Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Estatística

Modelo de Predição Para Dados Desbalanceados Utilizando Informações de Financiamentos de Veículos

Larissa Torres

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Estatística

Modelo de Predição Para Dados Desbalanceados Utilizando Informações de Financiamentos de Veículos

Larissa Torres
Orientadora: Profª Drª Maria Sílvia de Assis Moura

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Bacharel em Estatística.

São Carlos
Abril de 2023

Larissa Torres

Modelo de Predição Para Dados Desbalanceados Utilizando Informações de Financiamentos de Veículos

Este exemplar corresponde à redação final do trabalho de conclusão de curso devidamente corrigido e defendido por Larissa Torres e aprovado pela banca examinadora.

Aprovado em 28 de março de 2023

Banca Examinadora:

- Profª Drª Maria Sílvia de Assis Moura
- Profª Drª Andressa Cerqueira
- Profº Drº Danilo Lopes

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha querida mãe Valéria Greco Torres, que sempre esteve ao meu lado, incentivando e me dando força nos momentos mais difíceis. Sua presença foi fundamental para que eu pudesse chegar até aqui.

Agradeço também à minha irmã Amanda Torres, que me ajudou muito durante essa fase de estudos, compartilhando conhecimento e incentivando meu desenvolvimento acadêmico. Sua presença foi fundamental para compreender melhor os caminhos da vida a serem tomados, seu companheirismo sempre me proporcionou segurança para enfrentar os desafios da vida adulta.

Meus avós maternos, Valdir Torres e Irene Greco, já falecidos, sempre me inspiraram com sua sabedoria e amor incondicional. Sinto falta da presença deles, tenho certeza de que estão orgulhosos de mim, onde quer que estejam.

Não posso deixar de agradecer aos meus colegas de classe, Ana Beatriz Alves Monteiro, Natalia Vieira Paulino e Victor Alves Dogo Martins, por tornar os momentos desafiadores mais alegres e divertidos, sempre estiveram dispostos a ajudar, tirar dúvidas e compartilhar conhecimento. A troca de experiências e a amizade que construímos foram essenciais para que eu pudesse concluir este curso com sucesso.

Por fim, agradeço a todos os professores, tutores e funcionários da instituição de ensino, que sempre se dedicaram a nos ensinar e orientar. Seu empenho e comprometimento foram fundamentais para que pudéssemos adquirir o conhecimento necessário para atingir nossos objetivos.

Resumo

Para o trabalho de conclusão de curso em estatística, é proposto um ajuste de modelo para dados desbalançados, utilizando informações de financiamento de veículos, cuja a variável resposta é dicotômica, divida em adimplentes e inadimplentes. Será apresentado técnicas de seleção de variáveis, como peso da evidência e valor da informação, ajuste de modelo de regressão logística tanto com os dados desbalanceados e balanceados, métricas de qualidade do modelo e uma classificação final interpretável. O trabalho foi desenvolvido utilizando a linguagem de programação Python.

Palavras-chave: *ajuste de modelo, classificação, crédito, dados desbalanceados, financiamento de veículos, python, regressão logística.*

Abstract

For the statistics thesis project, a model adjustment is proposed for unbalanced data, using vehicle financing information, with the response variable being dichotomous, divided into defaulters and non-defaulters. Techniques for variable selection, such as weight of evidence and information value, will be presented, along with the adjustment of logistic regression models for both unbalanced and balanced data, model quality metrics, and an interpretable final classification. The project was developed using the Python programming language.

Keywords: *model adjustment, classification, credit, unbalanced data, vehicle financing, Python, logistic regression.*

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Crédito como garantia é uma forma de concessão de crédito em que se minimiza o prejuízo em caso de quebra de contrato. Esse tipo de crédito, quando bem estruturado, se torna um dos negócios mais rentáveis e lucrativos para instituições financeiras. Uma de suas modalidades é o financiamento de veículos, em que o automóvel é a garantia de retomada caso descumprimento de contrato.

Por outro lado, o mercado de veículos vem se tornando cada vez mais desafiador quando o assunto é conceder crédito pois, a pandemia afetou o cenário econômico do país trazendo instabilidades e, consequentemente, altas taxas de juros. Outro fator importante que prejudicou este segmento é a escassez de peças nos estoques das fábricas, sendo que o atraso na entrega de carros zero quilômetros fez com que modelos semi novos supervalorizassem, assim o consumidor ficou mais receoso para obter novos financiamentos (B3, 2022) (Votorantim, 2022).

Para que instituições financeiras funcionem de forma saudável e enfrente sazonalidades e variações atípicas, é necessário a construção de políticas de créditos sólidas, seguras e eficientes. Conhecimentos matemáticos, econômicos e estatísticos são essenciais para auxiliar a construção das regras.

Uma das metologias utilizadas é a classificação de probabilidade de não pagamento, popularmente conhecida como *rating*, cujo método consiste em combinações de variáveis que discriminam riscos e geram uma pontuação final para a chance de inadimplência. A classificação do proponente é um dos pilares da política, através dela são firmadas as condições do contrato, e também é possível identificar os bons e maus pagadores, isto é, conter riscos e oferecer condições mais rígidas para o grupo de alto risco e, em simultâneo, oferecer propostas mais permissíveis e não perder oportunidades de negócio

entre os melhores pagadores para a concorrência.

Este trabalho está estruturado da seguinte maneira. No capitulo 2 abordam-se as metodologias propostas. Para o ajuste do modelo será apresentada a regressão logística, enquanto na etapa de seleção de variáveis temos as técnicas de peso da evidência (WoE) e valor da informação (IV) e, em seguida, a árvore de classificação, sendo uma metodologia de aprendizado de máquinas; por fim, para a qualidade do modelo, são comentadas à estatística de Kolmogorov Smirnov, a curva ROC e a matriz de confusão. No capitulo 3 são apresentados os dados disponíveis para desenvolver as metodologias propostas. Os dados, contém variáveis relacionados a financiamento de veículo, sem seguida, será comentado previamente sobre os dados por meio de uma análise descritiva; as análises estão subdivididas em tópicos conforme o seu tema, sendo eles, informações de inadimplência, informações pessoais do proponente, informações do empréstimo e histórico financeiro. No capitulo 4 contém os resultados das metodologias propostas, as métricas de qualidade, e a apresentação final da classificação de risco do proponente para um financiamento de veículos. Por fim, o capitulo 5, apresenta a conclusão do trabalho.

1.1 Objetivo

O objetivo deste trabalho é ajustar um modelo de classificação com dados desbalanceados para a probabilidade de não pagamento de um financiamento de veículos. Com o auxílio de técnicas estatísticas, são selecionadas variáveis que mais ajudam a identificar riscos de crédito, em seguida é feito o ajuste com a metodologia de regressão logística, e por fim, é apresentada a classificação final obtida com o apoio da árvore de classificação. Também como objetivo, buscar um bom modelo, serão consideradas a parcimônia, as métricas de qualidade e, a nível em que o modelo separa as classes segmentadas com base nas características do grupo de proponentes presentes em função da variável resposta, a qual é responsável por reportar se ocorreu o pagamento ou não.

Capítulo 2

Metodologia

2.1 Regressão

Modelo de regressão é uma equação matemática, em que apresenta uma relação linear composta pela combinação de uma ou mais variáveis preditoras (x), do outro lado da igualdade temos uma única variável resposta (y), a variável resposta é aquela que está sendo explicada, enquanto a variável preditora, é aquela utilizada para explicar a variação na variável resposta, auxiliado a previsão comportamental dos dados.

A fim de ilustrar melhor, podemos observar uma equação de regressão estimada genérica com p variáveis explicativas, a seguir:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + ... + \hat{\beta}_p x_{ip} \quad i = 1, 2, \ldots, n \quad , \tag{2.1}$$

sendo que, o termo ($\hat{\beta}_0$) chamado de intercepto, é obtido com base no termo constante da equação e corresponde ao valor estimado de y_i quando os x_{ik}, $i = 1, 2, \ldots, n$ e $k = 1, 2, \ldots, p$ presentes na equação assumem valores zero. Já os demais coeficientes ($\hat{\beta}_1, ..., \hat{\beta}_p$) são as estimativas para ponderar cada covariável do modelo, por exemplo, coeficientes estimados com valores positivos significam que a variável específica pode agregar positivamente para o aumento médio da variável resposta, enquanto, coeficientes negativos apresentam a relação inversa, contribuindo negativamente para a variável resposta estimada, por fim, coeficientes estimados próximos a zero significam que a variável que o acompanha discrimina pouco para a informação da variável resposta.

2.1.1 Regressão Logística Simples

Regressão Logística é uma técnica estatística particular da Regressão Simples. É utilizada para atribuir probabilidade de sucesso a um evento, quando a variável resposta é dicotômica, composta por duas possibilidades, sucesso e fracasso.

Só existem duas possibilidades de resposta para mensurar o perfil do proponente, ou ele é adimplente, ou inadimplente, isto é, fracasso e sucesso, assim neste trabalho podemos modelar uma regressão logística utilizando uma variável aleatória Bernoulli.

$$\begin{cases} 1, & \text{caso o proponente é inadimplente ,} \\ 0, & \text{caso contrario .} \end{cases}$$

Paula (2013), descreve uma regressão logística de forma bem prática, considere inicialmente o modelo logístico linear simples em que $\pi(x)$, a probabilidade de "sucesso" dado o valor x de uma variável explicativa qualquer e definida tal que

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta x, \tag{2.2}$$

em que β_0 e β são parâmetros desconhecidos.

Para o contexto do trabalho a ser desenvolvido, pode-se ajustar um modelo, cujo objetivo é inferir a associação da inadimplência por um fator particular. Considerando uma variável preditora com apenas dois níveis, seriam então amostrados, independentemente, indivíduos com presença do fator $(x = 1)$ e indivíduos com ausência de um determinado fator $(x = 0)$ e $\pi(x)$ seria a probabilidade de não pagamento. Dessa forma, a chance do proponente inadimplir com a presença do fator fica dada por

$$\frac{\pi(1)}{1-\pi(1)} = e^{\beta_0+\beta}, \tag{2.3}$$

enquanto, a chance do proponente realizar o pagamento em dia com ausência do fator é simplesmente

$$\frac{\pi(0)}{1-\pi(0)} = e^{\beta_0}, \tag{2.4}$$

2.1.2 Regressão Logística Múltipla

Para casos em que se tem mais de uma variável preditora e, a variável resposta ainda é considerada dicotômica, se faz necessário o uso da Regressão Logística Múltipla. Esta me-

todologia estima conjuntamente todas as variáveis disponíveis e ainda considera possíveis interações, isto é, interagir duas ou mais variáveis entre si e representar na equação do modelo como apenas uma.

Por exemplo, seja x_1 o número de parcelas solicitadas pelo proponente para realizar o pagamento total acordado no contrato, e x_2 a idade do proponente no momento da contratação, o coeficiente acompanhado a x_1 será exclusivamente a estimação de inadimplência relacionado ao prazo de pagamento, enquanto β_2 será exclusivamente a estimação de risco de inadimplência para a idade, agora se é de interesse saber o risco de inadimplência considerando a idade do proponente junto ao prazo de pagamento, podemos utilizar a interação entre às duas variáveis, matematicamente dizendo, $x_3 = x_1 * x_2$.

A equação para representar o que foi descrito é dada por:

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3. \tag{2.5}$$

E por fim, uma equação genérica para um modelo de Regressão Logística Múltipla:

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p, \tag{2.6}$$

sendo p o número de variáveis e interações com interesse de serem estimadas.

Assim temos coeficientes ponderados exclusivamente para cada variável, ou seja, variáveis significativas tem coeficientes com valores mais relevantes, enquanto, variáveis pouco significativas para discriminar risco tem coeficientes próximos de zero.

Com o auxílio desta metodologia, é gerada uma equação cuja resposta é um único valor, ou seja, todas a variáveis x_p e interações estão resumidas neste valor, gerando uma métrica para estimar a inadimplência do proponente dado as suas características descritas através das variáveis preditoras.

