos de classificação dos clientes em adimplentes ou inadimplentes, que se destacam pela sua flexibilidade na manipulação de variáveis categóricas e numéricas, e além disso são amplamentes utilizados conforme a literatura.

2.2.1 REGRESSÃO LOGÍSTICA

A regressão logística é uma técnica estatística amplamente utilizada para modelar a relação entre uma variável dependente binária, portanto não contínua, que assume dois estados, sucesso ($Y = 1$) ou fracasso ($Y = 0$) e uma ou mais variáveis independentes. A regressão logística tem como objetivo modelar a probabilidade de ocorrência de um certo evento.

$$\pi(x) = E[Y|X] = P[Y = 1|X].$$

A regressão linear é uma técnica poderosa para modelar a relação entre variáveis independentes e dependentes contínuas. No entanto, ao lidar com variáveis dependentes binárias, a regressão linear pode resultar em previsões inadequadas e duas dessas razões são apresentadas a seguir:

1 Na regressão linear assume-se que os resíduos seguem uma distribuição normal. Entretanto, quando a variável dependente é binária, esta suposição é violada, já que os resíduos podem assumir apenas dois valores (0 ou 1), levando a uma distribuição não normal (Y tem uma distribuição Bernoulli);

2 Na regressão linear admite-se que os erros são homoscedásticas, e que portanto a variância da variável dependente não é proporcional ao seu valor esperado, essa suposição também é violada para o caso em que a variável resposta tenha distribuição Bernoulli, o que pode levar a resultados imprecisos ou enviesados na previsão $Var[Y] \propto E[Y]$.

Uma alternativa portanto, seria modelar a probabilidade de sucesso por meio de técnicas usuais de regressão linear. No entanto, por meio desta metodologia, previsões de probabilidades poderiam não pertencer ao intervalo de [0,1]. Além disso a relação entre as variáveis independentes e a probabilidade de sucesso pode não ser linear. Por meio da regressão logística, através da função logística, pode-se resolver esses dois problemas pois com esta modelagem obtém-se valores no conjunto dos números reais e considera-se a não-linearidade entre a probabilidade e as regressoras.

A regressão logística se baseia na suposição de que o logaritmo da chance de ocorrência de um evento tem uma relação linear com uma ou mais variáveis preditoras. A relação é expressa pela função logit, que é representada matematicamente na Eq. 2.1

$$ln\left(\frac{\pi}{1-\pi}\right) = logit(\pi) = \beta_0 + \beta_1 x, \tag{2.1}$$

em que π representa a probabilidade de ocorrência do evento de interesse, enquanto x é a variável preditora. Os coeficientes β_0 e β_1 são estimados, em geral pelo método da máxima verosimilhança e representam os efeitos da variável preditora na probabilidade de ocorrência do evento. É importante notar que π está contida no intervalo [0, 1], representando probabilidades e a transformação $ln\left(\frac{\pi}{1-\pi}\right)$ abrange todos os valores reais entre $-\infty$ e $+\infty$, o que permite a modelagem linear. Essa transformação logística é fundamental para garantir que a relação entre as variáveis preditoras e a probabilidade do evento seja adequada e linear.

A probabilidade de sucesso condicional à variável explicativa é expressa pela Eq.2.2:

$$E[Y|X] = \pi(x) = P[Y = 1|x] = \frac{exp(\beta_0 + \beta_1 x)}{1 + exp(\beta_0 + \beta_1 x)}, \tag{2.2}$$

onde $\pi(x)$ representa a probabilidade de sucesso dado um conjunto de variáveis explicativas x e β_0 e β_1 são os coeficientes a serem estimados. A função logística apresentada na Eq. 2.1 garante que a probabilidade estimada permaneça entre 0 e 1, o que é fundamental para eventos binários. Quanto maior o valor de $\beta_0 + \beta_1 x$, maior será a probabilidade de sucesso.

Na regressão logística, assim como em outras modelagens, a função de verossimilhança desempenha um papel crucial no ajuste do modelo aos dados observados. Ela é definida como o produto das funções de densidade ou de probabilidade para cada observação, dadas as variáveis explicativas e os parâmetros do modelo.

Define-se a função de verossimilhança na Eq. 2.3, em que $f(y_i;\theta)$ é uma função de densidade ou de probabilidade e é considerada uma função de θ para cada y fixo.

$$L(\theta; y) = \prod_{i=1}^{n} f(y_i;\theta); \quad \theta \in \Theta \tag{2.3}$$

Para o caso discreto, $\prod_{i=1}^{n} f(y_i;\theta)$ é a probabilidade conjunta, sob independência, de obter a amostra que de fato foi observada. Essa probabilidade é maximizada durante o processo de ajuste do modelo,

a fim de encontrar os valores dos coeficientes β_0 e β_1 que melhor explicam os dados.

Supondo que Y é uma variável aleatória com distribuição Bernoulli, isso significa que Y pode assumir apenas dois valores possíveis, 0 e 1, representando a não ocorrência e a ocorrência do evento, respectivamente, e que uma amostra aleatória de n elementos (independentes) foi obtida, então, a função de verossimilhança representada na Eq. 2.4 é definida como o produto das probabilidades individuais de observar os resultados da amostra:

$$\begin{aligned} L(\beta_0, \beta_1, \ldots, \beta_k; y) &= \prod_{i=1}^{n} \pi^{y_i}(1-\pi)^{1-y_i} \\ = &\left[\frac{exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_k x_{ki})}{1 + exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_k x_{ki})}\right]^{y_i} \\ \times &\left[1 - \frac{exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_k x_{ki})}{1 + exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_k x_{ki})}\right]^{1-y_i}, \end{aligned} \tag{2.4}$$

onde $\beta_0, \beta_1, \ldots, \beta_k$ são os coeficientes a serem estimados; y é o vetor de observações da variável dependente y_i representa a *i-ésima* observação da variável dependente na amostra; π é a probabilidade de sucesso, definida pela função logística.

A Eq. 2.4 é então maximizada para obter os estimadores dos parâmetros do modelo ($\beta_0, \beta_1, \ldots, \beta_k$), no entanto, é comum maximizar o logaritmo da função de verossimilhança, pois isso simplifica a maximização e evita problemas numéricos com números muito pequenos.

Por fim, a função de verossimilhança incorpora as variáveis independentes $x_{1i}, x_{2i}, \ldots, x_{ki}$ e seus coeficientes correspondentes $\beta_1, \beta_2, \ldots, \beta_k$ através da função logística e esta, por sua vez, transforma a soma ponderada das variáveis independentes em uma probabilidade entre 0 e 1.

Essa formulação da regressão logística nos permite modelar efetivamente a probabilidade de um evento ocorrer com base em múltiplas variáveis independentes, muito comum em modelos de previsão.

2.2.2 ÁRVORE DE DECISÃO

Árvore de decisão é o nome dado a um dos métodos estatísticos mais populares para tarefas de classificação e previsão de dados. Nesse método, um problema complexo é decomposto em subproblemas mais simples. O funcionamento da árvore de decisão visa formar particionamentos que vão dividindo os dados em pequenos grupos. Essa divisão acontece com base nas características dos dados para que, no final, possa entrar com um novo registro e, através do modelo, identificar em qual classe, supondo um problema de classificação, esse novo registro se encaixa melhor. A raiz, os nós, as ramificações e as folhas da árvore têm uma representação particular dentro do algoritmo que será apresentado [7].

COMPONENTES

A construção de uma árvore ocorre por meio de particionamentos recursivos no espaço das covariáveis. Cada divisão é denominada um nó, e o resultado final de cada ramificação é chamado de folha, os componentes de uma árvore de decisão são:

- Nó Raiz - o primeiro nó na árvore
- Nó - um ponto na árvore entre dois ramos, no qual uma regra é declarada
- Ramificações - conexões de um nó a outro e representa as possíveis respostas à condição de um nó de decisão
- Folha - representam as decisões ou as predições finais da árvore.

Para prever uma nova observação utilizando a árvore, segue-se o seguinte procedimento: a partir do topo da árvore, verifica-se se a condição especificada no nó atual é satisfeita e continua as ramificações até alcançar uma folha [20].

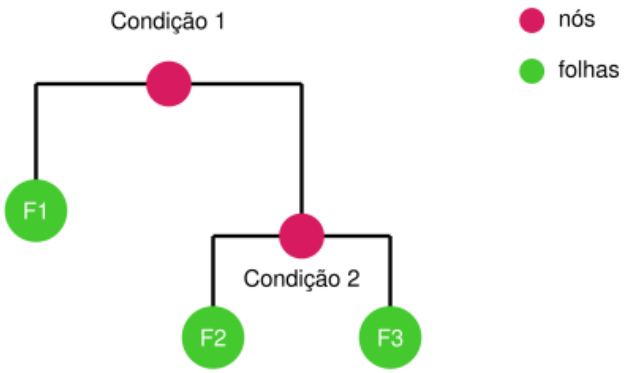

Figura 2.1: Exemplo de estrutura de uma árvore de decisão
Fonte: Imagem extraída do livro Aprendizado de Máquina[20]

No exemplo da Figura 2.1, se a condição 1 for satisfeita, a predição é determinada pela folha F1. Se não for, o caminho segue para a direita, onde outra condição é verificada. Se essa nova condição for satisfeita, a observação é prevista como F2; do contrário, é prevista como F3.

SELEÇÃO DAS VARIÁVEIS PARA DIVISÃO

Existem diversas análises disponíveis para determinar a melhor estratégia de divisão em nós da árvore. Essas análises são formuladas com base na distribuição das classes das observações antes e após a ocorrência da divisão.

O ponto inicial de uma árvore de decisão é chamado de "raiz", onde todas as instâncias da base de treinamento são agrupadas e é o ponto de partida para as ramificações subsequentes. A partir do nó raiz, a árvore se divide em dois filhos, um à direita e outro à esquerda conforme uma condição. Cada um desses filhos pode ser um nó adicional, se houver mais ramificações, ou uma folha, caso não haja mais subdivisões. Cada nó representa uma divisão nos dados de entrada, enquanto cada folha delimita uma área específica no conjunto de dados.

Um exemplo simples de representação da estrutura de uma árvore de decisão pode ser visualizado a seguir. Deseja-se tomar uma decisão de comprar ou não um automóvel baseado inicialmente em seu preço, depois consumo e por último os opcionais Figura 2.2.

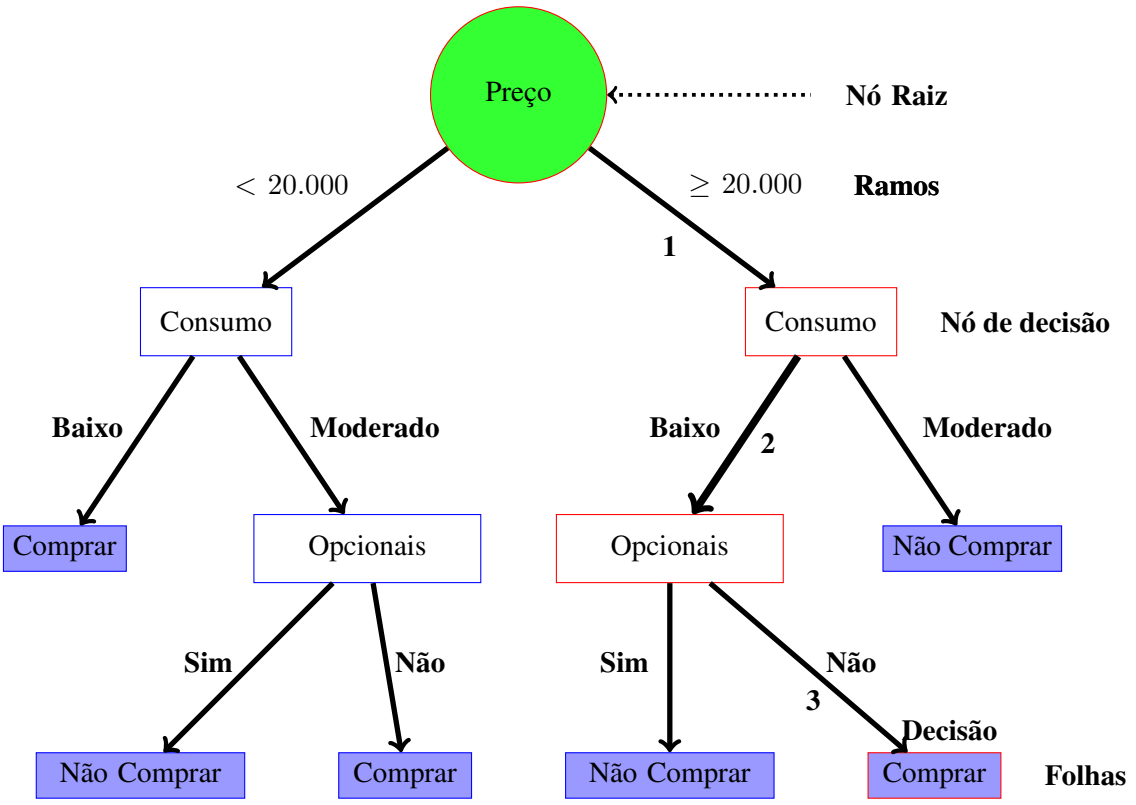

Figura 2.2: Ilustração de uma árvore de decisão.

Na Figura 2.2 o nó raiz se deu pela variável *preço* e isso acontece porque esta é a variável com o maior ganho de informação. Portanto, é a partir do *preço* que serão tomadas as próximas decisões.

Observa-se na ilustração (Figura 2.2) que a decisão foi por comprar um carro com preço maior ou igual a 20 mil (ramo **1**) e sendo assim, a decisão segue as ramificações do lado direito da árvore. Além disso, é possível observar que o próximo nó de decisão foi a variável *consumo* e que a decisão foi pelo consumo baixo. Portanto, esta decisão leva a um próximo passo por meio do ramo à esquerda (ramo **2**).