2.2 Seleção de Variáveis

Para obter um modelo preciso, é importante considerar a parcimônia, por exemplo, caso implementado poucas variáveis no modelo estamos sujeitos a omitir variáveis importantes e o aumento do viés. Em contrapartida, ao adicionar muitas variáveis sem potencial de discriminação, isto é, pouco correlacionada com a variável resposta, ou de forte multicolinearidade, significando dois parâmetros para uma mesma informação, estamos propícios

a cometer erros pela necessidade da estimação de muitos parâmetros, consequentemente gerando uma função de regressão estimada com baixo poder de predição.

2.2.1 Peso da Evidência (WoE) e Valor da Informação (IV)

- **Peso da Evidência (WoE):**

Peso da Evidência (WoE) é a medida da força da associação entre uma variável preditora categorizada e uma variável resposta binária. Uma metodologia muito utilizada no crédito, visa medir o poder preditivo de uma variável explicativa em relação à variável resposta, sendo possível dizer o grau da separação de bons e maus clientes.

"Bons" refere-se aos clientes adimplentes (não-eventos) e "Maus" refere-se aos clientes inadimplentes a um empréstimo (eventos). Assim, "% Eventos" é a proporção de casos positivos (bons pagadores) em relação ao total de casos em uma categoria da variável independente, e "% Não-Eventos" é a proporção de casos negativos (maus pagadores) em relação ao total de casos na mesma categoria da variável independente.

Considerando que o evento é o proponente inadimplir, sendo identificados como "Maus" e, "Bons" é não evento sendo clientes com faturas pagas em dia. O WoE é calculado da seguinte maneira:

A porcentagem de bons é dada por,

$$\%\ \text{Bons} = \frac{\text{nº clientes adimplentes}}{\text{total de clientes}}, \tag{2.7}$$

enquanto a de maus corresponde a

$$\%\ \text{Maus} = \frac{\text{nº clientes inadimplentes}}{\text{total de clientes}}. \tag{2.8}$$

Considere uma variável explicativa com k níveis, e seja $i = 1, 2, ..., k$, o valor de WoE para cada nível da variável explicativa é dado por,

$$WoE_i = log\left(\frac{\%\ \text{Bons}_i}{\%\ \text{Maus}_i}\right) \quad i = 1, 2, , ..., n. \tag{2.9}$$

Para interpretações, WoE positivo implica maior grau de associação com o fator de adimplência presente na variável resposta, e WoE negativo o oposto.

Para as variáveis contínuas, o próprio algoritmo as categoriza por decil, isto é, dividir o conjunto de dados em 10 partes iguais, cada parte representando 10% dos dados. Isso é

feito ordenando os dados em ordem crescente ou decrescente e dividindo-os em 10 grupos de tamanho igual.

- **Exemplo do Uso do WoE nos Dados:**

Suponha que desejamos inferir o peso da informação para a variável Prazo que representa o tempo que o cliente deseja pagar o financiamento solicitado, categorizada pelo algoritmo em "36x", "48x" e "60x".

Suponha que os resultados sejam os seguintes:

Tabela 2.1: Exemplo do Uso do WoE nos Dados

Categoria	% Bons	% Maus	Calculo	WoE
36x	10%	2%	ln(10%/2%)	1,609
48x	8%	6%	ln(8%/6%)	0,287
60x	5%	15%	ln(5%/15%)	-1,099

Isso significa que a categoria "60x" tem a associação mais forte com a variável de inadimplência, seguida pela categoria "48x" e, em seguida, a categoria "36x" conforme os cálculos obtidos via WoE.

No entanto, é importante lembrar que o WoE é apenas uma medida da força da associação entre as variáveis e não é uma medida direta da probabilidade de inadimplência em si. Para determinar a probabilidade real de inadimplência, é necessário usar um modelo preditivo mais complexo que considere outras variáveis relevantes, como as condições do financiamento, histórico financeiro, dívidas ativas, entre outras.

- **Valor da Informação (IV):**

Com o auxílio do WoE podemos obter o valor da informação (IV). O valor da informação é uma das técnicas mais úteis para selecionar variáveis importantes em um modelo preditivo, que fornece uma medida de como a variável preditora é boa na distinção entre uma variável resposta binária, contribuindo para classificar as variáveis com base em sua importância.

O IV de cada variável preditora é calculado utilizando a seguinte fórmula:

$$IV = \sum_{i=1}^{k} (\%Bons_i - \%Maus_i) \times WoE_i, \tag{2.10}$$

Sendo que k é o número de categorias presentes ou criadas por decil para cada covariável. Em outras palavras, a diferença de bons e maus em cada categoria multiplicado ao valor do WoE.

Para a interpretação do IV, podemos considerar a seguinte tabela:

Tabela 2.2: Interpretação do IV Rocheny (2020)

Valor da informação	Previsibilidade da variável
Menos de 0,02	Não é útil para previsão
0,02 a 0,1	Poder preditivo fraco
0,1 a 0,3	Potência preditiva média
0,3 a 0,5	Forte poder preditivo
>0,5	Poder preditivo suspeito

2.3 Árvore de Classificação

A Árvore de Classificação é um método de classificação não-paramétrico muito popular nas técnicas de Aprendizado e Máquinas por ser simples e de fácil interpretação.

O algoritmo realiza divisões no espaço das covariáveis $(R_1, ..., R_j)$ de forma que todas as regiões sejam distintas e disjuntas, isto é, classificar a informação das variáveis preditoras visando explicar melhor a variável resposta, Izbicki e dos Santos (2020) explica teoricamente a predição para a variável resposta pela equação:

$$g(x) = \frac{1}{|i : x_i \in R_k|} \sum_{i:x_i \in R_k} y_i, \tag{2.11}$$

sendo que $g(x)$ é a média das respostas (y) correspondentes as covariáveis separadas em cada subespaço. Assim, é obtida a predição naquele grupo.

Essas regiões, são definidas por etapas: a priori é feita a quebra das covariáveis x de forma que as variáveis y na subdivisão sejam totalmente homogêneas, no caso deste trabalho, por exemplo, localizar pontos de cortes nas variáveis onde se dividem totalmente os maus pagadores dos bons pagadores.

As árvores são apresentadas em um padrão de estrutura:

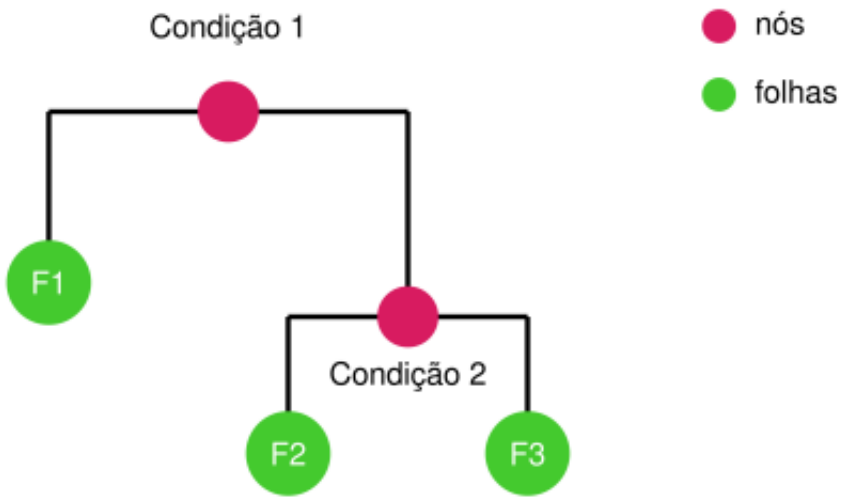

Figura 2.1: Exemplo de estrutura de uma árvore de regressão. Fonte: Izbicki e dos Santos (2020)

Os nós são as condições, onde a variável escolhida pelo algorítimo é representada junto da sua quebra ótima. Já as folhas apresentam a quantidade de observações presentes nesta quebra ótima e a média da variável resposta do grupo.

Com o auxílio da Árvore, podemos identificar padrões semelhantes dos clientes, classificá-los em grupos, e assim prever a chance de inadimplência dessas subpopulações. Para este trabalho, a árvore de classificação será utilizada para obter quebras ótimas do *score* final obtido.

2.4 Métricas de Qualidade do Modelo

Existem diversas metodologias para descrever dados e obter padrões de respostas, e para saber qual se destaca podemos mensurar o poder dos modelos ajustados. Basicamente aqui é mensurado o quão distante os valores ajustados estão dos valores observados, e assim temos a qualidade do modelo proposto.

2.4.1 Estatística Kolmogorov–Smirnov (KS)

Baseado no teste não-paramétrico de Kolmogorov-Smirnov, que consiste em avaliar se as amostras presentes são originadas de uma mesma distribuição. Neste trabalho, trata-se

de inferir se a população de inadimplente e adimplente provem da mesma populações ou não. O teste auxilia em compreender o quão bom o modelo é no quesito de separar os dois grupos de proponentes,

$$\hat{F_B}(s) = \frac{\text{nº clientes adimplentes com score} \leq s}{\text{total de clientes adimplentes}}, \tag{2.12}$$

$$\hat{F_M}(s) = \frac{\text{nº clientes inadimplentes com score} \leq s}{\text{total de clientes inadimplentes}}, \tag{2.13}$$

onde s é a quebra do score no intervalo

$$KS = max(|F_M(s) - F_B(s)|). \tag{2.14}$$

PICININI (2003) diz que, no mercado de crédito, é considerado que um bom modelo são aqueles que atingem mais de 30% de KS.

2.4.2 Curva ROC

A curva ROC é a sensibilidade em comparação à especificidade calculada em relação aos valores preditos, sendo a sensibilidade a capacidade de o teste detectar corretamente resultados positivos, neste caso identificação dos inadimplentes, e a especificidade, é a capacidade de o teste detectar corretamente resultados negativos, os adimplentes. Ou seja, são proporções de classificações corretas do modelo.

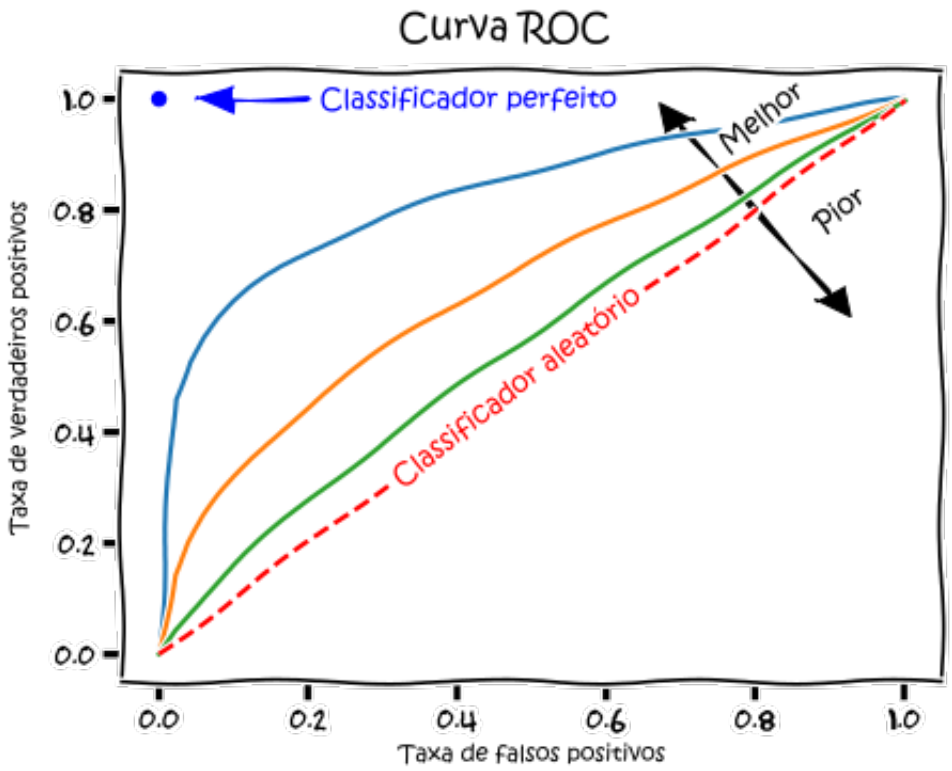

Figura 2.2: Ilustração da curva ROC. Fonte: MartinThoma (2021)

2.4.3 Matriz de Confusão

A Matriz de Confusão é uma matriz 2×2 contendo as frequências de observações enquadradas em cada uma das classificações descritas abaixo:

- **Verdadeiro negativo (VN):** ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente. Exemplo: o proponente é inadimplente, e o modelo previu corretamente que ele é inadimplente;
- **Falso positivo (FP):** ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente. Exemplo: o proponente é adimplente, mas o modelo classifica erroneamente como inadimplente;
- **Falso negativo (FN):** ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente. Por exemplo, o proponente é inadimplente e incorretamente o modelo previu que ele é adimplente;
- **Verdadeiro positivo (VP):** ocorre quando no conjunto real, a classe que estamos buscando foi prevista corretamente. Por exemplo, quando o proponente é adimplente e o modelo previu corretamente que ele é adimplente;

Tabela 2.3: Ilustração Matriz de Confusão

Valor do conjunto real	Valor predito	
	Y = 0	Y = 1
Y = 0	VP	FP
Y = 1	FN	VN

A partir das classes da tabela 2.3, é possível gerar novas métricas de qualidade:

- **Sensibilidade ou *recall*** $S = \frac{VP}{(VP+FN)}$ qual a proporção de positivos foi identificado corretamente. O quanto o modelo consegue prever corretamente os clientes adimplentes.

- **Especifidade** $E = \frac{VN}{(VP+FN)}$ dos clientes inadimplentes, qual a porcentagem de terem sidos classificados corretamente;

- **Precisão** $P = \frac{VP}{(VP+FP)}$ dos clientes classificados como adimplentes, quantos foram corretamente identificados;

- ***F1-score*** $F1 = \frac{2 \times P \times S}{P+S}$ uma maneira de observar em um único número a precisão e a sensibilidade, a fim de trazer um número único que determine a qualidade geral do nosso modelo;

- **Acurácia** $AC = \frac{VN+VP}{VN+FN+FP+VP}$ métrica resumo para identificar o quando o modelo previu corretamente.

2.5 Desbalanceamento dos Dados

O desbalanceamento dos dados é um problema para técnicas de aprendizado de máquinas, que ocorre quando o volume de observações nas classes não são proporcionalmente iguais, e assim, o algoritmo pode encontrar problemas para prever a classe minoritária, quando o algoritmo não tem dados o suficiente para aprender os padrões presentes na classe minoritária, causando uma predição tendenciosa para a classe majoritária, consequentemente sendo mais suscetível ao erro. Este problema é muito comum no cenário de crédito, geralmente em uma instituição financeira saudável não teremos a mesma proporção de adimplentes e inadimplentes. (Singh, 2020)

Algumas técnicas de balanceamento de dados:

- Under-sampling: nesta técnica, algumas observações da classe majoritária são removidos aleatoriamente para equilibrar o número de exemplos em cada classe.

- Over-sampling: nesta técnica, observações da classe minoritária são replicados ou gerados artificialmente. Por exemplo, replicar observações com base na média das covariáveis cuja a variável resposta corresponde ao fator minoritário, para aumentar o número de exemplos nessa classe.

- Ensemble: essa técnica combina várias abordagens de aprendizado de máquina para criar um modelo mais robusto e preciso. Por exemplo, pode-se treinar diferentes modelos em diferentes subconjuntos dos dados e, em seguida, combinar suas previsões.

- Class weights: nesta técnica, pesos diferentes são atribuídos às classes durante o treinamento do modelo. Isso dá mais importância à classe minoritária, fazendo com que o modelo preste mais atenção a ela durante o treinamento.

2.5.1 Técnica Class Weights Para Balanceamento de Dados

A técnica abordada neste trabalho para contornar o problema é a de Class weights, também conhecida como reponderação de classes, isto é, aplicar pesos maiores nas classes minoritárias modificando o algorítimo de treino, com o propósito de penalizar o erro de classificação feito pela classe minoritária, estabelecendo um peso de classe mais alto e, ao mesmo tempo, reduzindo o peso para a classe majoritária. Isso ajuda a garantir que o modelo aprenda a distinguir corretamente entre as classes, mesmo que haja uma desproporção significativa entre elas.

Uma vantagem da técnica de reponderação de classes é que ela é fácil de implementar e pode ser usada com uma variedade de algoritmos de aprendizado de máquina. No entanto, ela pode não ser eficaz em casos extremos de desequilíbrio de classe, onde a classe minoritária representa menos de 5% dos exemplos.

Será utilizado o auxílio da linguagem de programação *Python*, utilizando bibliotecas, como LightGBM e catboost, nelas tem o parâmetro embutido "class_weight", que ajuda a otimizar a pontuação para a classe minoritária da maneira.

Para encontrar uma ponderação ótima, estudaremos como a pontuação do *F1-score* se comporta a medida que modifica os pesos das classes, essa análise é importante de ser realizada, se reponderamos de forma incorreta estamos sujeitos ao efeito contrário, o

modelo se tornar tendencioso para a classe minoritária que agora passará a ser majoritária.

2.5.2 Uso da Técnica Class Weights em Relação aos Dados

Para melhor entendimento, vamos ilustrar como a técnica funciona em termos dos dados de financiamento de veículos.

Estamos trabalhando em um problema de classificação binária, onde queremos prever se um cliente é mau pagador ou não. O conjunto de dados tem 1000 observações, dos quais apenas 200 pertencem à classe positiva (inadimplentes) e 800 pertencem à classe negativa (adimplentes).

Suponha que atribuímos um peso de 19 para a classe positiva e um peso de 1 para a classe negativa. Isso significa que o modelo dará 19 vezes mais importância para a classe de inadimplentes durante o treinamento. Ou seja, o modelo tratará as observações de inadimplentes como se fossem 19 vezes mais importantes do que as observações de adimplentes. Isso significa que a perda (ou erro) do modelo será mais influenciada pelos exemplos da classe inadimplente, em comparação com os exemplos da classe adimplente.

Durante a etapa de treinamento, a função de perda do modelo será ponderada de acordo com esses pesos, de modo que os erros em indivíduos da classe positiva são penalizados mais fortemente do que os erros da classe negativa.

O modelo não replica as observações que corresponde aos inadimplentes, como acontece na técnica de Over-sampling. Em vez disso, ele ajusta os pesos atribuídos a cada observação durante o treinamento para levar em conta a desproporção entre as classes.

Em outras palavras, o modelo estará ajustando os pesos das amostras durante o treinamento de tal forma que as amostras da classe minoritária terão uma maior influência no ajuste dos parâmetros do modelo, o que ajudará a compensar o desequilíbrio na distribuição das classes e melhorar o desempenho do modelo para predições futuras.

Capítulo 3

Análise dos Dados

3.1 Dados

Os dados presentes para desenvolver o trabalho proposto foram coletados da publicação de Rodionov (2019) no site RPubs, site formado pela comunidade de cientista de dados em que o propósito é disseminar projetos de análise de dados e afins.

As informações são de um banco da Índia onde é realizado financiamento de veículos, trazem dados pessoais de clientes, histórico financeiro dos mesmos, informações contratuais, e por fim informa se houve ou não atraso de pagamento.

As variáveis presentes são:

- **Vl Desembolsado** Valor do empréstimo desembolsado para o financiamento;
- **Vl Custo do Ativo** Custo do Ativo;
- **LTV** Valor financiado;
- **Idade** Idade do proponente;
- **Tp Trabalho** Tipo do contrato de trabalho do proponente;
- **Fl Celular** Se o proponente apresentou o número do celular;
- **Fl Documento** Se o proponente apresentou o documento;
- **Fl N Conta** Se o proponente apresentou o número da conta;
- **Fl Titulo Eleitor:** Se o proponente apresentou o título de eleitor;

- **Fl CNH** Se o proponente apresentou a carteira de habilitação no momento da concessão;
- **Fl Passaporte** Se o proponente apresentou o passaporte;
- **Score Bureau** Pontuação do proponente no mercado financeiro;
- **Qt Emprestimos Tomados** Quantidade de empréstimos tomados;
- **Qt Emprestimos Ativos** Quantidade de empréstimos ativos;
- **Qt Restritivos Ativos** Quantidade de restritivos ativos;
- **Vl Emprestimos Ativos** Valor de empréstimos ativos;
- **Vl Emprestimos Tomados** Valor total de empréstimos tomados;
- **Qt Novos Emprestimos 6M** Quantidade de novos empréstimos nos últimos 6 meses;
- **Qt Restritivos 6M** Quantidade de restritivos adquiridos nos últimos 6 meses;
- **Qt Consultas Feitas** Quantidade de consultas feitas para financiamentos de veículos;
- **Prazo** Prazo para pagamento do financiamento;
- **Tempo Bancarizacao** Tempo de relacionamento com intituições financeiras;
- **Inadimplencia** Um dia de atraso em uma única parcela é considerado inadimplente.

3.2 Análise Descritiva

Para compreender melhor a qualidade da informação disponível no banco de dados, e obter uma prévia de comportamento de pagamentos dos proponentes, foi realizada a análise descritiva das variáveis explicativas em relação à variável resposta inadimplência.

3.2.1 Informações de Inadimplência

Existem diversas formas de mensurar a inadimplência, pode-se considerar como inadimplente depois de alguns dias de atraso, ou até mesmo olhar o atraso em um período, por exemplo, marcar como inadimplente proponentes que bateram 30 dias de atraso em

um período de 3 meses após a data do contrato, ou considerar um indicador mais curto, marcar como mau pagador clientes que bateram 30 dias de atraso logo na primeira parcela.

Neste banco de dados o autor apresenta um indicador mais rígido, caso o proponente atinja um dia de atraso em uma única parcela ele já é marcado como inadimplente na base.

Tabela 3.1: Proporção de Clientes Inadimplentes e Adimplentes

Classes	Contagem	Proporção
Adimplentes	103.924	77%
Inadimplentes	30.762	23%
Total	134.686	100%

O número total de observações é de 134.686, apresenta uma inadimplência de 23%, 30.762 indivíduos não realizaram o pagamento de uma parcela do financiamento, contra os 103.924 que realizaram o pagamento de todas parcelas.

3.2.2 Informações Pessoais do Proponente

Informações pessoais são fundamentais para a identificação do proponente, e contribuem diretamente para a prevenção de falsidade ideológica, garantindo que, o empréstimo de fato está sendo realizado pela pessoa ali presente no momento da concessão, não por um indivíduo se passando por outro.

As informações presentes são, tipo de contrato de emprego do proponente, idade, e variáveis dicotômicas que informam se uma informação em específico foi compartilhado pelo cliente no momento do financiamento ou não, como, por exemplo número do celular, identidade, número da conta bancária, título de eleitor, carteira de habilitação, número do passaporte, sendo sinalizadas como “Sim” ou “Não”.

Espera-se que quanto maior a quantidade de documentos compartilhados, mais fidedigno seja o proponente.