O próximo passo, após a decisão sobre o consumo, foi a decisão sobre os itens opcionais do veículo (*opcionais*), o que configura um nó e a partir deste, tem-se a decisão por um veículo sem opcionais (ramo **Não** ou ramo **3**).

Por fim, depois de cada decisão quanto ao preço, consumo e opcionais, tem-se ilustrado na Figura 2.2 o caminho percorrido ao longo dos nós até a folha (decisão). Este caminho é identificado pelos ramos **1**, **2** e **3**. Neste caso, as decisões tomadas (caminho) levou à folha que representa a resposta $comprar$.

A configuração da árvore de decisão quanto a ordem das variáveis e quanto a definição dos nós, se dá por meio de medidas de impureza, como o índice de Gini, que será apresentado a seguir.

Suponha, de maneira geral, uma base de dados cuja variável alvo é Y e as covariáveis são $x_1, x_2, \ldots, x_P$ e que seja formada por n observações. Desta forma Y_i é o valor da variável alvo para a observação i e x_{ij} é o valor da covariável j para a observação i. Posto isso, segue a ideia principal para a construção da árvore ou modelo a partir do algoritmo árvore de decisão para prever o valor de Y.

No nó raiz chegam todos as observações. Busca-se então a covariável X_j e uma constante (cte), ou valor específico desta covariável, de tal forma que a base de dados fique definida em duas categorias $A1 = \{i|X_{ij} \leq cte\}$ e $A2 = \{i|X_{ij} > cte\}$. por exemplo, na Figura 2.2 para $J = 1$ tem-se que X_1 é o preço e a cte é 20 mil.

Tanto a escolha de X_j quanto da cte deve implicar no menor grau impureza entre todos as possíveis partições formadas com diferentes valores de j e cte. A medida de impureza a ser utilizada depende se a árvore é de classificação ou de regressão [1].

ÍNDICE DE GINI

O índice de Gini (IG), apresentado na Eq. 2.5, também chamado de coeficiente ou razão de Gini, é uma medida de impureza e foi concebido pelo estatístico italiano Corrado Gini em 1912 [18]. Nos algoritmos de classificação, esse índice estatístico de dispersão tem um papel fundamental na avaliação da heterogeneidade dos dados e na seleção de atributos.

$$IG = 1 - \sum_{i=1}^{c} p_i^2. \tag{2.5}$$

A expressão para o índice de Gini é apresentada na Eq. 2.5, em que c equivale ao número de classes e p_i representa a proporção das observações da região pertencentes a classe i. O índice de Gini assume um valor próximo de zero se os conjuntos definidos por um nó apresentarem predominantemente observações de instâncias pertencentes a mesma classe e se afasta de zero quando os conjuntos apresentam as observações distribuídas igualmente entre todas as classes [1].

Uma das grandes vantagens do modelo de árvore de decisão é sua habilidade de lidar tanto com dados categóricos quanto numéricos, além de sua facilidade de interpretação. As árvores de decisão podem ser visualizadas e compreendidas facilmente, o que auxilia na explicação das decisões tomadas pelo modelo.

No entanto, árvores de decisão também possuem algumas limitações. Elas podem ser sensíveis a pequenas variações nos dados de treinamento, o que pode resultar em problemas de sobreajuste. Para lidar com isso, técnicas como o uso de conjuntos de árvores, como o algoritmo Floresta Aleatória, podem ser aplicadas.

2.2.3 FLORESTA ALEATÓRIA

O algorítimo de floresta aleatória, tem por definição, um conjunto de árvores de decisões, conforme apresentado na subseção 2.2.2, para formar um único modelo com a maior assertividade possível.

No contexto da floresta aleatória, a seleção da variável para o nó raiz não é feita a partir de todas as variáveis disponíveis. Em vez disso, o algoritmo opta por escolher aleatoriamente duas ou mais variáveis e realiza os cálculos com base nas amostras selecionadas para determinar quais delas serão utilizadas no nó inicial.

No processo de construção da floresta, o procedimento da árvore aleatória é repetido, resultando na formação de uma nova árvore. É importante notar que esta árvore provavelmente será distinta da primeira, devido à aleatoriedade envolvida tanto na escolha das amostras quanto na seleção das variáveis. É possível criar um número arbitrário de árvores, e quanto mais árvores forem construídas, geralmente os melhores serão os resultados do modelo. No entanto, existe um ponto de saturação em que adicionar mais árvores não contribuirá significativamente para a melhoria do desempenho do modelo[7].

Aplicando o modelo cada árvore terá seu resultado e a floresta, em problemas de regressão, utilizará a média dos valores previstos como resultado final e em problemas de classificação o resultado que for mais abundante na votação individual será a resposta final.

Foi demonstrado por [3], que a taxa de erro de uma floresta de árvores de decisão depende da robustez das árvores individuais na floresta (sua taxa individual de erro) e da correlação entre suas classificações, ou seja, o algorítimo é dependente da qualidade de suas árvores, mas a utilização desse método tem a capacidade de melhorar a generalização do modelo e aumentar a capacidade preditiva, evitando o overfitting [10], isto é o sobreajuste, que seria um ótimo modelo para os dados de treino, mas um modelo ruim para novos dados.

2.3 SEPARAÇÃO DA BASE DE DADOS EM BASE DE TREINO E BASE DE TESTE

A separação da base de dados em conjuntos de treino e teste é uma prática fundamental na construção e avaliação de modelos em ciência de dados e aprendizado de máquina. Este processo visa garantir que o modelo seja treinado em uma parte dos dados e testado em outra parte independente, a fim de avaliar sua capacidade de generalização para novos dados não vistos.

No geral, uma base de dados é dividida em dois conjuntos: o conjunto de treino, que é utilizado para ajustar as configurações do modelo, e o conjunto de teste, que é utilizado para avaliar o desempenho do modelo em dados não visíveis durante o treinamento [8]. Essa divisão em treinamento e teste é realizada para avaliar se o modelo não apenas se ajusta bem aos dados de treinamento, mas também possui a capacidade de generalizar para novas observações durante o teste. Normalmente, as divisões mais utilizadas são 60:40%, 70:30% ou 80:20%, variando de acordo com o tamanho do conjunto de dados, sendo que a proporção maior é geralmente atribuída ao grupo de treinamento. Em geral, quanto mais dados disponíveis, maior será a proporção designada ao grupo de treinamento[24].

De modo geral, uma abordagem comum é escolhida aleatoriamente cerca de 20-30% dos dados para compor o grupo de teste, enquanto o restante é utilizado como grupo de validação. Ao separar os dados, é importante garantir que a distribuição das classes (no caso de problemas de classificação) ou a distribuição das variáveis dependentes e independentes seja mantida em ambos os conjuntos. Isso ajuda a evitar problemas de viés nos resultados. Este problema pode ser amenizado com técnicas de balanceamento (seção 2.4).

Além disso, é comum também utilizar a técnica de validação cruzada durante o treinamento, para melhorar a eficácia da validação do modelo e evitar overfitting (seção 2.7).

Na avaliação do modelo no conjunto de teste, métricas como acurácia, recall, F1 score (para problemas de classificação) ou erro quadrático médio (para problemas de regressão) são frequentemente utilizados. Neste trabalho em especial foram utilizadas a acurácia, o recall e o F1 score, que são apresentadas na seção 2.6.

Portanto, a separação da base de dados em conjuntos de treino e teste desempenha um papel crucial na construção de modelos confiáveis e robustos, fornecendo uma avaliação objetiva do desempenho do modelo em dados não visíveis.

2.4 TRATAMENTO PARA BASES DESBALANCEADAS

Para aumentar as chances de bons resultados de predição, o conjunto de dados deve ser balanceado, ou pelo menos, o desbalanceamento deve ser reduzido. Diz-se que os dados são balanceados quando a proporção de ocorrência das diversas classes dos rótulos é igual em todas essas classes, ou apresenta apenas pequenas diferenças, garantindo que todas as classes sejam representadas de maneira equilibrada por suas distribuições[16].

Uma base de dados desbalanceada pode provocar ineficiência nas previsões. Por exemplo, no caso de uma base com poucos casos de inadimplência, o algorítimo pode não contar com informações suficientes para treinar o modelo para prever tal categoria.

2.4.1 SMOTE

Para o tratamento de bases desbalanceadas, dentre as várias técnicas, existe a chamada SMOTE (*Synthetic Minority Oversampling Technique*) [4] que cria dados de maneira artificial através de interpolação [6]. Essa técnica emprega o algoritmo KNN (Vizinhos Mais Próximos) para gerar representantes a partir dos exemplos e seus k vizinhos. Durante cada iteração do algoritmo, um exemplo da classe com a menor ocorrência é selecionado. Em seguida, o vetor resultante da diferença entre o exemplo analisado e seu vizinho é multiplicado por um número aleatório que varia entre 0 e 1. Esse vetor modificado é então aplicado ao exemplo em questão, resultando na criação de um novo ponto no espaço, situado ao longo da reta que conecta o exemplo e seu vizinho. Nesse ponto, um novo elemento da classe minoritária é gerado.

Detalhadamente, o algorítimo SMOTE compreende os seguintes passos [4]:

- Escolha uma amostra da classe minoritária s_i;
- Aplique o algorítimo k vizinhos mais próximos KNN para identificar as k amostras da classe minoritária cuja distância euclidiana em relação a x_i sejam as menores;
- Escolha, de maneira aleatória, uma das amostras retornadas pelo passo anterior e chame-a de s_h;
- Calcule a amostra sintética s'_k a partir da Eq.2.6. A amostra é gerada a partir da soma dos valores de s_i com os valores de s_h multiplicados por um número σ no intervalo [0,1];

- Adicione s'_k no conjunto de dados;
- Repita os passos anteriores até que todas as amostras originais da classe minoritária do conjunto de dados tenham gerado uma amostra sintética.

$$s'_k = s_i + s_h \times \sigma \mid \sigma \in [0,1] \tag{2.6}$$

Na Figura 2.3 ilustra-se a execução de uma iteração do algoritmo SMOTE. Em (a) é possível ver o início com as amostras da classe minoritária e majoritária representadas por pontos verdes e azuis, respectivamente. Em (b), uma amostra da classe minoritária é selecionada (ponto preto) e seus 3 vizinhos mais próximos (pontos amarelos) são identificados com algorítimo KNN. Em (c), uma das amostras vizinhas é selecionada aleatoriamente (ponto marrom), e finalmente uma amostra sintética (ponto vermelho) é gerada e posicionada no segmento que conecta o ponto preto ao ponto marrom [4].

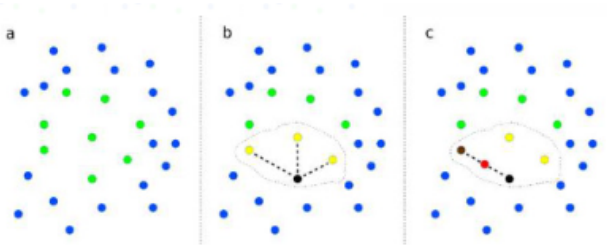

Figura 2.3: Ilustração do SMOTE

2.4.2 ADASYN

Outra técnica para o tratamento de bases desbalanceadas conhecida é o ADASYN (*Adaptive Synthetic Sampling*) [14] [6]. Sua abordagem envolve a geração de exemplos sintéticos para a classe minoritária, adaptando a taxa de geração com base na dificuldade percebida de classificação. O ADASYN avalia a dificuldade de classificação para cada exemplo da classe minoritária, medindo a proximidade desses exemplos à fronteira de decisão do modelo. A técnica dá mais atenção à geração de exemplos sintéticos para as observações minoritárias que estão mais distantes da fronteira de decisão, ou seja, aquelas que são mais difíceis de classificar corretamente. Ao contrário de métodos que geram exemplos sintéticos uniformemente em toda a classe minoritária, o ADASYN opera de maneira local, concentrando-se nas regiões onde a classe minoritária é menos representada. Durante a geração sintética, o ADASYN procura preservar a distribuição original dos dados, evitando introduzir viés significativo. A densidade é uma medida da proximidade das instâncias da classe minoritária umas das outras em um espaço de características, e é utilizada para adaptar a geração de amostras sintéticas de forma a lidar com o desbalanceamento de classes em conjuntos de dados. A taxa de geração de exemplos sintéticos é adaptada dinamicamente, proporcionando um equilíbrio entre melhorar o desempenho do modelo na classe minoritária e preservar a integridade dos dados originais. A ideal

principal do ADASYN é utilizar a distribuição de densidade ri como critério de decisão do número de dados sintéticos que serão gerados para classe minoritária. Esse critério da uma ênfase maior nas observações que são difíceis de generalizar [2].

Detalhadamente, o algorítimo ADASYN segue os seguintes passos: O calculo do grau de desbalanceamento, para ver se é realmente necessário a aplicação da técnica Eq.2.7,

$$d = \frac{m_s}{m_l}, \tag{2.7}$$

onde m_s e m_l são amostras da classe minoritária e majoritária, respectivamente. O número dados sintéticos gerados 2.8:

$$G = (m_s - m_l) \times \beta, \tag{2.8}$$

onde $\beta \in [0, 1]$ regula o nível de equilíbrio. O cálculo do peso da amostra 2.9,

$$r_i = \frac{\triangle_i}{K}, \tag{2.9}$$

onde $i = 1, ..., m_s$ e $\triangle_i$ é o número de exemplos nos K vizinhos de x_i que pertencem a classe majoritária, esse peso será normalizado de acordo com a Eq.2.10

$$R_i = \frac{r_i}{\sum_{i=1}^{m_s} r_i}, \tag{2.10}$$

onde R_i é a densidade de distribuição. E o número de dados gerados de cada observação da classe minoritária na Eq. 2.11

$$g_i = R_i \times G. \tag{2.11}$$

Caso uma observação minoritária tiver poucos vizinhos, então estará localizada em uma região onde a classe minoritária é menos representada. Nesse cenário, nota-se que uma diminuição de K (vizinhos) o que leva a um índice r_i maior. Isso é uma consideração importante para o ADASYN, já que o algoritmo ajusta a taxa de geração de amostras sintéticas com base na densidade local, permitindo uma geração adaptativa de amostras sintéticas para lidar com regiões do espaço de características onde a classe minoritária é escassa.