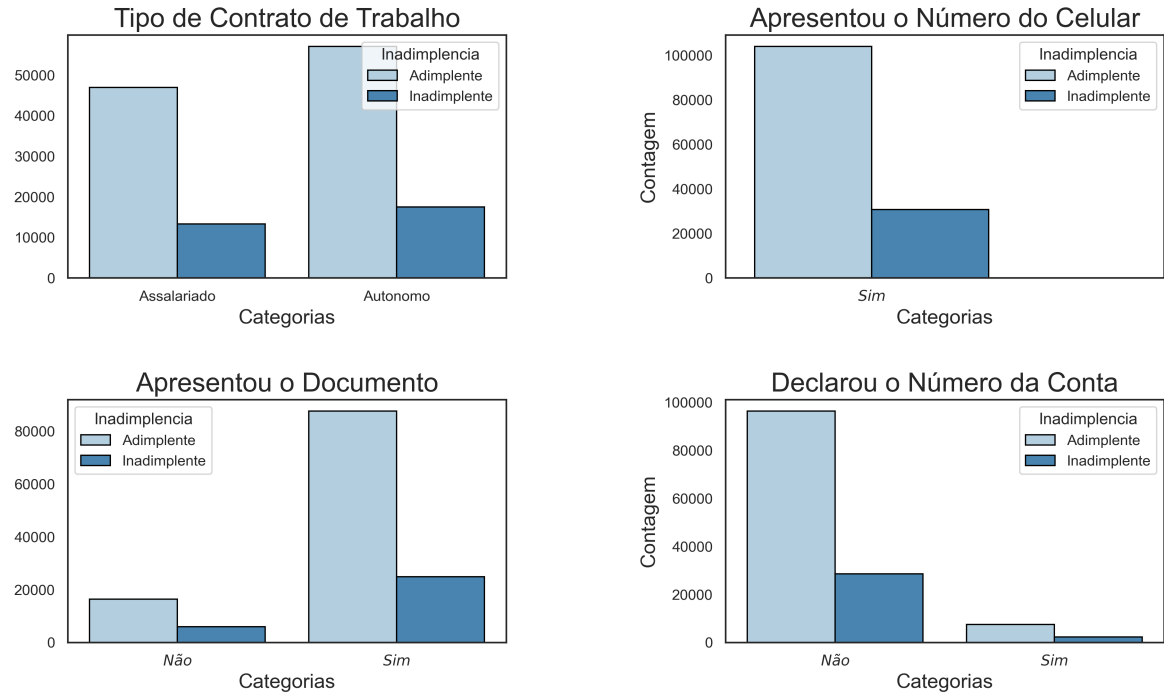

Figura 3.1: Contagem de Observações por Categoria

Para variável tipo de contrato de trabalho há mais trabalhadores autônomos do que assalariados, sendo que a sua proporção de inadimplência entre as classes tem a mesma proporção, sendo inconclusivo se algum tipo de contrato de trabalho tem mais inadimplentes do que adimplentes.

Todos os proponentes apresentaram o número do celular no momento da concessão, sendo assim a variável não discrimina entre as classes "sim" e "não", a inadimplência observada para o fator "não" é equivalente ao número de inadimplentes da base.

Na variável apresentou o documento, observamos que quase todos apresentaram documentos no momento da concessão, analisando apenas a classe "não" nota-se que a proporção de inadimplentes é maior do que a de inadimplentes para a classe que apresentou o documento.

Por fim, para a variável declarou o número da conta, cerca de 114 mil proponentes não declararam o número da conta, e apenas 20 mil acabam declarando, a proporção de inadimplentes em cada categoria são equivalentes, ou seja, não há confirmação de proponentes que não compartilham o número da conta no momento da concessão são piores pagadores.

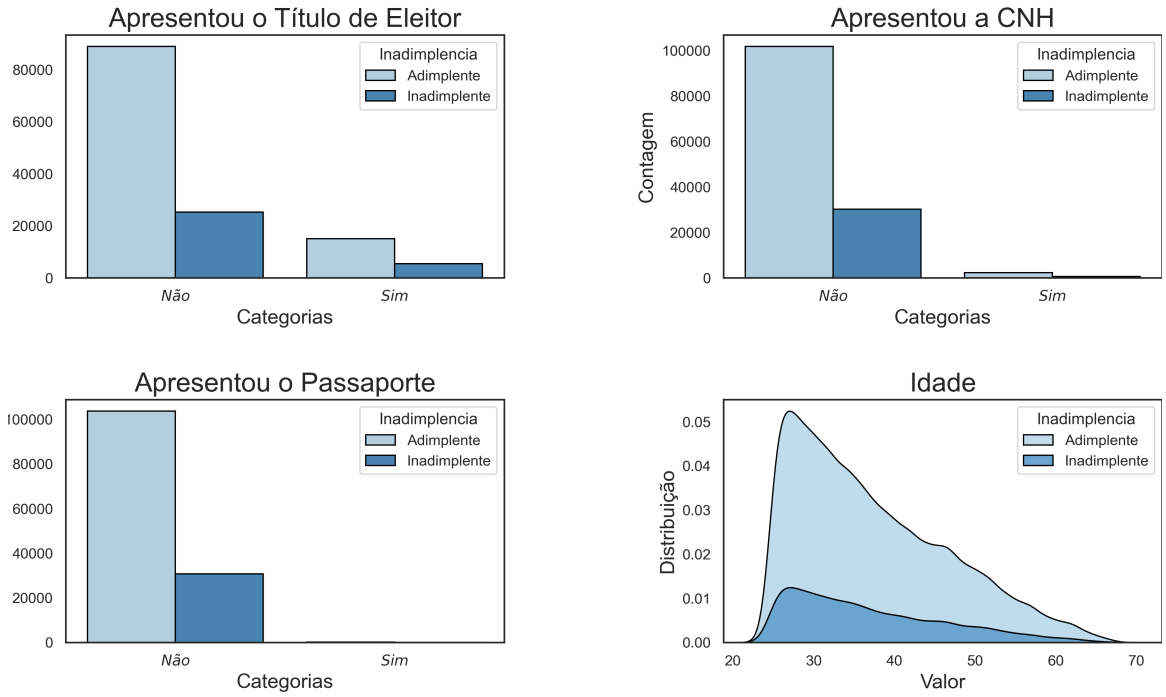

Figura 3.2: Contagem de Observações por Categoria

Apresentou o título de eleitor, grande parte dos proponentes da base não apresentam o título de eleitor, novamente não há nenhum fator desta variável que apresente um número maior de inadimplência.

Apresentou a carteira de habilitação, cerca de 95% dos proponentes não apresentaram a carteira de habilitação, a variável gera dúvidas sobre a sua veracidade, dado que, quando se trata de financiamento de veículos, é essencial que o proponente tenha carteira de habilitação.

Todos os proponentes não apresentaram o passaporte no momento da concessão, sendo assim a variável não discrimina entre as classes “sim” e “não”, a inadimplência observada para o fator “não” é equivalente ao número de inadimplentes da base.

Por fim, observando a distribuição da variável idade, é notável que grande parte dos proponentes tem menos de 40 anos, e que a inadimplência é mais concentrada no público jovem que é o majoritário na base, novamente a variável parece apresentar baixo poder de discriminação.

3.2.3 Informações do Empréstimo

Informações do empréstimo são as condições firmadas no contrato, as variáveis desta categoria são: porcentagem do valor financiado em relação ao valor total do automóvel (LTV), prazo do pagamento, valor do empréstimo desembolsado pelo banco e custo do ativo.

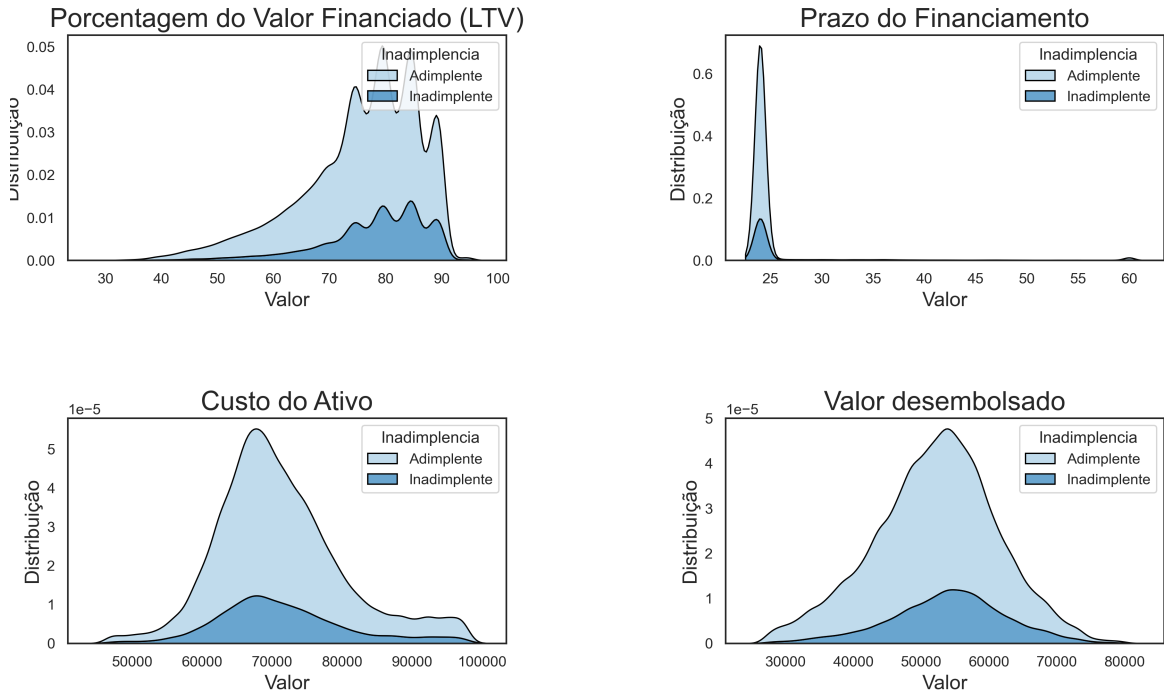

Figura 3.3: Curvas de Densidade

Para a porcentagem do valor financiado, a distribuição está concentrada principalmente entre 70%–90% apresentando uma assimetria a esquerda, esse formato de distribuição é dado porque dificilmente o proponente deseja financiar baixos valores, há uma redução para zero logo após o 90% possivelmente, não é comum financiamento sem entrada.

O prazo do financiamento é outra variável que se suspeita de erros, grande parte dos proponentes estão com prazo zero, o que não faz sentido, dado que a base se trata de financiamento de veículos, então necessita de tempo para quitar.

O custo do ativo e valor desembolsado apresentam a mesma forma entre os adimplentes e inadimplentes, indicando mais uma vez de que a variável não tem forte poder de discriminação.

3.2.4 Histórico Financeiro

O histórico financeiro descreve o seu comportamento de pagamento em contratos anteriores, essas informações ajudam a predizer se ele será um bom pagador ou não. Essas variáveis também é utilizado como uma estimativa para capacidade de pagamento do proponente, cuja intenção é prever uma quantia segura de empréstimo, por exemplo, se o indivíduo já tem muitos empréstimos tomados, não é saudável oferecer um novo de valor muito elevado.

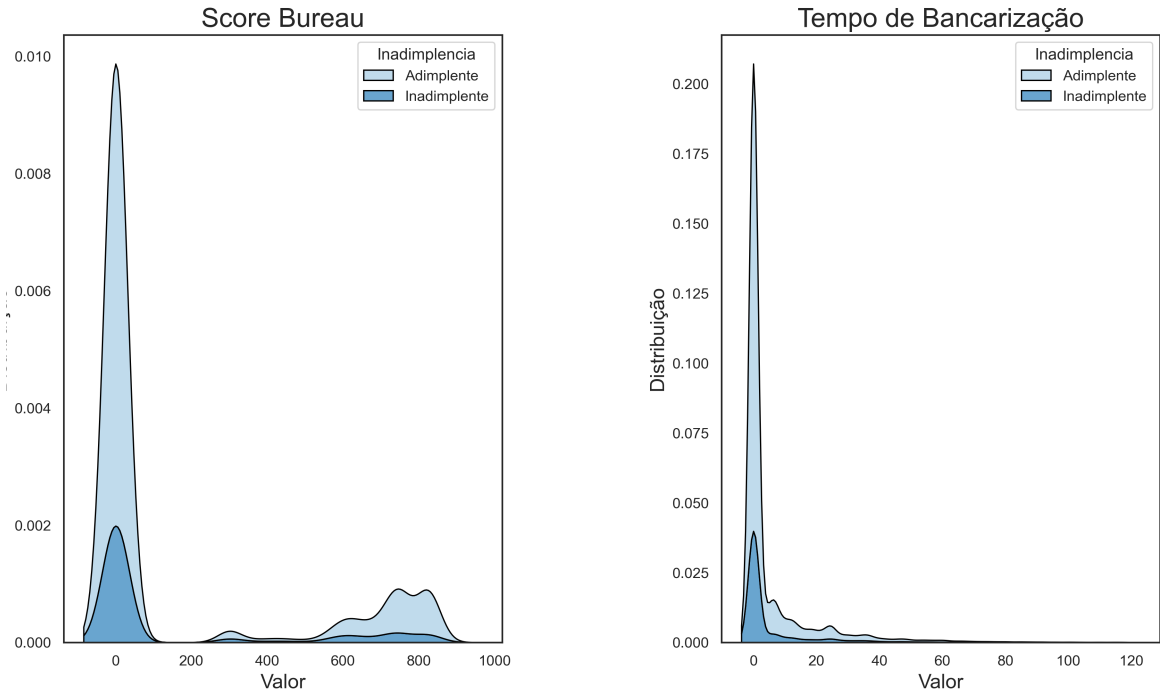

Figura 3.4: Curvas de Densidade

Para o *Score* Bureau, a variável mais promissora para discriminar inadimplência, tem grande parte de observações com pontuação zero, cuja descrição é que o bureau não conseguiu informar o *Score*, observando os proponentes que tem a informação, grande parte está concentrado entre as pontuações de 600 – 900.

Para tempo de bancarização, a maioria dos proponentes é não bancarizado, ou seja, tem valor zero, para os inadimplentes observa-se que toda a concentração está no zero, porém quase todas as observações estão também no zero, novamente fica difícil observar discriminação para a variável.

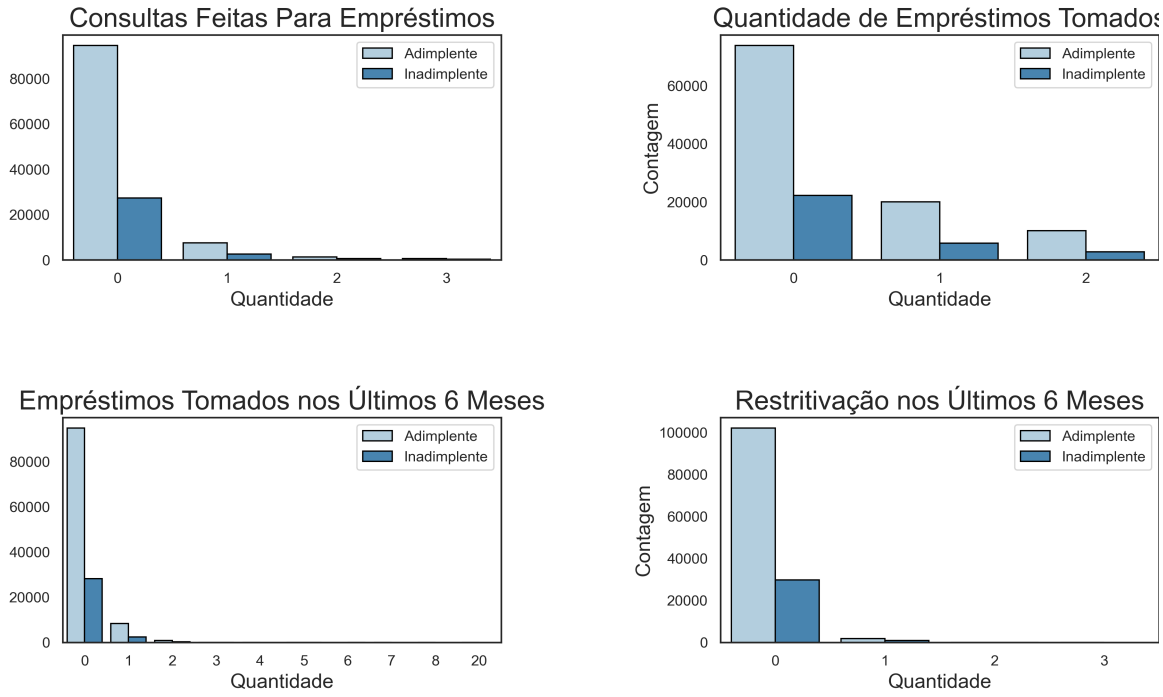

Figura 3.5: Informações Financeiras

Em consultas feitas para empréstimos, 80% da base não havia realizado consultas anteriormente para conseguir um empréstimo no mercado financeiro, nota-se que quanto maior a quantidade de consultas, maior é a taxa de inadimplência, as barras de adimplentes e inadimplentes passam a se igualar em tamanho.

Quantidade de empréstimos tomados, grande parte dos proponentes não tem empréstimos tomados, dificultando a discriminação esperada, quanto maior a quantidade de empréstimo tomados pior o proponente é, observando apenas os fatores um empréstimo tomado e dois empréstimos tomados, temos que a medida que se aumenta, a taxa de inadimplência também aumenta. Para as informações nos últimos meses, temos parte da população alocada na classe zero, novamente indicando que a variável possui baixo poder de discriminação para os adimplentes e inadimplentes.

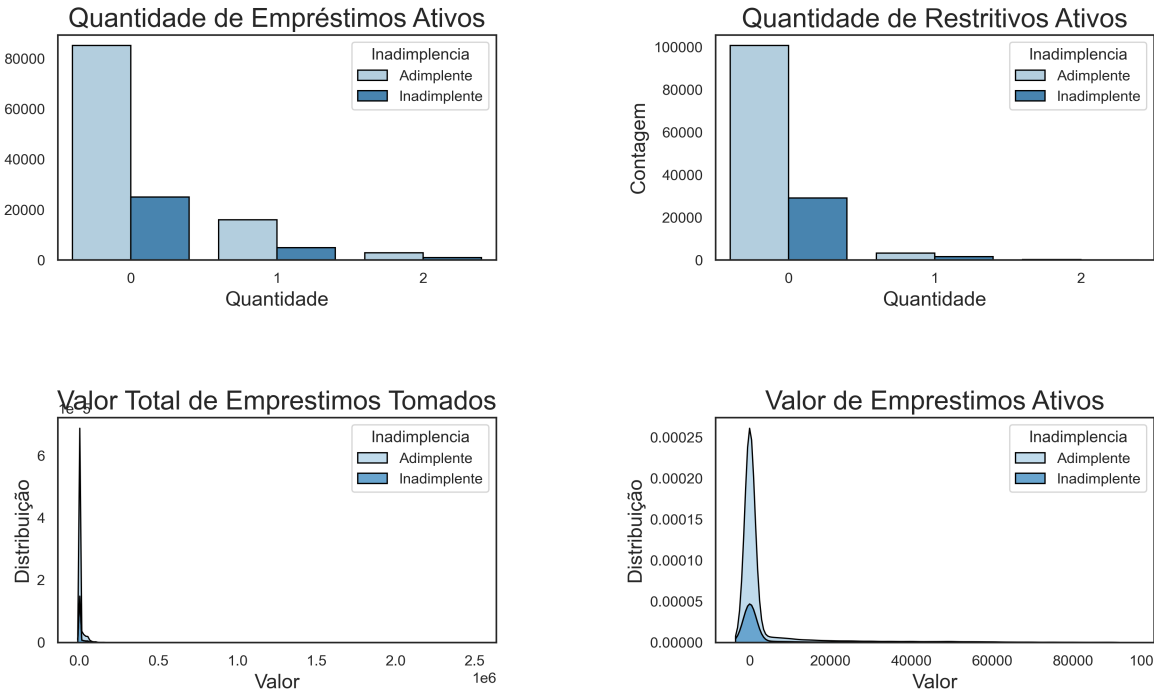

Figura 3.6: Informações Financeiras

Para as variáveis, quantidade de empréstimos ativos e quantidade de restritivos ativos, ao observar apenas as classes 1 e 2, com um volume observações bem pequenas, temos que à medida que se aumenta a quantidade também aumenta a taxa de inadimplência.

Já as outras duas variáveis de valores, toda a concentração está no zero, dado que grande parte da base não tem empréstimos ou restritivos ativos, novamente notamos o baixo poder de discriminação nas variáveis.

3.2.5 Correlação Variáveis Preditoras Quantitativas

A correlação de Pearson é uma associação estatística usualmente utilizada para medir o grau de relação entre um par de variáveis. Os valores de correlação vão de -1 até 1. Se a correlação for positiva, significa que o aumento de uma variável é acompanhado pelo aumento da outra, já com uma correlação negativa, temos o inverso, o aumento de uma variável é acompanhado pela diminuição da outra e vice-versa. Na matriz de correlação, temos os elementos fora da diagonal principal como as correlações entre os pares de variáveis.

A Figura 3.7, representa visualmente a matriz de correlações entre as variáveis, as cores mais intensas indicam um maior grau de correlação.

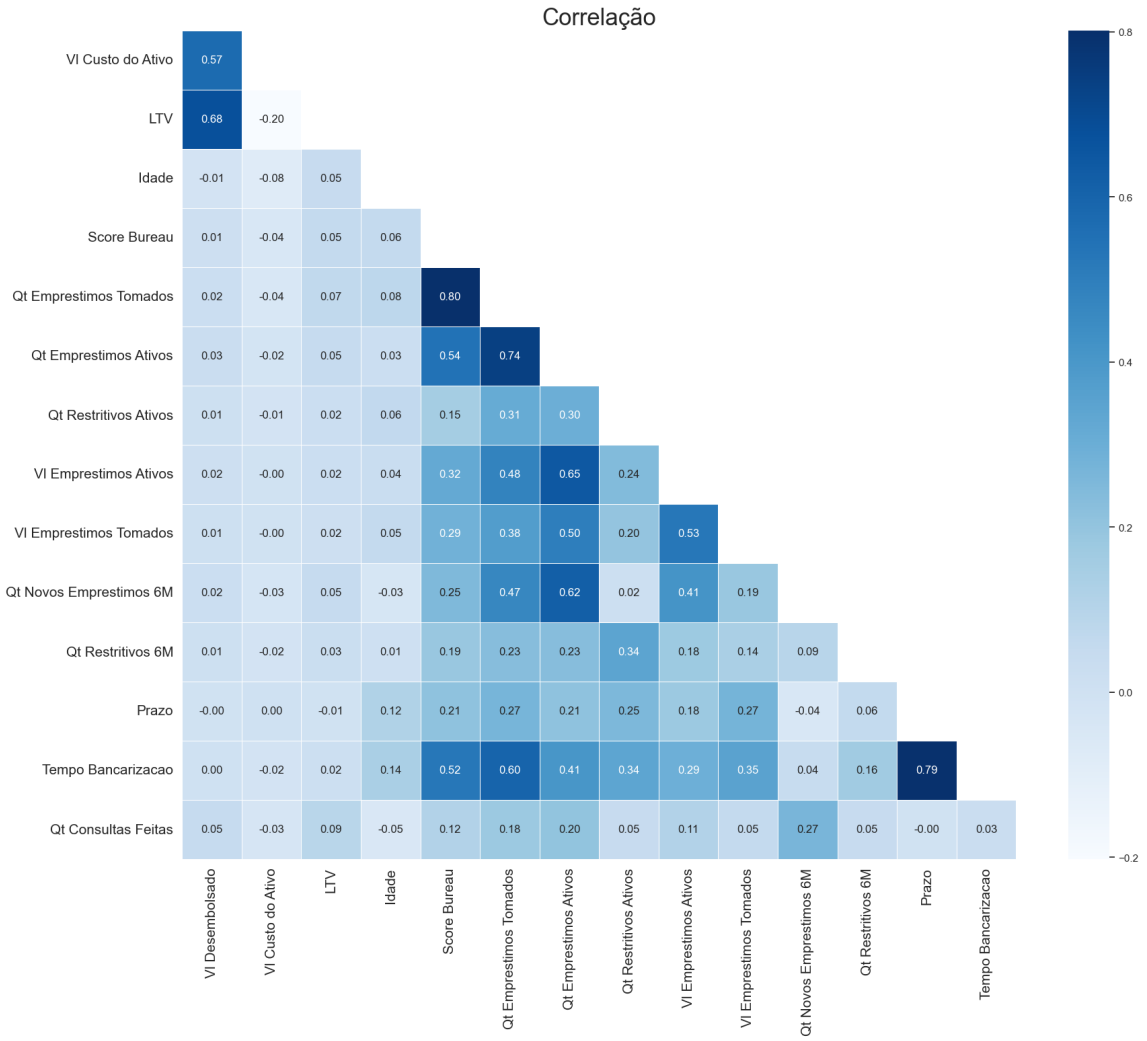

Figura 3.7: Correlação de Pearson

Podemos observar que o únicos pares com valores de correlação acima de 70% são os que tem grande volume de observações alocadas no número zero, e por isso não é necessário removê-las, pois não é um problema de colinearidade.