2.5 NORMALIZAÇÃO DE VARIÁVEIS

A normalização de variáveis é uma etapa crítica no pré-processamento de dados para algoritmos de previsão. Esta prática tem um impacto significativo no desempenho e na estabilidade dos modelos, aumentando as chances de que eles sejam mais eficazes na aprendizagem dos padrões presentes nos dados. Diferentes variáveis podem ter escalas e unidades de medida distintas, neste trabalho como

exemplo, uma variável do conjunto de dados inclui limite de crédito, a escala dessa variável pode variar drasticamente. Normalizar as variáveis para uma escala similar permite que o modelo atribua pesos adequados a cada uma delas durante o treinamento, evitando que uma variável com uma escala maior domine a contribuição para a previsão. Algoritmos de otimização utilizados para ajustar os parâmetros do modelo podem ser sensíveis a grandes diferenças de escala entre as variáveis. Isso pode resultar em problemas de convergência lenta ou até mesmo em divergência durante o treinamento. A normalização das variáveis ajuda a melhorar a estabilidade numérica desses algoritmos, facilitando o processo de otimização e garantindo resultados mais confiáveis. Em resumo, a normalização de variáveis desempenha um papel fundamental na construção de modelos de previsão precisos e confiáveis. Ao garantir que todas as variáveis estejam na mesma escala, essa prática ajuda a equilibrar a contribuição de cada variável para o modelo, melhora a estabilidade numérica dos algoritmos de otimização e facilita a interpretação dos resultados.

Para o trabalho foi utilizado o escalonador Mín-Máx, onde o menor valor corresponderá ao valor zero na nova escala e, o valor máximo corresponderá ao valor um. A biblioteca em Python MinMax Scaler, uma ferramenta valiosa para a normalização dos dados, fornecida pelo pacote Scikit-Learn (uma biblioteca de código aberto extremamente popular e poderosa para aprendizado de máquina, mineração de dados e análise estatística), reduz os dados dentro de um determinado intervalo, geralmente de 0 a 1. Ele transforma os dados escalando os valores para um determinado intervalo sem alterar a forma da distribuição original [26]. A formula por trás da função do Python é como apresentado na Eq. 2.12,

$$Z = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{2.12}$$

em que Z é o valor normalizado, X é o valor original, a ser normalizado e, X_{min} e X_{max} são, respectivamente, o menor e o maior valor dentre os valores a serem normalizados, de uma mesma variável.

2.6 MEDIDAS DE AVALIAÇÃO

A avaliação de modelos em ciência de dados e aprendizado de máquina desempenha um papel crucial na determinação de sua eficácia, confiabilidade e adequação para a aplicação desejada. Esta seção se dedica a explorar três das medidas de avaliação mais comuns e essenciais, utilizadas para analisar o desempenho de modelos preditivos. Medidas de avaliação são ferramentas fundamentais para quantificar o quão bem um modelo é capaz de generalizar para dados não vistos e capturar padrões relevantes. Elas fornecem insights valiosos para a melhoria dos modelos. A ineficácia do modelo para fazer previsões precisas, identificada a partir dos valores das métricas, indicam que o modelo deve ser melhor treinado, seja com mais dados ou com uso de outros algorítimos.

Nesta seção serão apresentadas as métricas de avaliação: acurácia, recall e F1 score, e as diferentes modelagens (3.3) serão comparadas por meio dos valores dessas métricas [23].

Os modelos de classificação construídos foram comparados com o intuito de determinar o melhor,

em termos de desempenho, quanto às métricas de avaliação. A interpretação e extração de informações destas medidas pode respaldar a tomada de decisão sobre o desenvolvimento ou ajuste e implantação de modelos preditivos em uma variedade de domínios de aplicação [25].

2.6.1 ACCURACY (ACURÁCIA)

A acurácia (Eq. 2.13) é uma medida básica de desempenho que indica a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões

$$Acuracia = \frac{VP + VN}{VP + VN + FN + FP} \tag{2.13}$$

em que VP e VN são, respectivamente, a quantidade de verdadeiros positivos e verdadeiros negativos.

Para a acurácia, valores mais altos são melhores. No entanto, ela pode ser enganosa em conjuntos de dados desbalanceados, onde uma classe pode ser muito mais frequente do que outras, haja vista a baixa capacidade do modelo para prever casos na(s) classe(s) minoritária(s).

2.6.2 RECALL (REVOCAÇÃO)

O recall (Eq. 2.14), também conhecido como sensibilidade, mede a proporção de instâncias positivas que foram corretamente identificadas pelo modelo

$$Recall = \frac{VP}{VP + FN} \tag{2.14}$$

em que VP e FN são, respectivamente, a quantidade de verdadeiros positivos e falsos negativos.

O recall é particularmente importante quando o custo de um falso negativo é alto. Valores mais altos de recall são desejáveis.

2.6.3 F1-SCORE

O F1-score (Eq. 2.16) é uma métrica que combina precisão e recall em uma única medida. Ele fornece uma média harmônica dessas duas métricas. Em que a precisão é conforme apresentada na Eq. 2.15,

$$Precisao = \frac{VP}{VP + FP} \tag{2.15}$$

e o F1-score é apresentado na Eq. 2.16,

$$F1 = 2 \times \frac{Precisao \times Recall}{Precisao + Recall}. \tag{2.16}$$

O F1-score é útil quando há um desequilíbrio entre as classes ou quando ambas as taxas de falsos positivos e falsos negativos precisam ser consideradas. Valores mais altos de F1-score indicam um

melhor equilíbrio entre precisão e recall.

2.7 VALIDAÇÃO CRUZADA

A validação cruzada desempenha um papel crucial no contexto da avaliação de modelos preditivos em machine learning. Ao enfrentar o desafio de desenvolver modelos que não apenas se ajustem bem aos dados de treinamento, mas também generalizem para novos dados, a validação cruzada oferece uma abordagem robusta. Este método envolve a divisão do conjunto de dados em subconjuntos de treino e teste, permitindo que o modelo seja treinado em diferentes partições e testado em conjuntos independentes.

A popular validação cruzada k-fold, por exemplo, divide os dados em k partes iguais, treinando o modelo em k-1 folds e validando em um fold diferente em cada iteração, neste trabalho foi ultilizado k=10 [15]. Essa abordagem proporciona uma avaliação mais confiável no desempenho do modelo, mitigando o risco de sobreajuste aos dados de treinamento.

A validação cruzada, portanto, não apenas fornece uma estimativa mais precisa do desempenho geral do modelo, mas também contribui para a identificação de possíveis fontes de variabilidade, fortalecendo assim a confiança na capacidade do modelo de generalizar para dados não observados.

Neste trabalho foi realizada a validação cruzada para todos os três algoritmos utilizados, a saber, regressão logística, árvore de decisão e floresta aleatória, para os dados normalizados e não normalizados e para os casos sem balanceamento e para os balanceamento por SMOTE e ADASYN, conforme apresentado na Seção 2.4, totalizando 18 análises.

3. RESULTADOS

Ao longo deste estudo, foram empregadas técnicas avançadas de análise de dados como o aprendizado de máquina para identificar padrões e tendências que possam indicar potenciais clientes inadimplentes. Neste contexto, este capítulo se propõe a apresentar de forma detalhada os resultados obtidos, incluindo análise das principais variáveis influentes na inadimplência e a avaliação do desempenho do modelo. Além disso, é explorado um exemplo de recomendação com base no melhor modelo treinado. A existência de um modelo de recomendação contribui para aprimorar as politicas de créditos das instituições financeiras e nesta aplicação em especial, para o banco Stone.

3.1 OS DADOS

Para a realização do trabalho foi utilizada uma base de dados da Stone, uma instituição de pagamento regulada e autorizada pelo Banco Central do Brasil, com 15 colunas, sendo a última uma indicadora de inadimplência e portanto, a classificação do cliente em inadimplente ou não. As demais, isto é, as outras 14 são características que permitem descrever o perfil dos clientes. Com o objetivo de melhor compreender as características ou variáveis, nos itens a seguir são apresentadas as categorias ou intervalo de ocorrência, se categórica ou numérica, para cada uma. Nesta análise prévia foi observada a presença de informação sobre todas as características para todos os indivíduos, em outras palavras, não foram observados dados faltantes.

- Idade: entre 26 e 73 anos
- Sexo: Feminino e Masculino
- Números de Dependentes: entre 0 e 5
- Escolaridade: Sem educação formal, Ensino Médio, Graduação, Mestrado e Doutorado.
- Estado civil são: Casado, Divorciado e Solteiro.
- Faixa salarial: Menos que 40 mil reais, entre 40 e 60, entre 60 e 80, entre 80 e 120 e mais que 120 mil reais.
- Tipo de cartão de crédito: blue, gold, platinum e silver

Em geral o tipo de cartão é definido através do limite disponível para o cliente, quanto maior o seu limite maior os privilégios disponibilizados no cartão, sendo o inicial (com menos benefícios) o Blue e o último (com mais benefícios) o Silver.

3.2 LIMPEZA E ANÁLISE EXPLORATÓRIA DOS DADOS

A partir da análise exploratória não foram encontradas observações atípicas e, portanto, a implementação dos algoritmos foi realizada com a totalidade dos dados com 7.081 clientes, isto é, o que configura também a quantidade de linhas do banco de dados.

A fim de identificar o comportamento de cada uma das variáveis previsoras nas duas categorias de inadimplência (sim ou não), são apresentados os gráficos nas Figuras 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13 e 3.14.

Foi observado que a distribuição da variável idade, conforme apresentado na Figura 3.1, é similar nas duas categorias. Em ambos os casos tem-se uma distribuição simétrica com maior concentração de clientes entre as idades de 40 a 55 anos. Além disso a idade mínima de 26 anos abrange os dois tipos de categoria de clientes e a máxima de 73 anos com apenas um caso, sendo de adimplência.

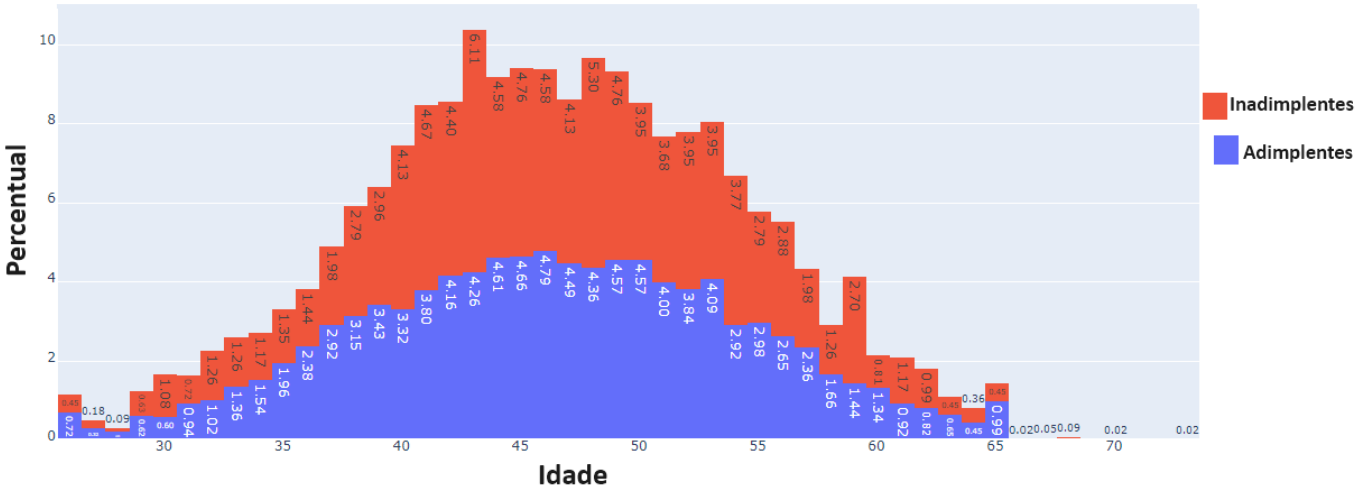

Figura 3.1: Distribuição das idades por categoria de inadimplência.

Quanto a variável sexo (Figura 3.2) e número de dependentes (Figura 3.3) , também se observa que são dados bem equilibrados por categoria de adimplentes e inadimplentes. Vale ressaltar que em número dependentes tem-se a maior concentração de dados entre 2 e 3 dependentes, mas mesmo assim o equilibro das categorias permanece.

Quanto a variável salário anual (Figura 3.4), observou-se uma concentração de clientes em salário menor que 40 mil, porém com relação a categoria de inadimplência os dados permanecem bem distribuídos, visto seus percentuais bem próximos em cada faixa de salário anual.