Conclusão Desta Etapa

Imagina-se que o ajuste dos modelos será um grande desafio, visto que temos muitas variáveis com informações defasadas, grande acumulo de observações em único fator ou agrupadas em uma região da distribuição e, pouca discriminação entre as classes das covariáveis em relação a categorias da variável resposta de adimplentes e inadimplentes.

Capítulo 4

Resultados

Nesta seção serão apresentados os ajustes de modelos propostos: para o primeiro as variáveis preditoras serão selecionadas via método valor da informação seguido com as métricas de qualidade para compreender o quão eficiente é a metodologia proposta para os dados.

4.1 Treino e Teste

Para o desenvolvimento do trabalho, em todos os ajustes serão utilizadas as mesmas observações do treino e do teste, para que no final, a comparação entre os modelos propostos seja a mais justa possível. Foi utilizado 70% da população para o conjunto de treino e 30% da população para o conjunto de teste.

Tabela 4.1: Separação Entre Treino e Teste

	Treino	Teste
Adimplentes	72747	31177
Inadimplentes	21533	9229
Taxa de Inadimplência	29,6%	29,6%

4.2 Ajuste de Modelos via Valor da Informação

4.2.1 Seleção de Variáveis

Com o auxílio do algoritmo implementado no *Python*, em que se calcula o peso da evidência e em seguida o valor da informação, temos a lista de variáreis ranqueadas em ordem decrescente em função dos seus respectivos valor da informação obtidos:

Tabela 4.2: Valor da Informação Obtidos

Variáveis	IV
LTV	0.096353
Vl Desembolsado	0.076305
Score Bureau	0.032786
Qt Restritivos Ativos	0.012989
Qt Restritivos 6M	0.009177
Fl Documento	0.007992
Fl Titulo Eleitor	0.007893
Qt Consultas Feitas	0.006757
Idade	0.004349
Vl Emprestimos Ativos	0.002094
Vl Custo do Ativo	0.001908
Tp Trabalho	0.001604
Tempo Bancarizacao	0.001314
Prazo	0.001023
Vl Emprestimos Tomados	0.000389
Qt Emprestimos Ativos	0.000233
Fl Passaporte	0.000063
Qt Novos Emprestimos 6M	0.000042
Fl Carteira de Habilitação	0.000015
Fl N Conta	0.000001
Fl Celular	0.000000

Após obter o valor da informação de cada preditora para a variável resposta, é notável que as variáveis presentes não aparentam ser muito promissoras para discriminar risco de inadimplência, pois o maior valor obtido é para a variável LTV, onde conforme a Tabela 2.2 está na classificação “poder preditivo fraco” em seguida temos Vl Desembolsado, *Score Bureau* e Qt Restritivos Ativos. As demais se enquadram na categoria “não é útil para a previsão” e não serão implementadas no ajuste.

Conclusão Desta Etapa

Esta etapa de seleção de variáveis via valor da informação, confirma o que foi concluído na etapa de análise descritiva dos dados, há um grande desafio na discriminação dos dados, a distribuição de adimplentes e inadimplentes em cada covariável seguem o mesmo formato, dificultado distinguir o comportamento de um cliente bom pagador em relação a um mau pagador.

Também foram implementadas metologias tradicionais de seleção de variáveis, como o StepWise e o Lasso, a fim de testar se as variáveis escolhidas pelo o IV também eram escolhidas por essas, as mesmas variáveis apareceram como promissoras para o ajuste do modelo em ambas metodologias. As metodologias tradicionais também selecionavam

outras variáveis presentes, porém pioravam os resultados para a qualidade do modelo. Sendo assim tivemos confiança para seguir com o proposto, a seleção de variáveis via valos da informação (IV).

4.2.2 Regressão Logística Múltipla

Após realizar o ajuste do modelo de regressão logística Múltipla, podemos realizar os diagnósticos do poder de predição do modelo:

Tabela 4.3: Valores Obtidos Teste KS

Estatística KS	Valor Obtido
Treino	14.99%
Teste	14.89%

O valor de KS para o treino e teste apresentado na Tabela 4.3 é de aproximadamente 15%, o que não é um valor significativo, ou seja, já na etapa de treino há dificuldade de avaliar se os proponentes inadimplentes e adimplentes provem da mesma população. Uma das causas de um KS baixo é o desbalanceamento dos dados ou as variáveis presentes quando implementadas no método proposto, que não contribui para diferenciar os bons e os maus pagadores.

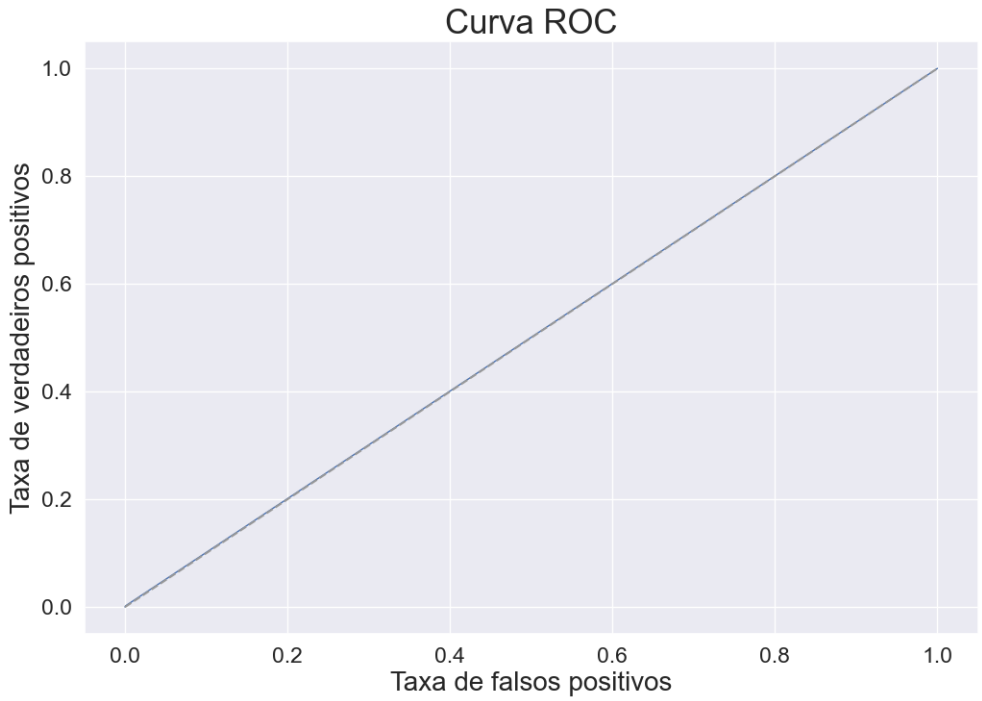

Figura 4.1: Curva ROC Regressão Logística

A curva ROC indica que a acurácia do ajuste do modelo está exatamente em 50%, ou

seja, exatamente em cima da linha do classificador aleatório, significando que o modelo fez uma péssima predição, assim não podemos afirmar que é um bom classificador.

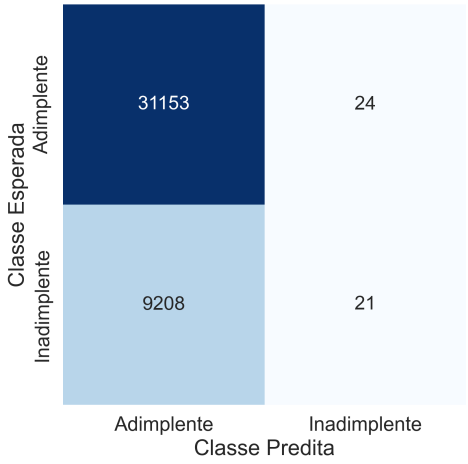

Figura 4.2: Matriz de Confusão Regressão Logística

A Figura 4.2 apresenta a matriz de confusão do modelo ajustado, o eixo y tem as informações da classe predita pelo modelo, enquanto, o eixo x está a classe esperada que são os valores verdadeiros. Pelos valores apresentados o modelo já indica que há problemas para realizar a predição, o modelo classificou quase todos os proponentes presentes como adimplentes, a qual é a classe majoritária, isso é consequência do desbalanceamento dos dados.

Tabela 4.4: Valores Obtidos Métricas de Qualidade

Métricas de Qualidade	Valor Obtido
Sensibilidade	77%
Precisão	99%

- **Sensibilidade:** a proporção de proponentes adimplentes sendo identificados corretamente é de 77%, esse valor é significativo, porém temos que tomar cuidado para afirmar que o ajuste se obteve uma boa sensibilidade, dado que, por conta do desbalanceamento o ajuste tem um viés para a classe majoritária que são os próprios adimplentes.

- **Precisão:** o modelo apresentou 99% de precisão, ou seja, o modelo classificou corretamente 99% dos verdadeiros adimplentes. O número é relevante, mas não significa que o modelo tem uma boa precisão, dado que, o modelo está classificando

grande parte dos proponentes como adimplente por conta dos desbalanceamento dos dados.

4.2.3 Regressão Logística Múltipla com Balanceamento dos Dados

Para o balanceamento dos dados, será utilizada a técnica de ponderação das variáveis. Para encontrar os pesos ideais utilizaremos o gráfico 4.3, no qual é apresentado a relação de peso para a classe inadimplente em relação ao *F1-score*, buscando o peso que maximize o *F1-score*.

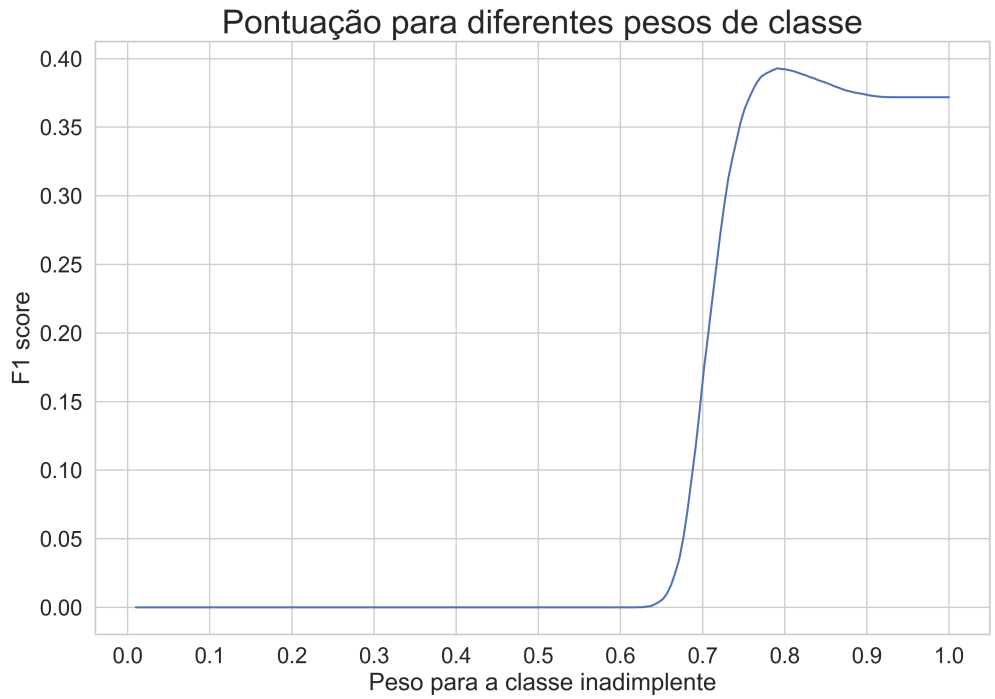

Figura 4.3: Curva Para Obter o Melhor Corte

Assim, para o ajuste do modelo, os pesos para a classe minoritária (inadimplentes) será de 0.77 enquanto o peso da classe majoritária (adimplentes) será de 0.23. Isso é, o peso para a classe de inadimplentes serão aproximadamente 3.3 vezes maiores do que a classe de adimplentes.

Em seguida temos a qualidade do modelo de regressão logística com o balanceamento dos dados:

Tabela 4.5: Valores Obtidos Teste KS

Estatística KS	Valor Obtido
Treino	15.00%
Teste	14.94%

Não há um ganho em KS quando comparado com os resultados da Tabela 4.3, o modelo continua apresentando um KS relativamente abaixo do esperado mesmo após o procedimento de balanceamento dos dados, indicando novamente que as variáveis presentes quando implementadas no método proposto, não tem forte contribuição para diferenciar os bons dos maus pagadores.

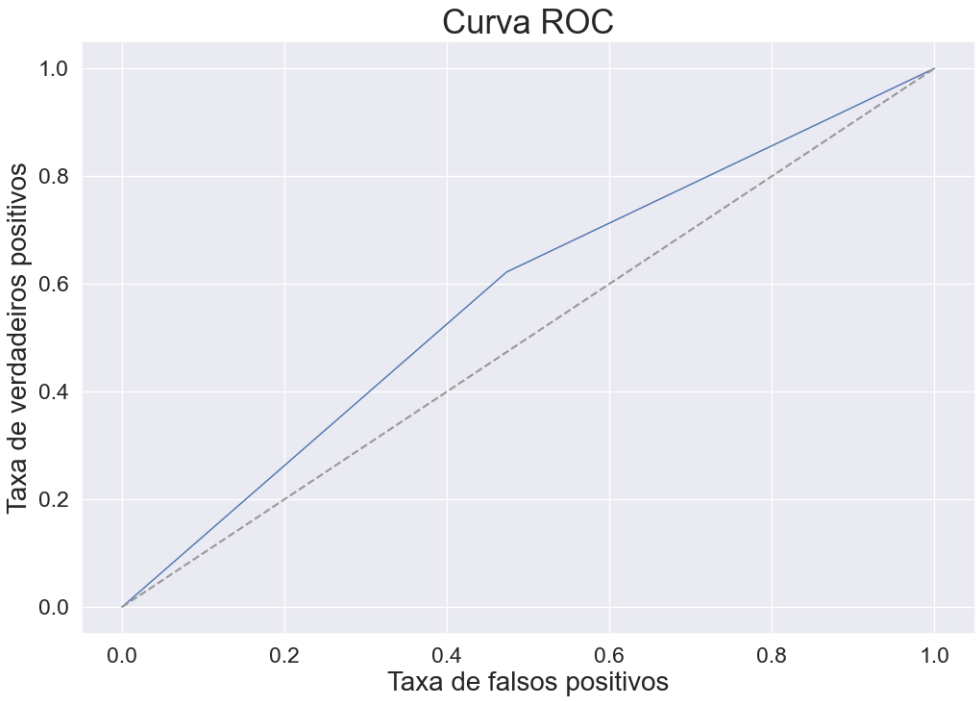

Figura 4.4: Curva ROC Regressão Logística Balanceada

Para a curva ROC, a linha se desloca levemente para a região considerada melhor, havendo um ganho para a taxa de verdadeiros positivos, a acurácia balanceada atinge o valor de 57%, sete pontos percentuais a mais que a acurácia do modelo de regressão logística sem o balanceamento dos dados.

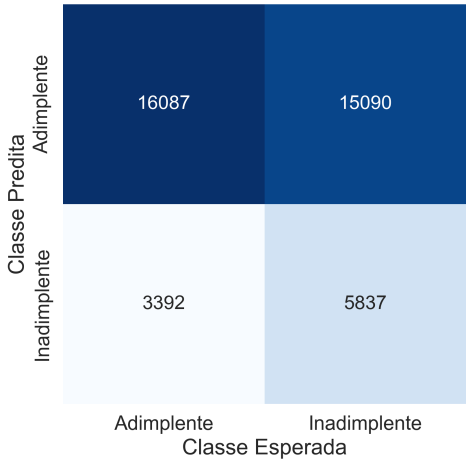

Figura 4.5: Matriz de Confusão Regressão Logística Balanceada

Interpretando a Figura 4.5, temos que ao realizar o procedimento de balanceamento das classes, o modelo comete mais erros, principalmente o mais grave, o qual é classificar o proponente como adimplente quando, na verdade, ele é inadimplente, esse erro para o cenário de crédito é extremamente prejudicial. Por outro lado, agora o modelo consegue identificar proponentes que de fato são maus pagadores.

Tabela 4.6: Valores Obtidos Métricas de Qualidade

Métricas de Qualidade	Valor Obtido
Sensibilidade	83%
Precisão	52%

- **Sensibilidade:** a proporção de proponentes adimplentes sendo identificados corretamente é de 83%, apresentando uma melhora na sensibilidade após o balanceamento das classes.

- **Precisão:** o modelo apresentou 52% de precisão, ou seja, o modelo classificou corretamente apenas 52% dos verdadeiros adimplentes. O valor é inferior ao ajuste anterior, porém esse resultado traz mais credibilidade por conta do balanceamento dos dados, removendo o viés de classificação da classe majoritária.

Conclusão Desta Etapa

Apesar do modelo apresentar métricas de qualidade plausíveis, conclui-se que para a concessão de crédito, não é um modelo acurado, de acordo com os números apresentados matriz de confusão ilustrada na Figura 4.5, pois o modelo classifica grande parte de

observações como adimplente quando na realidade é inadimplente. Concluindo, os dados disponíveis não se ajustam para a metodologia proposta, apenas para fins didáticos apresentaremos a classificação final nos capítulos seguintes.

Também foram implementadas técnicas tradicionais de reamostragem para os dados, como o Over-Sampling e o Under-Sampling, a fim de testar se o ajuste também cometia o pior erro de classificar adimplentes como inadimplentes na mesma intensidade que a técnica de reponderação de classes. Ambas as técnicas cometo o erro com a mesma intensidade.

4.3 Classificação Final

A partir da pontuação obtida através do modelo de regressão logístico ajustado, é apresentada a classificação final de risco do proponente para financiamento de veículos.

O algoritmo da árvore de classificação auxilia a se ter quebras ótimas do *Score* obtido com o ajuste do modelo, gerando classes balanceadas e com ordenação para a probabilidade de inadimplência:

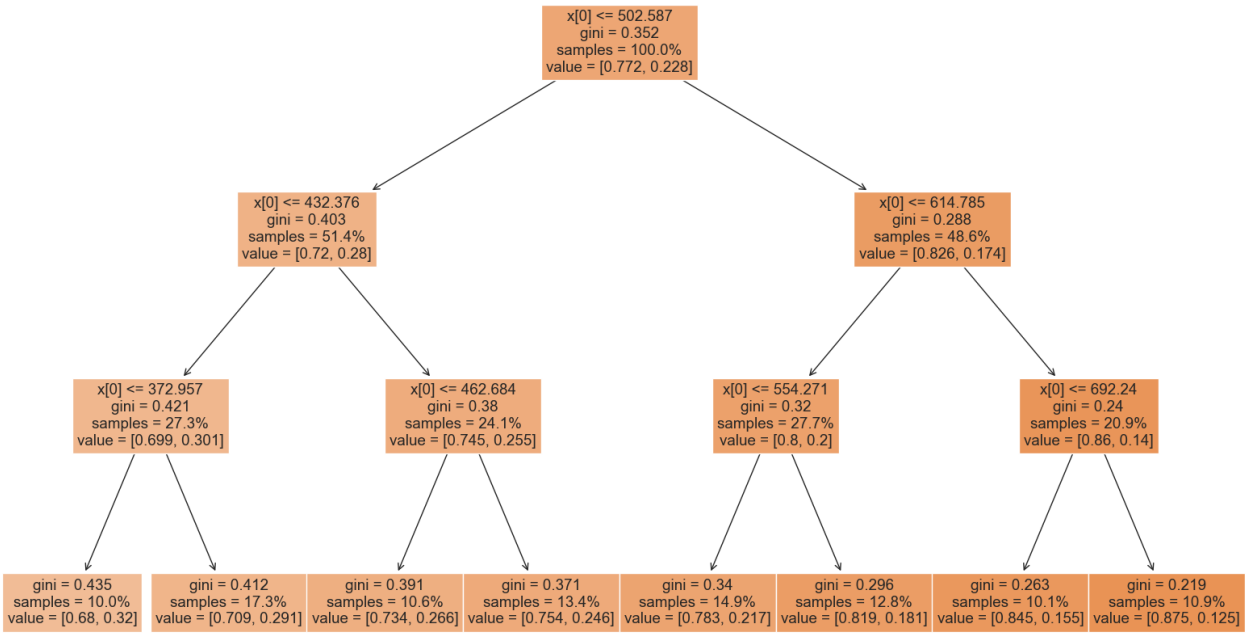

Figura 4.6: Saída do *Score* e Faixas Obtidas Através da Árvore de Classificação.

As faixas são dadas pelos nós, e a volumetria e inadimplência é dada pelas folhas da última camada.

E finalmente temos a classificação final, ou seja, a partir das quatro variáveis selecionadas, sendo elas, LTV, valor desembolsado, score bureau, e quantidade de restritivos ativos, foi possível construir uma única pontuação final, buscando uma melhor discriminação de risco para o público que busca por um financiamento de veículos.

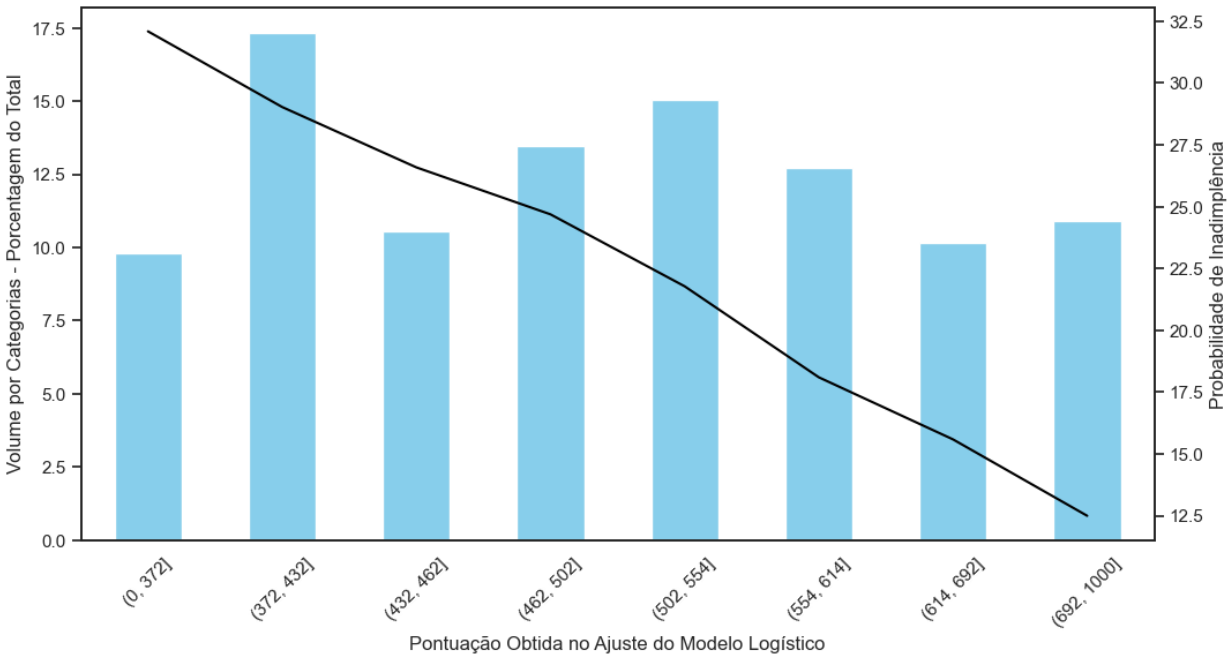

Figura 4.7: Classificação de Risco do Proponente

A Figura 4.7 ilustra a classificação final, temos 8 classes distribuídas, todas elas com volumetria considerável e com uma clara ordenação da inadimplência, a medida que se aumenta a pontuação a inadimplência diminui.