Sobre o nível de escolaridade (Figura 3.5), é possível observar uma concentração de dados em mestrado e ensino médio, mas os percentuais entre cada categoria são similares e portanto não é

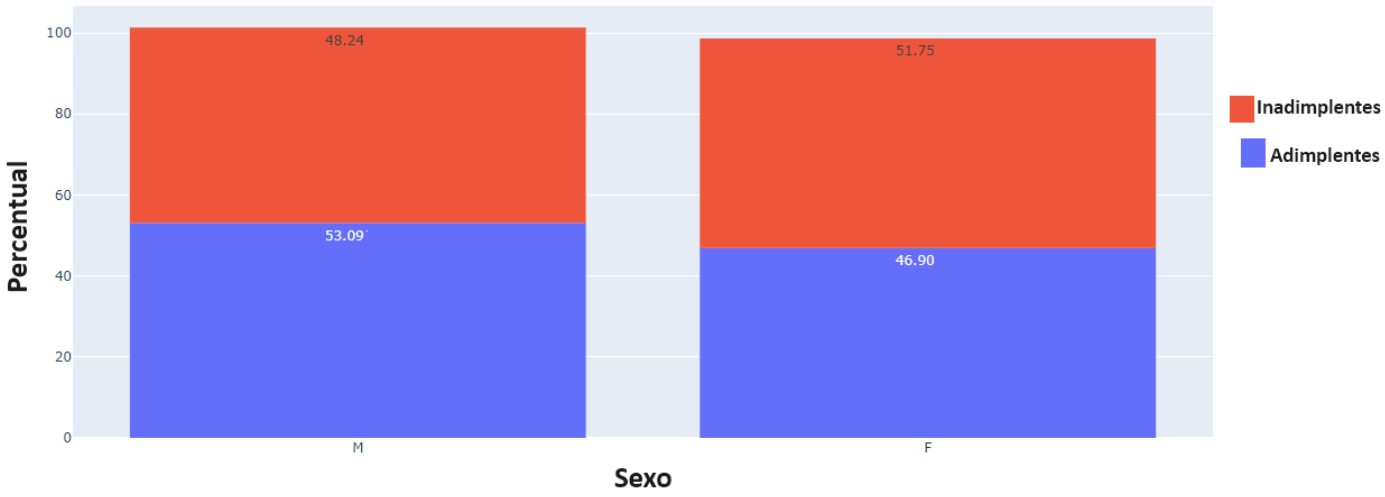

Figura 3.2: Distribuição do sexo por categoria de inadimplência.

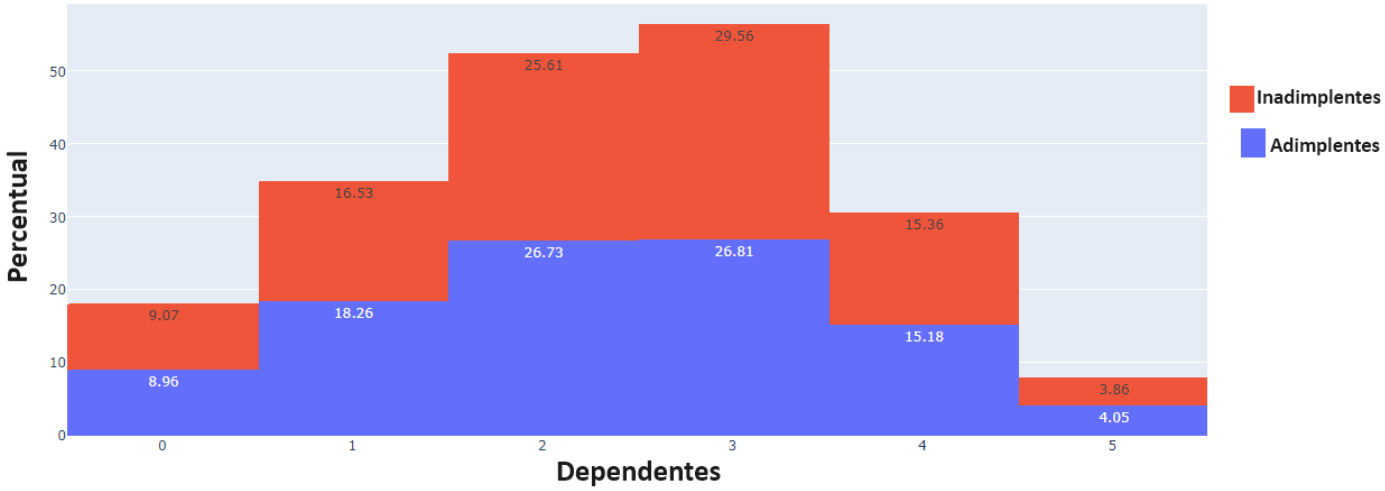

Figura 3.3: Distribuição do número de dependentes por categoria de inadimplência.

possível observar alguma categoria com maior propensão a inadimplência.

Nota-se com a variável tipo de cartão (Figura 3.6), que mais de 90% usam o cartão tipo Blue o que não implica análise sobre a inadimplência, devido a concentração de dados em um único tipo.

Sobre os meses de relacionamento (Figura 3.7), tem-se por volta de 25% dos clientes concentrados em 36 meses de relacionamento, tanto para clientes adimplentes, quanto inadimplentes e o restante deles dissipados entre 0 e 56 meses de relacionamento. Portanto observa-se uma grande variedade de cadastro de clientes dentro da base da instituição.

Quanto ao estado civil (Figura 3.8), a maioria dos clientes estão entre casados ou solteiros, portanto não é possível observar padrões diferentes no comportamento dos clientes inadimplentes e adimplentes, concluindo que as diferentes categorias não tem relação com a inadimplência.

Analisando a quantidade de produtos que o cliente usufrui da instituição (Figura 3.9), é possível observar que quem possui 1 ou 2 produtos tendem a ter um percentual maior de inadimplentes, já quando chega em 3 nota-se um certo equilíbrio em percentual e de 4 pra frente o maior percentual

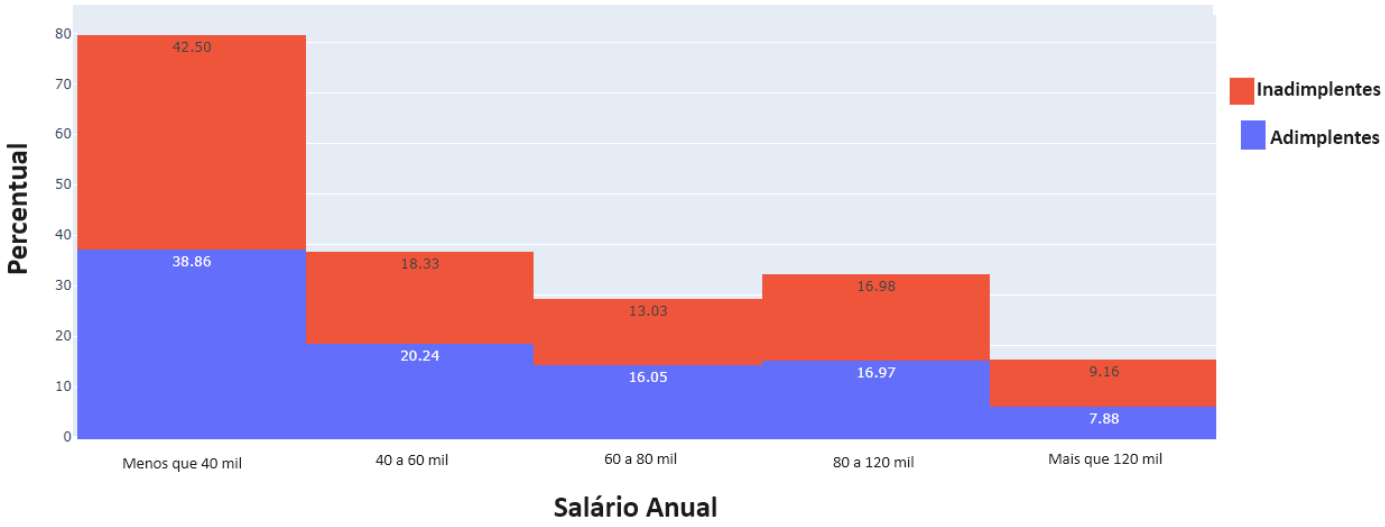

Figura 3.4: Distribuição do salário anual por categoria de inadimplência.

muda para adimplentes.

Em iterações dos últimos 12 meses (Figura 3.10), observa-se que tanto inadimplentes quanto adimplentes tem entre 2 e 3 iterações na maioria dos casos, porém ao falar de 2 iterações o percentual de adimplentes tem quase 10% a mais de clientes e ao falar de 3 iterações o percentual de inadimplentes é quase 11% de clientes. Quando se trata de 1 iteração o percentual de adimplentes é maior aproximadamente 8% e quando se trata de 4 iterações o cenário muda para o maior percentual de inadimplentes com aproximadamente 7%.

Quanto aos meses inativos (Figura 3.11), tem-se uma concentração de 51% dos inadimplentes em 3 meses e dos adimplentes apenas 36% e um cenário contrário em 2 meses que dita 24% para adimplentes e 5% para inadimplentes o que sugere talvez um ponto de atenção a esse tipo de perfil com meses de inatividade.

Em limite de crédito (Figura 3.12), é possível observar uma maior concentração de inadimplentes até mil reais de limite, o que seja talvez outro ponto de atenção, quanto a adimplentes é possível observar o maior percentual entre mil e 2 mil reais.

Ao observar os valores das últimas transações (Figura 3.13), tem-se mais de 50% da base de inadimplentes entre 2 e 2,5 mil reais, em quanto na base de adimplentes esse valor passa para entre 4 e 4,5 mil com 30% da base.

E por último, observa-se a quantidade de transações (Figura 3.14), que possui um cenário interessante, para clientes adimplentes a curva de maiores percentuais entre 60 e 90 transações e para clientes inadimplentes entre 30 e 50 transações, permitindo perceber que possa existir uma questão sobre quanto menor o número de transações no ano, mais a chance do cliente ser inadimplente. Por fim, é possível concluir que clientes que inadimplentes tendem a fazer menos transações anuais e que os valores dessas transações são mais baixos quando comparado aos clientes adimplentes.

Apesar da similaridade das distribuições da maioria das variáveis por categoria de inadimplência, foi observado apenas o valor e a quantidade das transações com uma certa relação para a inadimplência,

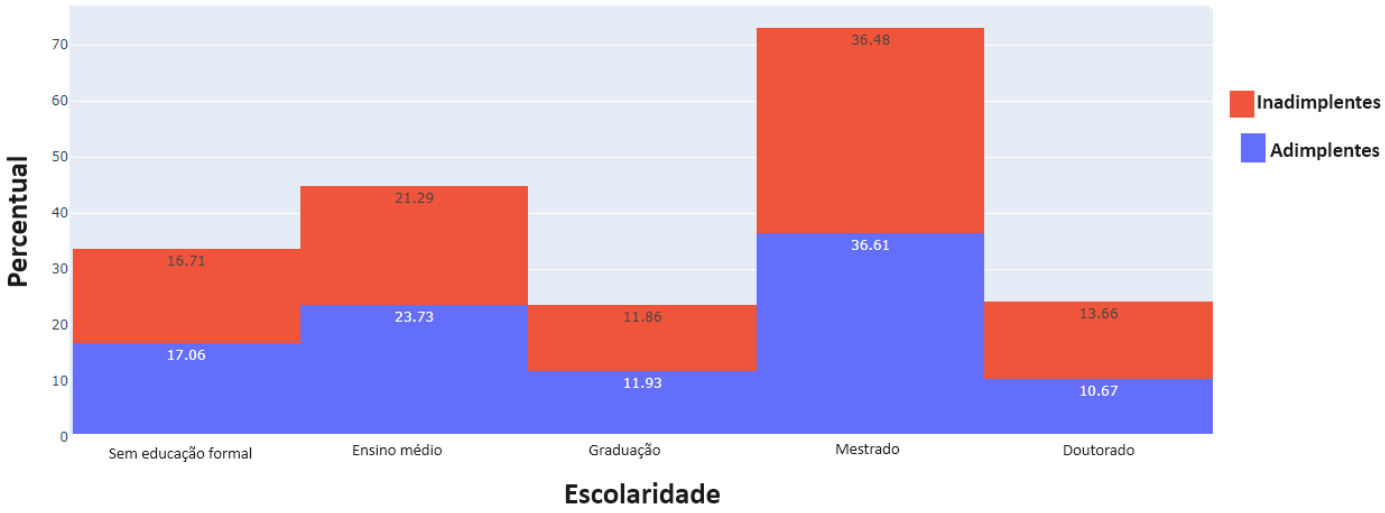

Figura 3.5: Distribuição da escolaridade por categoria de inadimplência.

este fato justifica o uso dos algorítimos para treinamento dos modelos de predição. O uso do algorítimo de previsão tem grande utilidade, além da facilidade de uso, permite-o acesso a combinações de padrões que não são visíveis individualmente

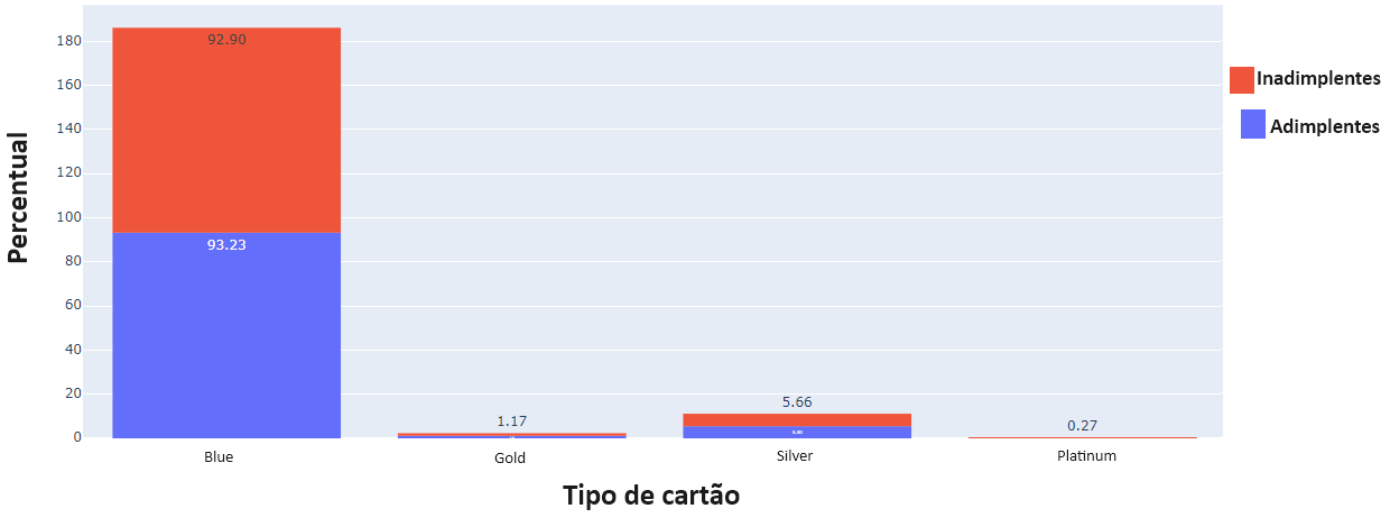

Figura 3.6: Distribuição do tipo de cartão de crédito por categoria de inadimplência.

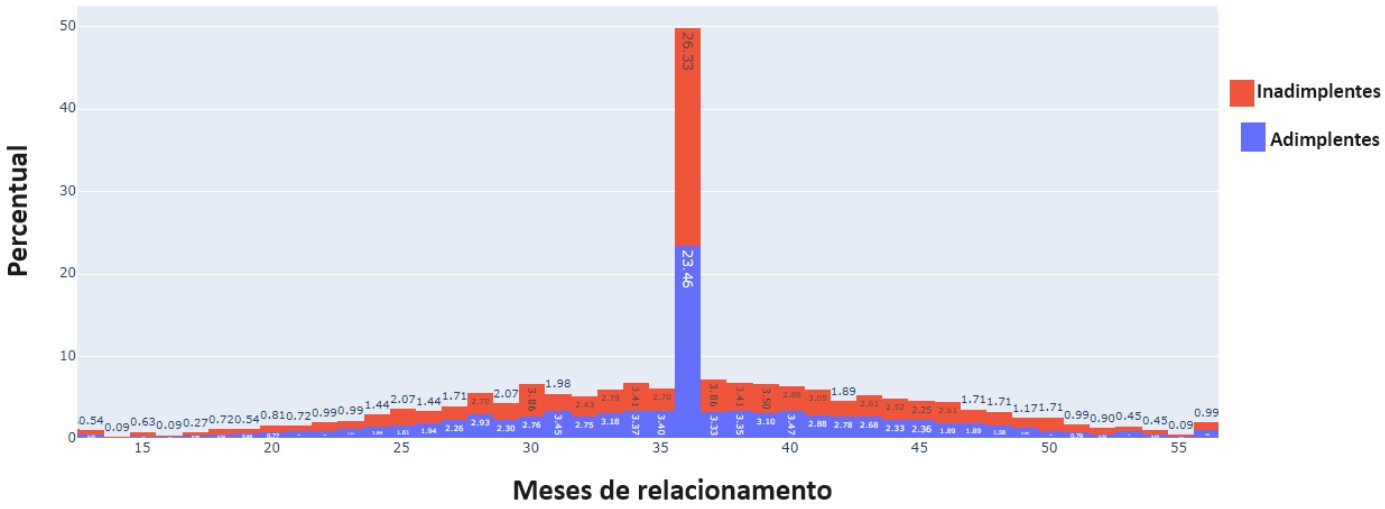

Figura 3.7: Distribuição do número de meses de relacionamento por categoria de inadimplência.

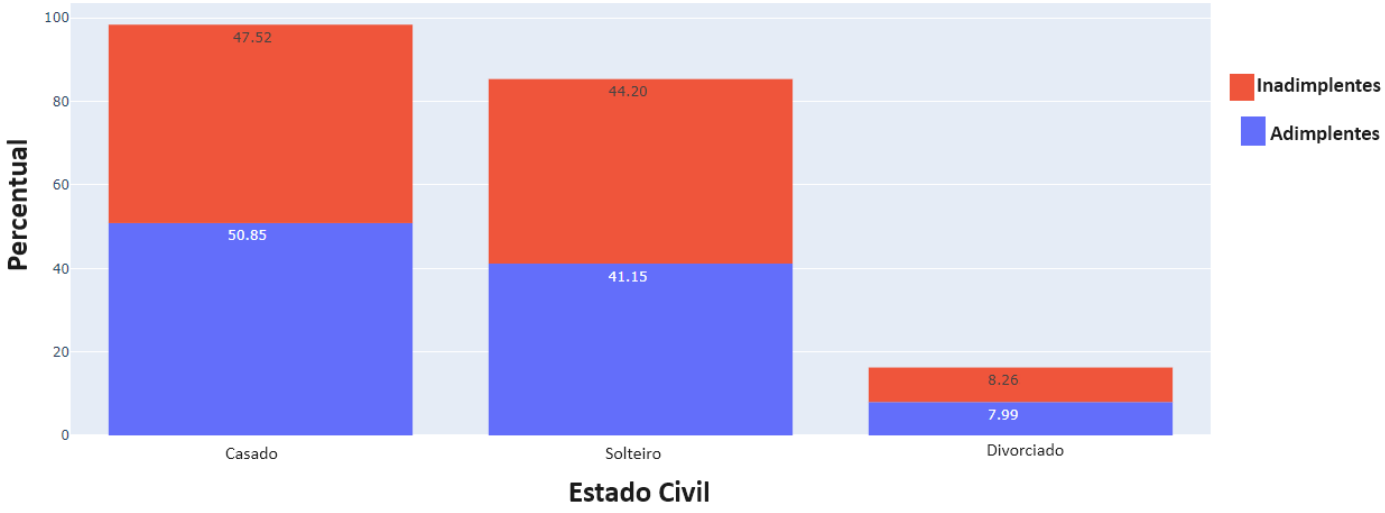

Figura 3.8: Distribuição do status civil por categoria de inadimplência.

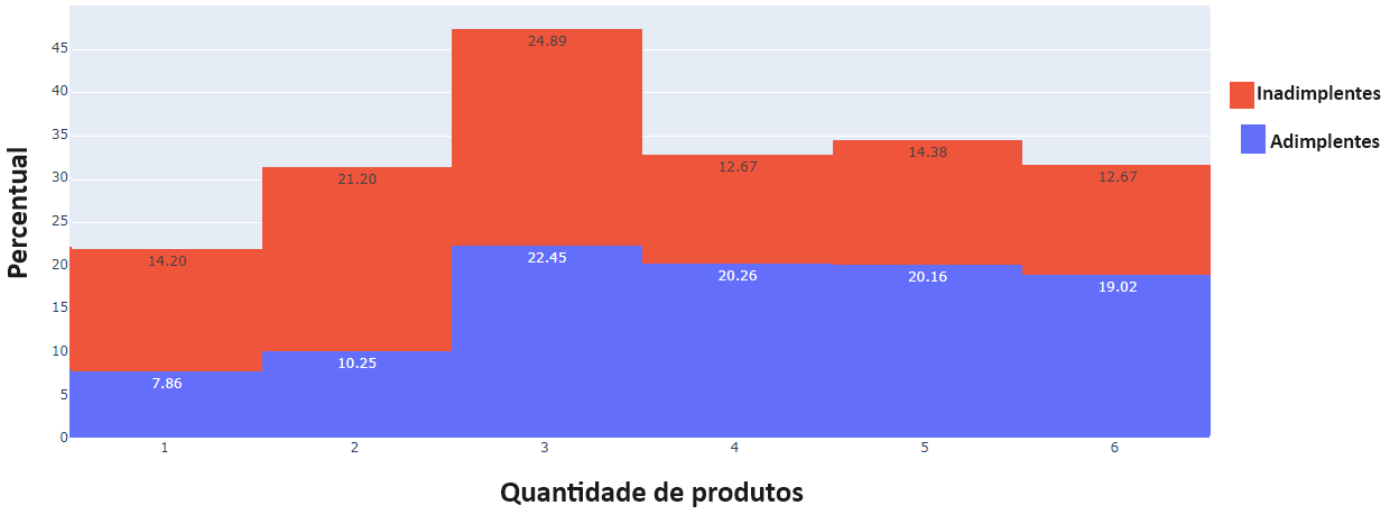

Figura 3.9: Distribuição da quantidade de produtos contratados pelo cliente por categoria de inadimplência.

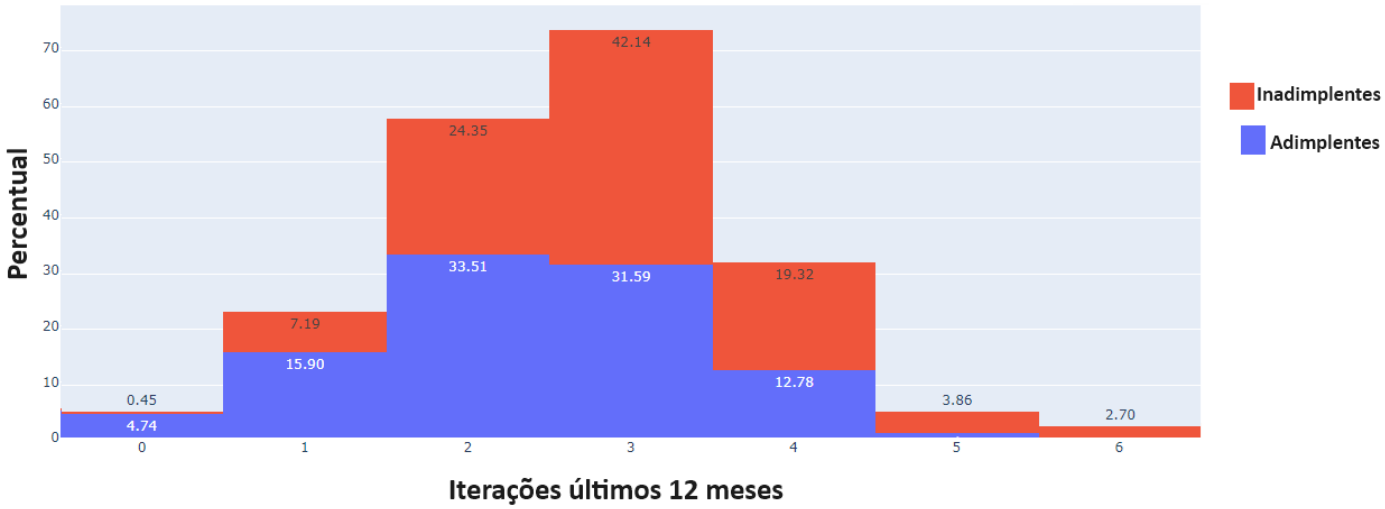

Figura 3.10: Distribuição de iterações nos últimos 12 meses por categoria de inadimplência.

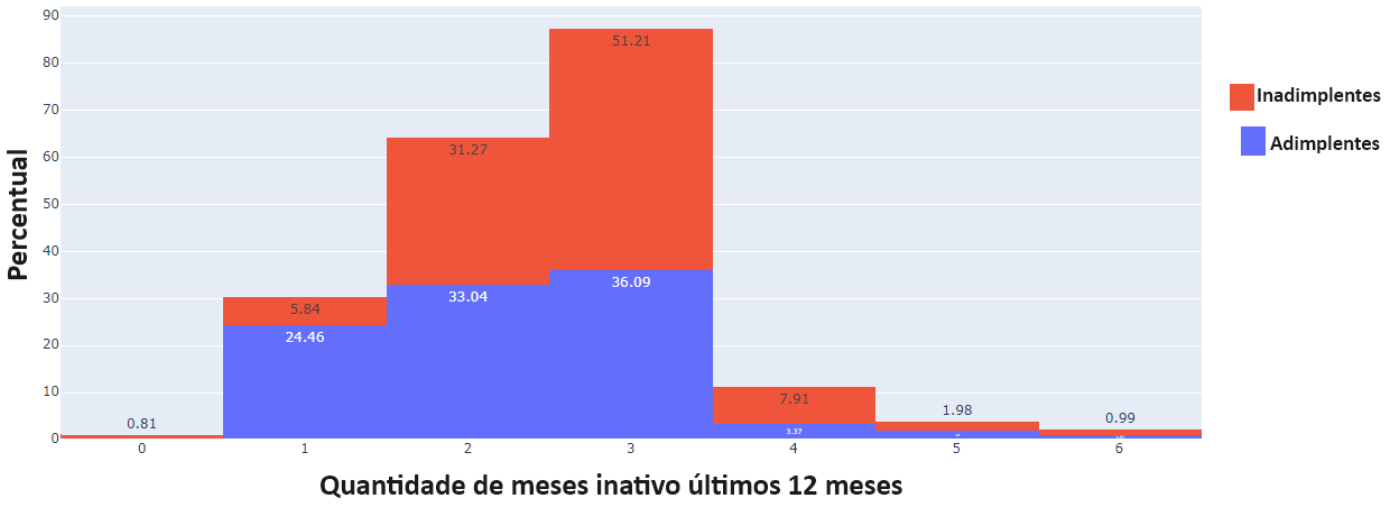

Figura 3.11: Distribuição do número de meses inativo por categoria de inadimplência.

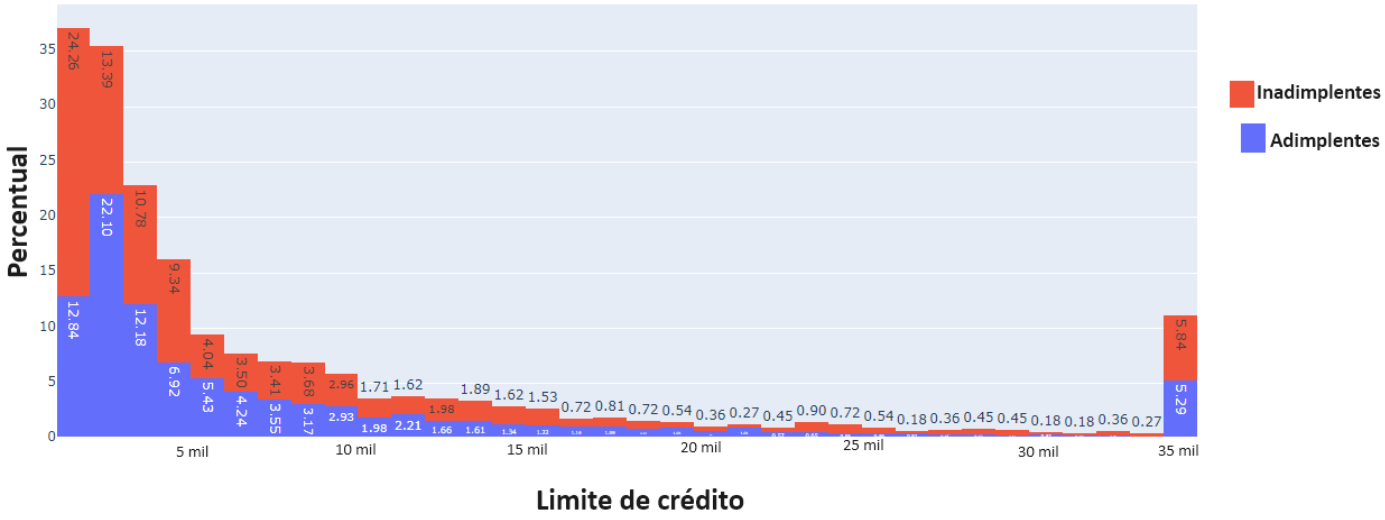

Figura 3.12: Distribuição do limite de crédito do cliente por categoria de inadimplência.

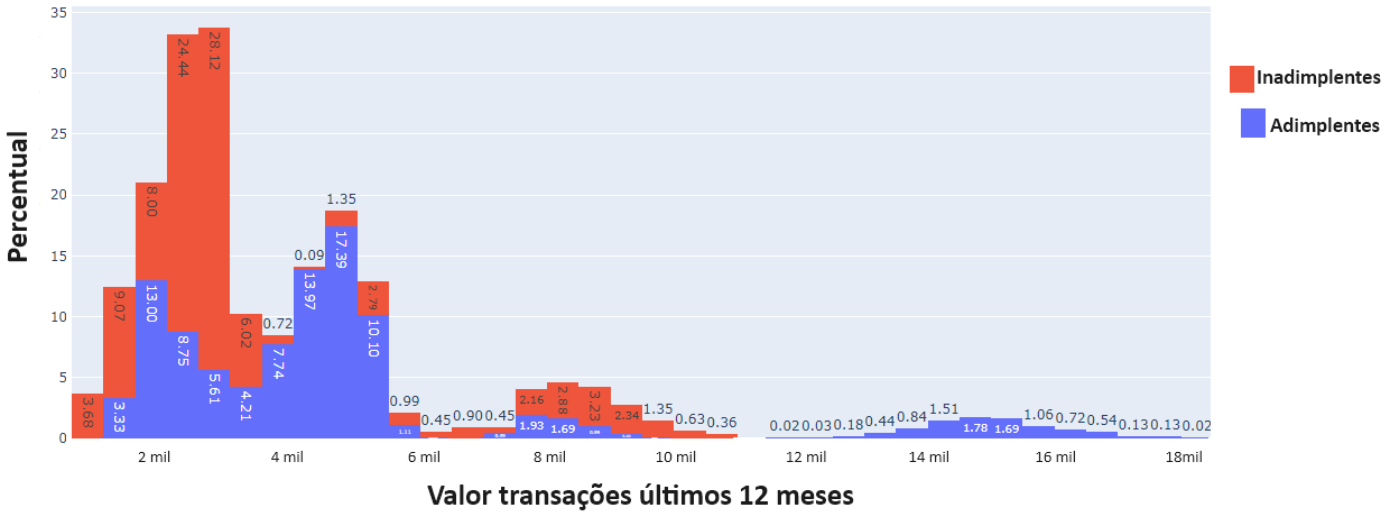

Figura 3.13: Distribuição dos valores das últimas transações por categoria de inadimplência.

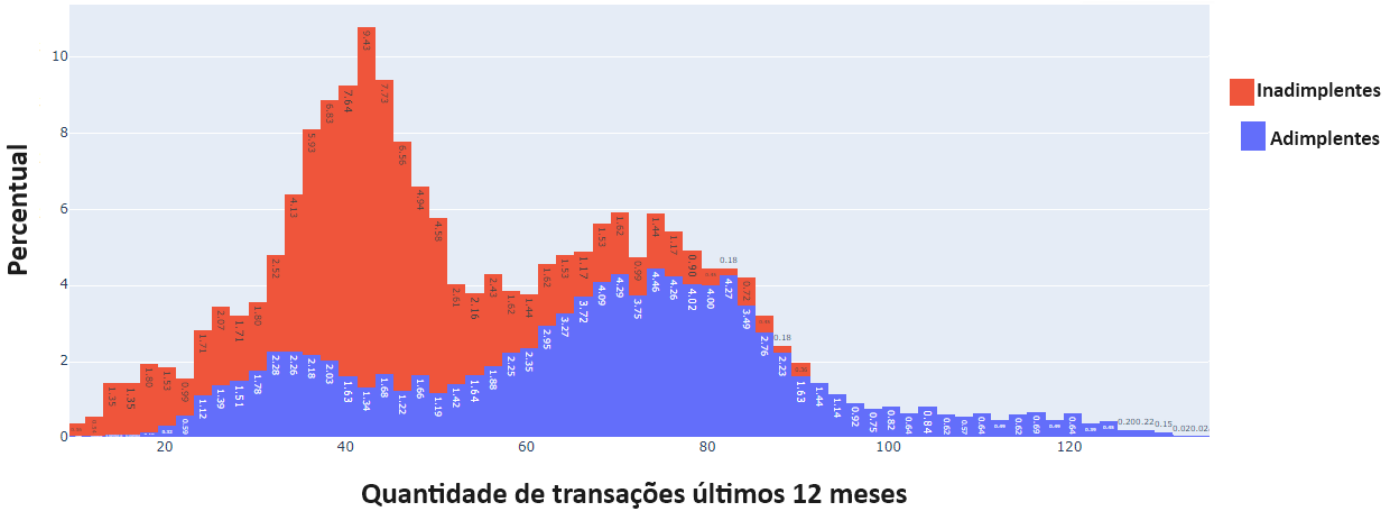

Figura 3.14: Distribuição da quantidade de transações por categoria de inadimplência.

3.3 ANÁLISES

Para facilidade de identificação e de apresentação dos resultados, as diferentes análises são enumeradas como na Tabela 3.1.

Tabela 3.1: Identificação e enumeração da análises realizadas.

Algoritmo	Normalização	Balanceamento	Análise
Regressão Logística	Sem	Sem	1
		SMOT	2
		ADASYN	3
	Com	Sem	4
		SMOT	5
		ADASYN	6
Árvore de Decisão	Sem	Sem	7
		SMOT	8
		ADASYN	9
	Com	Sem	10
		SMOT	11
		ADASYN	12
Floresta Aleatória	Sem	Sem	13
		SMOT	14
		ADASYN	15
	Com	Sem	16
		SMOT	17
		ADASYN	18

3.4 RESULTADOS POR ALGORITMO

Nessa seção serão apresentados os resultados das análises nos diversos cenários resultantes das combinações de algorítimo (regressão logística, árvore de decisão e floresta aleatória), balanceamento (sem, SMOTE e ADASYN) e normalização (sem e com). Estes diversos cenários serão comparados afim de identificar o melhor modelo, por meio das métricas acurácia, recall e F1.

Diversos estudos têm investigado e comparado algoritmos de aprendizado de máquina de forma semelhante a este trabalho. Foi encontrado na literatura um estudo em que os autores faz uma análise abrangente das performances de 9 diferentes algoritmos, inclusive os 3 utilizados neste trabalho, destacando a importância da escolha do algoritmo apropriado para encontrar o melhor resultado[22]. No caso do trabalho citado, os autores fizeram validação cruzada e utilizaram como métrica a acurácia, a precisão, a revocação (recall ou sensibilidade) e o F1 score e portanto, com exceção da precisão, as mesmas métricas utilizadas neste trabalho. Por outro lado, não fizeram uso do balanceamento e nem da normalização dos dados, o que foi realizado nesta pesquisa. Observaram que o melhor algoritmo

foi o Ensemble Stack, seguido pelo XGBoost e em terceiro a Regressão Logística, todos avaliados pela acurácia média. Os dois primeiros são, portanto, sugeridos para trabalhos futuros sobre modelos de predição de inadimplência.

Na revisão da literatura para este trabalho, foi identificado outro estudo em que os autores não usaram os mesmos algoritmos, mas compararam modelos ditos tradicionais com modelos de aprendizado de máquina e concluíram pela melhor eficiência dos modelos com aprendizado de máquina[19].

Portanto, o tema abordado neste trabalho e extremamente atual e justifica-se pela eficiência dos modelos treinados a partir dos algoritmos de aprendizado de máquina os quais são perfeitamente possíveis haja vista a disponibilidade de grandes bancos de dados e da capacidade computacional nos dias atuais.

3.4.1 ALGORITMO REGRESSÃO LOGÍSTICA

É possível visualizar na Tabela 3.2 os valores das três métricas, obtidos a partir do algoritmo Regressão Logística, para o caso sem balanceamento e balanceamentos SMOT e ADASYN e dados não normalizados. Resultados obtidos a partir dos mesmos cenários, porém, com normalização dos dados, são apresentados na Tabela 3.3.

Tabela 3.2: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Regressão Logística, sem normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,873	0,380	0,482	0,817	0,829	0,828	0,821	0,851	0,813
2	0,877	0,441	0,454	0,824	0,839	0,808	0,805	0,845	0,828
3	0,871	0,424	0,508	0,830	0,833	0,841	0,801	0,831	0,833
4	0,888	0,361	0,507	0,840	0,817	0,824	0,823	0,839	0,797
5	0,899	0,42	0,528	0,820	0,848	0,808	0,796	0,831	0,824
6	0,864	0,292	0,479	0,830	0,846	0,820	0,820	0,817	0,821
7	0,874	0,392	0,567	0,803	0,828	0,831	0,816	0,861	0,822
8	0,862	0,355	0,455	0,823	0,892	0,814	0,809	0,856	0,807
9	0,879	0,333	0,477	0,821	0,856	0,828	0,814	0,854	0,820
10	0,871	0,392	0,467	0,824	0,825	0,827	0,819	0,842	0,814
Média	0,88	0,38	0,49	0,82	0,84	0,82	0,81	0,84	0,82
DP	0,011	0,045	0,035	0,010	0,021	0,011	0,009	0,014	0,011

Tabela 3.3: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Regressão Logística, com normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,882	0,424	0,502	0,794	0,802	0,811	0,781	0,770	0,797
2	0,901	0,284	0,553	0,798	0,793	0,790	0,777	0,783	0,775
3	0,853	0,463	0,369	0,804	0,810	0,788	0,779	0,814	0,790
4	0,874	0,347	0,470	0,792	0,829	0,785	0,808	0,780	0,785
5	0,862	0,269	0,558	0,798	0,805	0,806	0,800	0,803	0,834
6	0,838	0,411	0,409	0,775	0,795	0,782	0,795	0,792	0,780
7	0,868	0,361	0,528	0,776	0,782	0,797	0,796	0,846	0,791
8	0,878	0,324	0,512	0,799	0,781	0,798	0,787	0,810	0,798
9	0,888	0,329	0,47	0,773	0,779	0,793	0,780	0,808	0,755
10	0,884	0,39	0,443	0,776	0,773	0,780	0,780	0,771	0,773
Média	0,87	0,36	0,48	0,79	0,79	0,79	0,79	0,80	0,79
DP	0,018	0,062	0,062	0,012	0,017	0,010	0,011	0,023	0,021

A partir dos resultados dos modelos treinados a partir do algoritmo Regressão Logística, visualiza-se um melhores resultados para o caso de dados não normalizados (Tabela 3.2) quando comparados aos dados normalizados (Tabela 3.3).

No caso dos dados não normalizado observa-se a maior acurácia, em média, na situação sem balanceamento porém as métricas recall e F1 são as menores dentre os três tipos de balanceamento, também em média. Nos balanceamentos SMOT e ADASYN tanto a acurácia, quanto o recall e F1 média não diferem expressivamente.

Neste trabalho o critério para definir o melhor modelo será a superioridade na maioria das métricas. Sendo assim, no algoritmo Regressão Logística pode-se eleger como melhor modelo tanto o treinado com balanceamento SMOT quanto com ADASYN haja vista a similaridade dos valores das métricas nestes dois casos. No entanto, o algoritmo de balanceamento SMOT é relativamente mais simples o que sugere menor esforço computacional e, portanto, deve ser escolhido.

3.4.2 ALGORITMO ÁRVORE DE DECISÃO

Na Tabela 3.4 são apresentados valores das três métricas, obtidos a partir do algoritmo Árvore de Decisão assim como no caso do algoritmo de Regressão Logística, para o caso sem balanceamento e balanceamentos SMOT e ADASYN e dados não normalizados. Para o caso de dados normalizados, tem-se os resultados na Tabela 3.5.

Tabela 3.4: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Árvore de Decisão, sem normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,916	0,778	0,695	0,922	0,941	0,940	0,917	0,933	0,927
2	0,913	0,718	0,760	0,938	0,936	0,935	0,924	0,922	0,926
3	0,901	0,763	0,713	0,924	0,948	0,941	0,924	0,939	0,923
4	0,905	0,745	0,714	0,928	0,937	0,934	0,935	0,920	0,916
5	0,920	0,766	0,761	0,933	0,926	0,920	0,917	0,933	0,918
6	0,913	0,728	0,742	0,926	0,926	0,930	0,940	0,932	0,923
7	0,916	0,739	0,689	0,944	0,925	0,919	0,915	0,929	0,925
8	0,919	0,723	0,745	0,939	0,935	0,929	0,932	0,931	0,922
9	0,895	0,773	0,722	0,927	0,930	0,945	0,926	0,931	0,916
10	0,929	0,75	0,748	0,923	0,932	0,939	0,918	0,918	0,921
Média	0,91	0,75	0,73	0,93	0,93	0,93	0,92	0,93	0,92
DP	0,010	0,021	0,026	0,008	0,007	0,009	0,009	0,007	0,004

Tabela 3.5: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Árvore de Decisão, com normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,918	0,709	0,716	0,934	0,946	0,935	0,938	0,936	0,950
2	0,927	0,669	0,675	0,946	0,938	0,924	0,916	0,940	0,923
3	0,908	0,739	0,727	0,943	0,940	0,920	0,928	0,947	0,941
4	0,908	0,728	0,669	0,947	0,947	0,931	0,936	0,929	0,932
5	0,887	0,718	0,684	0,926	0,945	0,940	0,928	0,949	0,912
6	0,913	0,747	0,640	0,928	0,949	0,935	0,929	0,946	0,938
7	0,906	0,760	0,75	0,943	0,944	0,938	0,922	0,942	0,936
8	0,919	0,760	0,70	0,940	0,948	0,914	0,921	0,936	0,930
9	0,891	0,722	0,778	0,937	0,935	0,946	0,945	0,938	0,928
10	0,916	0,730	0,755	0,927	0,952	0,929	0,929	0,927	0,928
Média	0,91	0,73	0,71	0,94	0,94	0,93	0,93	0,94	0,93
DP	0,012	0,027	0,044	0,008	0,005	0,010	0,009	0,007	0,010

Os resultados dos modelos treinados a partir do algoritmo Árvore de Decisão, a situação em que os

dados não são normalizados (Tabela 3.4), obtém-se superioridade apenas em modelos sem balanceamento.

Para os dados normalizados, observa-se que quanto a acurácia, no balanceamento SMOT, tem-se melhor ajuste do que no balanceamento ADASYN e, por outro lado, quanto às métricas F1 e Recall os dois tipos de balanceamento são similares em média. Mas, com relação ao desvio padrão, o SMOT destaca-se por uma pequena margem.

Sendo assim, no algoritmo Árvore de Decisão o modelo escolhido é aquele obtido com normalização dos dados e balanceamento SMOT.

3.4.3 ALGORITMO FLORESTA ALEATÓRIA

É possível visualizar na Tabela 3.6 os resultados dos modelos treinados por meio do algoritmo Floresta aleatória com os dados não normalizado e, na Tabela 3.7, com os dados normalizados.

Tabela 3.6: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Floresta Aleatória, sem normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,939	0,595	0,786	0,973	0,971	0,958	0,966	0,965	0,959
2	0,920	0,737	0,792	0,960	0,960	0,961	0,966	0,969	0,964
3	0,935	0,664	0,789	0,954	0,953	0,963	0,965	0,972	0,964
4	0,942	0,737	0,745	0,971	0,976	0,970	0,955	0,958	0,969
5	0,943	0,714	0,804	0,972	0,976	0,963	0,957	0,973	0,968
6	0,949	0,669	0,779	0,956	0,971	0,951	0,973	0,959	0,966
7	0,942	0,692	0,753	0,966	0,974	0,971	0,967	0,972	0,954
8	0,936	0,676	0,729	0,972	0,972	0,962	0,965	0,975	0,967
9	0,932	0,675	0,775	0,961	0,969	0,975	0,964	0,969	0,967
10	0,942	0,740	0,756	0,968	0,959	0,974	0,972	0,969	0,972
Média	0,94	0,69	0,77	0,97	0,97	0,96	0,97	0,97	0,97
DP	0,008	0,045	0,024	0,007	0,008	0,008	0,006	0,006	0,005

Tabela 3.7: Métricas Acuracy (M1), Recall (M2) e F1 (M3), com média e desvio padrão (DP), em cada tipo de balanceamento, no algoritmo Floresta Aleatória, com normalização dos dados.

Fold	Balanceamento								
	Sem			SMOT			ADASYN		
	M1	M2	M3	M1	M2	M3	M1	M2	M3
1	0,929	0,677	0,758	0,958	0,972	0,971	0,963	0,970	0,962
2	0,942	0,669	0,803	0,957	0,960	0,963	0,969	0,965	0,968
3	0,944	0,946	0,741	0,964	0,961	0,966	0,963	0,968	0,968
4	0,943	0,708	0,840	0,967	0,966	0,962	0,963	0,971	0,965
5	0,933	0,695	0,783	0,957	0,968	0,965	0,959	0,961	0,962
6	0,933	0,737	0,763	0,972	0,968	0,955	0,968	0,962	0,960
7	0,943	0,625	0,721	0,956	0,963	0,965	0,959	0,963	0,961
8	0,935	0,725	0,780	0,969	0,954	0,965	0,955	0,958	0,959
9	0,939	0,641	0,751	0,961	0,965	0,965	0,962	0,960	0,970
10	0,932	0,632	0,755	0,977	0,947	0,960	0,963	0,946	0,968
Média	0,94	0,71	0,77	0,96	0,96	0,96	0,96	0,96	0,96
DP	0,006	0,093	0,034	0,007	0,007	0,004	0,004	0,007	0,004

No algoritmo Floresta Aleatória o modelo sem normalização dos dados foi superior por poucos décimos e portanto, uma superioridade não expressiva.

Quanto a acurácia e recall, no caso de dados não normalizados (Tabela 3.6) os balanceamentos SMOT e ADASYN tem o mesmo valor. Porém quanto ao desvio padrão no ADASYN obtém-se valores ligeiramente menores. Portanto o melhor modelo no algoritmo Floresta Aleatória foi obtido utilizando o algoritmo de balanceamento ADASYN sem normalização de dados. Apesar desta escolha é importante ressaltar que os dois tipos de balanceamentos não diferem expressivamente.

3.5 COMPARAÇÃO DAS DIFERENTES MODELAGENS

O melhor cenário em cada algoritmo de treinamento foi determinado a partir das análises apresentadas nas Seção 3.4. No entanto, a comparação entre algoritmos de treinamento não foi realizada. Nesta Seção os resultados são apresentados em gráficos de maneira a facilitar a comparação dos dezoito cenários quanto a cada métrica.

O valor médio ou mediano bem como a homogeneidade da métrica na validação cruzada também pode fornecer indícios sobre a qualidade do modelo treinado. Um valor de métrica atípico, no treinamento sem a validação cruzada, pode indicar que para aquela situação o modelo treinado foi muito bom ou muito ruim, mesmo que esta não seja a situação real, portanto, um resultado ao acaso. Por outro lado, na validação cruzada, dado um valor atípico, ainda restam $k-1$ valores da métricas para avaliação. Sendo assim, um resumo dos k valores da métrica se torna interessante pois permite avaliar a melhor situação ou treinamento dentre as dezoito análises, como apresentado na Tabela 3.1, quanto

ao valor médio dos 10 ($k = 10$) valores da métrica adotada, e, além disso, quanto a homogeneidade dos valores obtidos, o que indicaria fidedignidade dos modelos treinados.

Estes resultados além de apresentados nas Tabelas 3.2, 3.3, 3.4, 3.5, 3.6 e 3.7, podem ser visualizados na Figuras 3.15, 3.16 e 3.17.

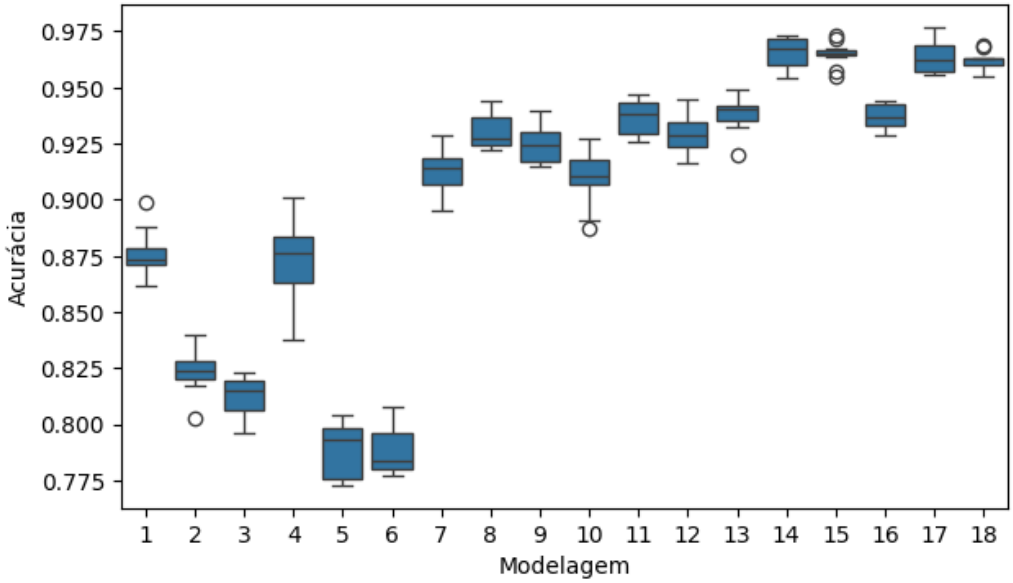

Figura 3.15: Box plot para a métrica acurácia por modelagem, sendo de 1 a 6 referente a regressão logística, de 7 a 12 referente a árvore de decisão e de 13 a 18 referente a floresta aleatória.

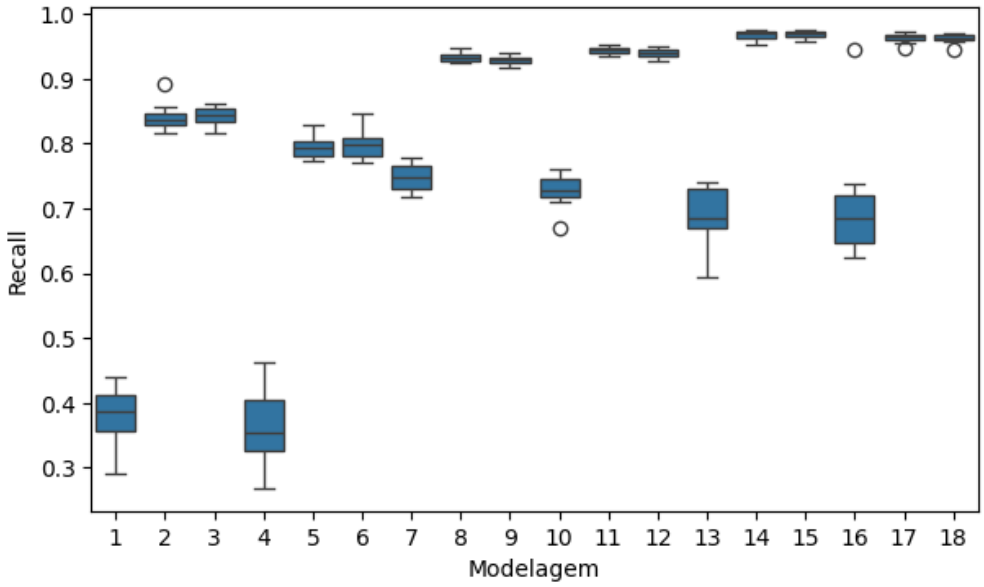

Figura 3.16: Box plot para a métrica recall por modelagem, sendo de 1 a 6 referente a regressão logística, de 7 a 12 referente a árvore de decisão e de 13 a 18 referente a floresta aleatória.

Observa-se através dos resultados visuais dos box plots que em relação aos treinamentos com normalização dos dados não foi possível observar grande efeito ou mudanças. No entanto, pode ser observada grande disparidade entre a Regressão Logística, como a menos eficiente, e os outros dois.

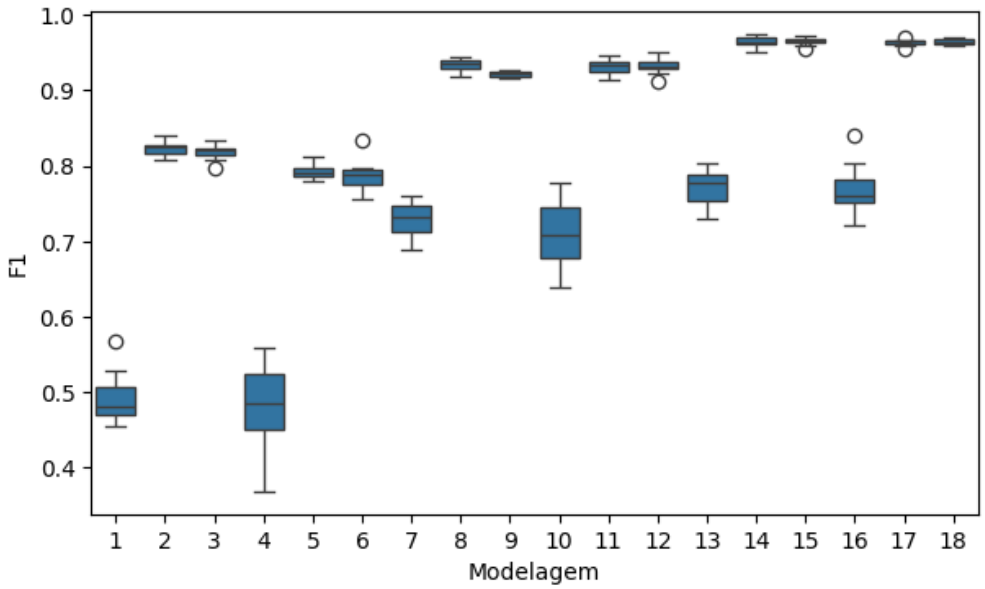

Figura 3.17: Box plot para a métrica F1 por modelagem, sendo de 1 a 6 referente a regressão logística, de 7 a 12 referente a árvore de decisão e de 13 a 18 referente a floresta aleatória.

A Árvore de Decisão e a Floresta Aleatória são similares com o algoritmo Floresta Aleatória ligeiramente superior.

Por fim, observa-se que o fator mais importante foi o balanceamento, que eleva consideravelmente os valores das métricas.

Conclui-se que os algorítimos Árvore de Decisão e Floresta Aleatória apresentam os melhores resultados. Eles tem suas diferenças porém pouco expressivas com ligeira superioridade, quanto à métrica acurácia (Figura 3.15), o Floresta Aleatória sem normalização de dados e com balanceamento ADASYN.

Embora um modelo tenha demonstrado uma leve vantagem sobre o outro, ambos exibiram resultados próximos. Dada a qualidade de ambos, qualquer um deles poderia ser selecionado com confiança para a previsão de inadimplência.

3.6 EXEMPLIFICAÇÃO DO USO DO MODELO

Classificando o modelo por sua melhores métricas, concluímos que o escolhido foi o floresta aleatória sem normalização de dados e com balancemento de ADASYN, dessa forma em termos usuais a instituição financeira poderá utilizar o modelo de uma maneira bem simples, afim de todos os analistas conseguirem chegar na predição do cliente selecionado.

Através de um formulário o analista responsável que está avaliando, irá preencher com as características que o cliente tiver, ao finalizar o preenchimento do mesmo, o analista irá simplesmente apertar um botão executador do modelo que lhe trará uma mensagem informando uma variável 0 para clientes adimplentes e com 1 para não adimplentes.

Simulando uma ação, supomos um cliente com 51 anos, sexo masculino , com 3 dependentes, uma faixa salarial anual de 80 mil a 120 mil, com uma escolaridade de mestrado, usando o cartão tipo blue,

com 36 meses de relacionamento, tendo o status civil de casado, com uma quantidade de produtos de 4, com nenhuma iteração nos últimos 12 meses, tendo 1 mês de inatividade, com um limite de crédito de 3.418,56, com um valor das últimas transações de 1.887,72 e quantidade de transações de 20, ao apertar o botão tivemos uma resposta para esse cliente de 0, sendo assertivo no modelo escolhido para este caso.

Afim de ficar uma exemplificação mais completa, na Tabela 3.8 é possível observar todos os resultados para todos os modelos testados com a suposição feita. Nota-se que tivemos uma assertividade em dois dos três modelos.

Tabela 3.8: Resultados dos três modelos sem normalização com balanceamento ADASYN.

Regressão Logística	Árvore de Decisão	Floresta Aleatória
Inadimplente	Adimplente	Adimplente

4. CONCLUSÕES

Para concluir, o presente trabalho de pesquisa sobre a inadimplência em um banco revelou resultados altamente promissores em relação à capacidade de predição do modelo desenvolvido. Com uma impressionante acurácia de 96%, com o modelo de floresta aleatória sem normalização de dados e balanceamento ADASYN, fica evidente que o modelo desempenha um papel fundamental na identificação e mitigação dos riscos de inadimplência.

Os resultados alcançados destacam a importância das técnicas de aprendizagem de máquina, com destaque, para a aplicação da floresta aleatória, na análise de risco de crédito.

Além disso, o trabalho revelou que mesmo com a escolha do algorítimo floresta aleatória, o algoritmo árvore de decisão teve um resultado muito próximo do escolhido, portanto, dada a qualidade dos dois, qualquer um deles poderia ser selecionado com confiança para o caso em questão.

No entanto, é importante mencionar que, apesar do alto desempenho do modelo, existem limitações específicas à análise de dados e à previsão de eventos financeiros. Consequentemente, a manutenção e o aprimoramento contínuo do modelo são essenciais para acompanhar as mudanças nas condições comportamentais dos clientes.

Em suma, este trabalho demonstrou a eficácia de abordagens baseadas no aprendizado de máquina na previsão de inadimplência em uma instituição financeira. Os resultados alcançados são promissores e podem servir como base para aprimorar as práticas de gerenciamento de risco e tentar aumentar a estabilidade financeira da instituição. À medida que o setor financeiro continua a evoluir, a pesquisa e o desenvolvimento contínuo de modelos preditivos são essenciais para enfrentar os desafios em constante mudança do mercado.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] *Curso de Machine Learning*. https://bookdown.org/jessicakubrusly/cursodemachinelearning/book/04arvores de decisao.html.

[2] Alemar, B.: *Técnicas para Dados Desbalanceados (SMOTE e ADASYN)*. https://medium.com/@balemar/t2023.

[3] Breiman, L.: *Random forests*. Machine learning, 45, 2001.

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O. e Kegelmeyer, W. P.: *SMOTE: synthetic minority oversampling technique*. Journal of artificial intelligence research, 16:321–357, 2002.

[5] Corsini, I. e Araújo, T.: *Número de inadimplentes no Brasil atinge recorde da série histórica, aponta Serasa*. CNN Brasil. Disponível em: https://www.cnnbrasil.com.br/business/numero-de-inadimplentes-no-brasil-atinge-recorde-da-serie-historica-apo Acessado em: 18/11/2022, 2022.

[6] Daiany Francisca Lara, Aurora Trinidad Ramirez Pozo, L. M. L. d. S. G. C. A. P. F. A.: *Estudos Empíricos dos Métodos de Balanceamento para a Classificação*. UNEMAT,UFPR,FAPAN.

[7] DINIZ, L.: *ESTUDO DO JOGO DE PÔQUER UTILIZANDO MÉTODOS ESTATÍSTICOS E APRENDIZADO DE MÁQUINA*. Universidade Federal de Uberlândia, 2022.

[8] Fontana, : *Introdução aos algoritmos de aprendizagem supervisionada*. Departamento de Engenharia Química, Universidade Federal do Paraná, 2020.

[9] G1: *Inadimplência tem o maior índice em 12 anos; endividamento atinge 76,6% das famílias*. Disponível em: https://g1.globo.com/economia/noticia/2022/03/03/inadimplencia-tem-o-maior-indice-em-12-anos-endividamento-atinge-766per ghtml, 2022.

[10] Gareth James, D. W.: *An Introduction to Statistical Learning with Applications in R*. 2ª ed., 2014.

[11] GERENCIANET: *Inadimplência: veja tudo o que você precisa saber sobre o assunto*. Disponível em: https://gerencianet.com.br/blog/tudo-sobre-inadimplencia/, 2022.

[12] Gomes, P. C. T.: *Introdução ao aprendizado de máquina*. https://www.datageeks.com.br/aprendizado-de-maquina/, 2019.

[13] HAO, J.: *Solving the Problem of Unbalanced Dataset in Machine Learning*. https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28, 2020.

[14] He, H., Bai, Y., Garcia, E. A. e Li, S.: *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*. Em *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pp. 1322–1328. Ieee, 2008.

[15] ICHI.PRO: *O que é validação cruzada? Como funciona?* Disponível em: https://ichi.pro/pt/o-que-e-validacao-cruzada-como-funciona-22627888732644, 2020.

[16] Maione, C.: *Balanceamento de dados com base em oversampling em dados transformados*. Tese de Doutoramento, Universidade Federal de Goiás, 2020.

[17] Niklas Bussmann, Paolo Giudici, D. M. e Papenbrock, J.: *Explainable Machine Learning in Credit Risk Management*. Computational Economics, 2021.

[18] Pereira, L.: *Índice de Gini: o que é e como é calculado? Gini do Brasil e do mundo*. https://www.dicionariofinanceiro.com/indice-de-gini/.

[19] Pinto, A. C.: *O pode preditivo dos modelos de aprendizado de máquina é superior aos modelos tradicionais para ánalise do risco de crédito?* IDP, 1, 2021.

[20] Rafael Izbicki, T. M. d. S.: *Aprendizado de máquina: uma abordagem estatística*, vol. 1. 1ª ed., 2020.

[21] REIS, E.: *Estatistica descritiva*. Lisboa Silabo, 4(1), 1998.

[22] Renato De Sant'anna Lopes, L. R. C. e. F. M.: *Comparação de Algoritmos de Aprendizado de Máquina para Predição de Pontuação de Crédito*. XIV Computer on the Beach, 2023.

[23] Rodrigues, V.: *Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças?* https://vitorborbarodrigues.medium.com/m2019.

[24] Santos, H. G. d., N. C. F. d. I. R. D. Y. A. d. O. C. F. P.: *Machine learning para análises preditivas em saúde: exemplo de aplicação para predizer óbito em idosos de São Paulo, Brasil*. Cadernos de Saúde Pública, 35:e00050818, 2019.

[25] Scudilio, J.: *Qual a melhor métrica para avaliar os modelos de Machine Learning?* https://www.flai.com.br/juscudilio/qual-a-melhor-metrica-para-avaliar-os-modelos-de-machine-learning/, 2020.

[26] Silva, D. L. da: *PRÉ-PROCESSAMENTO DE DADOS COM SKLEARN USANDO ESCALONADOR STANDARD E MINMAX*. https://acervolima.com/pre-processamento-de-dados-com-sklearn-usando-escalonador-standard-e-minmax/google$_vignette$.

[27] Van Rossum, G. e Drake Jr, F. L.: *The python language reference*. Python software foundation, 2014.

[28] Zeinab Hassani, M. A. M. e Hajihashemi, V.: *Credit Risk Assessment Using Learning Algorithms for Feature Selection*. FUZZY INFORMATION AND ENGINEERING, 12, 2020.

A. CÓDIGO PYTHON

```
# Importanadp as bibliotecas necessarias
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, make_scorer, accuracy_score,
 precision_score, recall_score, f1_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
from imblearn.combine import SMOTEENN
from imblearn.combine import SMOTETomek
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import classification_report
from sklearn import preprocessing
from pandas import read_excel
from openpyxl import Workbook
from sklearn.linear_model import LogisticRegression
from sklearn import tree
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from imblearn.over_sampling import ADASYN
from collections import Counter
```

```
# Importando os dados
df = pd.read_excel('datasetbancostone.xlsx')
df
df.rename(columns={'inadimplente/adimplente': 'inadimplente'}, inplace = True)
# 1 = inadimplete; 0 = adimplentes

```

```
#Porcentagem de inadimplencia
n=df['inadimplente'].count()
print((df.groupby('inadimplente').size()/n)*100)

# Analises exploratorias
for coluna in df.columns:
     grafico = px.histogram(df, x=coluna, histnorm='percent',color="inadimplente
    ", text_auto=True)
     grafico.show()

# Transformando variaveis categoricas em dummies
df2 = pd.get_dummies(df, columns=['sexo', 'escolaridade', 'estado_civil','
    salario_anual','tipo_cartao',], drop_first=True)
X=df2.drop(['inadimplente'],axis=1) #  previsores
y=df2['inadimplente'].values #  classe
index = pd.Index(y)
index.value_counts()/y.size

# Algoritimos
models = []
models.append(('LR', LogisticRegression(random_state=0)))
models.append(('Arv', DecisionTreeClassifier(random_state=0)))
models.append(('Flo', RandomForestClassifier(n_estimators = 100,random_state=0))
    )
# Aplicar o scoring de acordo com a metrica desejada: accuracy, recall, f1
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, shuffle = True)
    cv_results = cross_val_score(model, X, y, cv=kfold, scoring='accuracy')
    print(cv_results)
    results.append(cv_results)
    names.append(name)
    msg ="%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

# Normalizacao (se optar por normalizacao)
scaler = MinMaxScaler()
scaler.fit(X)
Xn = scaler.transform(X)

# aplicar o scoring de acordo com a metrica desejada: accuracy, recall, f1
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, shuffle = True)
    cv_results = cross_val_score(model, Xn, y, cv=kfold, scoring='accuracy')
    print(cv_results)
    results.append(cv_results)
```

```
    names.append(name)
    msg ="%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)


# Balanceamento SMOTE (opcional)
# transformacao do banco de dados
oversample = SMOTE()
X_sm, y_sm = oversample.fit_resample(Xn, y)
pd.Index(y_sm).value_counts()/y_sm.size
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, shuffle = True)
    cv_results = cross_val_score(model, Xn, y, cv=kfold, scoring='accuracy')
    print(cv_results)
    results.append(cv_results)
    names.append(name)
    msg ="%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)


# Balanceamento ADASYN (opcional)
# transformacao do banco de dados
oversample2 = ADASYN()
X_ad, y_ad = oversample2.fit_resample(Xn, y)
pd.Index(y_ad).value_counts()/y_ad.size
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, shuffle = True)
    cv_results = cross_val_score(model, Xn, y, cv=kfold, scoring='accuracy')
    print(cv_results)
    results.append(cv_results)
    names.append(name)
    msg ="%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```