É possível identificar que a pior faixa da classificação apresenta uma taxa de inadimplência de 32.5%, já no outro extremo do gráfico, temos a melhor faixa com uma inadimplência média de 12.5%.

Capítulo 5

Conclusão

Logo na etapa de análise dos dados, já temos indícios do quão desafiador será o ajuste do modelo. Os dados disponíveis para as covariáveis estão extremamente desbalanceados e, ao observar as covariáveis com as quebras dos fatores existentes na variável resposta (inadimplente e adimplente), nota-se baixo poder de discriminação, dificultando a discriminação das populações.

Na etapa de seleção de variáveis, se confirma o que foi concluído na análise descritiva, a variável LTV que atingiu a maior pontuação possível para o valor da informação, está na classificada como "baixo poder preditivo". Por fim, selecionam quatro variáveis pertencentes a essa classificação, sendo elas LTV, valor desembolsado, score bureau e quantidade de restritivos ativos, todas com baixo poder de discriminação da variável resposta.

Ao realizar o primeiro ajuste logístico, pelas métricas de qualidade do modelo, conclui-se que se faz necessária a técnica de balanceamento dos dados, a matriz de confusão ilustra o quão enviesado o modelo está em classificar apenas a classe majoritária (adimplentes), com uma precisão e sensibilidade falsa.

Ajustando o modelo logístico com os dados balanceados pela técnica de ponderação das variáveis, ganha-se poder preditivo, melhoram-se as métricas de sensibilidade e precisão, porém, cometemos o pior erro possível, classificar inadimplentes como adimplentes, o valor obtido do KS de 15% também é um ponto de atenção, dado que o mercado financeiro considera um bom KS valores acima de 30%. Para o crédito o erro mencionado deve ser altamente considerado, em cenários reais não é recomentado o uso do modelo com esse resultado obtido pelas métricas de qualidade, os dados não se ajustaram bem as técnicas propostas, uma forma de melhorar a predição é utilizar metodologias XGBoost com otimização de hiperparametros.

Na etapa final, a árvore de classificação se mostrou eficiente, retornando classificações com ordenação na inadimplência e, simultaneamente, classes com volumes consideráveis.

Por fim, concluindo sobre o tema de financiamentos de veículos, com as variáveis LTV, valor desembolsado, *score bureau* e quantidade de restritivos ativos, é possível mensurar em um único valor o risco para financiamentos de veículos, obtendo oito classes cuja inadimplência diminui a medida em que se aumenta o valor do **rating**. É possível localizar grupos mais críticos nas faixas de $(0 - 372]$ com inadimplência média de 32.5%, fazendo necessário condições menos permissíveis de financiamentos, e também grupos menos críticos nas faixas de $(692 - 1000]$ com inadimplência média de 12.5% sendo possível conceder mais crédito com condições mais permissíveis para os proponentes localizados nessa classificação.

Referências Bibliográficas

B3 (2022). Mercado de financiamentos de veículos. https://www.b3.com.br/pt_br/market-data-e-indies/informacoes-para-mercado-de-financiamentos/veiculos/. Acesso em 25 de julho.

Izbicki, R. e dos Santos, T. M. (2020). *Aprendizado de máquina: uma abordagem estatística*.

MartinThoma (2021). Deutsch: Roc-kurve - die abszisse ist die falsch-positiv-rate und die ordinate die richtig-. https://commons.wikimedia.org/wiki/File:Roc-draft-xkcd-style.svg#filelinks. Acesso em 18 de março de 2023.

Paula, G. A. (2013). *MODELOS DE REGRESSÃO com apoio computacional*. IME.

PICININI, R. (2003). *Mineração de Critério de Credit Scoring Utilizando Algoritmos Genéticos.*. VI Simpósio Brasileiro de Automação Inteligente.

Rocheny, S. (2020). *Does Psychometric Testing in Microfinance Actually Work?—The Case of Sogesol*. Journal of Financial Risk Management.

Rodionov, A. (2019). Vehicle loan default prediction (classification problem). https://rstudio-pubs-static.s3.amazonaws.com/523852_e9f2abf3ebff486c8ed0fb0920d84b22.html. Acesso em 29 de agosto.

Singh, K. (2020). How to improve class imbalance using class weights in machine learning. https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/. Acesso em 20 de março.

Votorantim, B. (2022). O mercado de automóveis em retomada: veja as tendências futuras. https://www.bv.com.br/bv-inspira/parceiro-veiculos/mercado-de-automoveis. Acesso em 25 de julho.

Capítulo 6

Apêndice:

.1 Bibliotecas Utilizadas

```
#basics
import pandas as pd
import datetime as dt
import numpy as np
import scipy
import matplotlib.pyplot as plt
import matplotlib.pyplot as annotate
%matplotlib inline
import seaborn as sns
sns.set()
import plotly.express as px
import plotly.graph_objects as go
import plotly
import scikitplot as skplt
import missingno as msno
import scorecardpy as sc
from scipy.stats import ks_2samp
from sklearn.linear_model import Ridge, Lasso
import shap
import statsmodels.api as sm

from sklearn.model_selection import GridSearchCV, StratifiedKFold
from stepwise_regression import step_reg

#padronozar os dados
from sklearn.preprocessing import StandardScaler
import numpy as np

# import the class
from sklearn.linear_model import LogisticRegression
```

```
#encoder
from category_encoders import OrdinalEncoder

##models
from sklearn.model_selection import (train_test_split,
                                     cross_val_score,
                                     RandomizedSearchCV,
                                     StratifiedKFold,
                                      validation_curve,
                                      cross_val_score,
                                      KFold)
from sklearn.preprocessing import QuantileTransformer
from sklearn.preprocessing import (MinMaxScaler,StandardScaler,RobustScaler,Normalizer)
from plot_metric.functions import BinaryClassification
from sklearn.metrics import  (accuracy_score,
                              auc,
                              precision_score,
                              recall_score,
                              classification_report,
                              balanced_accuracy_score,
                              make_scorer,
                              roc_auc_score,
                              roc_curve,
                              confusion_matrix,
                              mean_squared_error)
from sklearn.tree import DecisionTreeClassifier,export_graphviz, plot_tree
from scipy.stats import randint

from warnings import filterwarnings
filterwarnings(action='ignore', category=DeprecationWarning, message='`np.bool` is a deprecated alias')

from sklearn import tree
```

.2 Funções

```
#funcoes

def ks_stat(y, yhat):
    return ks_2samp(yhat[y==1], yhat[y!=1]).statistic

ks_scorer = make_scorer(ks_stat, needs_proba=True)

def ks_geral(df, var, bad):
    ks = round(ks_2samp(df[(df[bad]==0)][var],
        df[(df[bad]==1)][var])[0]*100,2)
    return ks
```

```
def plot_description(ax, x, title):
    ax.text(0.3, 0.7, title, fontsize=18)
    ax.text(0.3, 0.6, 'mean\t{:.4f}'.expandtabs().format(x.mean()), fontsize=18,
    fontfamily='monospace')
    ax.text(0.3, 0.5, 'stdev\t{:.4f}'.expandtabs().format(x.std()), fontsize=18,
    fontfamily='monospace')
    ax.text(0.3, 0.4, 'min\t{:.4f}'.expandtabs().format(x.min()), fontsize=18,
    fontfamily='monospace')
    ax.text(0.3, 0.3, 'max\t{:.4f}'.expandtabs().format(x.max()), fontsize=18,
    fontfamily='monospace')
    ax.axis('off')


def val_bins(df):
    return [0] + [df.sort_values().values.unique()[i].right for i in range(20)]

def importance_shap(model, df, vars_model):
    explainer_model = shap.TreeExplainer(model)
    expected_value = explainer_model.expected_value

    shap_values = explainer_model.shap_values(df[vars_model])

    fig, ax = plt.subplots(figsize=(25,15))

    plt.subplot(211)
    shap.summary_plot(shap_values, df[vars_model], plot_type="bar",class_inds=[0])

    plt.subplot(212)
    shap.summary_plot(shap_values[0], df[vars_model], plot_type="dot", show = False)

    plt.show()

def evaluate_model(df, x_train, x_test, y_train, y_test, idx_train, idx_test,
                   score, target, model, vars_model):
    print("\n===== KS MODELO =====")
    print("KS Treino", ks_geral(df.loc[idx_train],
                                score,
                                target))

    print("KS Teste", ks_geral(df.loc[idx_test],
                                score,
                                target))

    y_predicted = model.predict(x_test[vars_model])

    print(classification_report(y_test,
                                y_predicted,
                                target_names=["Bom", "Mau"]))
```

```
    balanced_accuracy = balanced_accuracy_score(y_test, y_predicted)

    print("====Acurácia Balenceada====\n\t", balanced_accuracy)

    print("====ROC Curve====")
    fpr, tpr, thresold = roc_curve(y_test, y_predicted)
    plt.figure(figsize = ((12,8)))
    plt.plot(fpr, tpr, lw = 1)
    plt.plot([0,1],[0,1], '--', color=(0.6, 0.6, 0.6), label="sorte")
    plt.xlim([-0.05,1.05])
    plt.ylim([-0.05,1.05])
    plt.xlabel("Taxa de falsos positivos", fontsize=20)
    plt.ylabel("Taxa de verdadeiros positivos", fontsize=20)
    plt.title("Curva ROC", fontsize=25)
    plt.show()


# Function to create a confusion matrix
def conf_matrix(y_test, y_pred):

    # Creating a confusion matrix
    con_mat = confusion_matrix(y_test, y_pred)
    con_mat = pd.DataFrame(con_mat, range(2), range(2))

    #Ploting the confusion matrix
    plt.figure(figsize=(6,6))
    sns.set(font_scale=1.5)
    sns.heatmap(con_mat, annot=True, annot_kws={"size": 16}, fmt='g', cmap='Blues', cbar=False ,
                xticklabels= ["Adimplente", "Inadimplente"],
                yticklabels= ["Adimplente", "Inadimplente"] )

    print(classification_report(y_test,
                                y_pred,
                                target_names=["Bom", "Mau"]))

def ks_por_safra_df(df, bad_col, score_cols, safra_col):


    #Função do IV
def get_IV(df, feature, target, cat = True, q = 10):
    if cat == True:
        lst = []
    # optional
     # df[feature] = df[feature].fillna("NULL")
        unique_values = df[feature].unique()
        for val in unique_values:
              lst.append([feature,
                      val,
                      df[(df[feature] == val) & (df[target] == 0)].count()[feature],
```

```
                    df[(df[feature] == val) & (df[target] == 1)].count()[feature]
                ])
        data = pd.DataFrame(lst, columns=['Variable', 'Value', 'Good', 'Bad'])
        total_bad = df[df[target] == 1].count()[feature]
        total_good = df.shape[0] - total_bad
        data['Distribution Good'] = data['Good']/ total_good
        data['Distribution Bad'] = data['Bad'] / total_bad
        data['WoE'] = np.log(data['Distribution Good'] / data['Distribution Bad'])
        data = data.replace({'WoE': {np.inf: 0, -np.inf: 0}})
        data['iv'] = data['WoE'] * (data['Distribution Good'] - data['Distribution Bad'])
        data = data.sort_values(by=['Variable', 'Value'], ascending=[True, True])
        data.index = range(len(data.index))
        print(data)
        iv = data['iv'].sum()
    else:
        data = pd.crosstab(pd.cut(df[feature], q), df[target])
        data.columns = ['0', '1']
        total_bad = data['1'].sum()
        total_good = data['0'].sum()
        data['Distribution Good'] = data['0']/ total_good
        data['Distribution Bad'] = data['1'] / total_bad
        data['WoE'] = np.log(data['Distribution Good'] / data['Distribution Bad'])
        data = data.replace({'WoE': {np.inf: 0, -np.inf: 0}})
        data['iv'] = data['WoE'] * (data['Distribution Good'] - data['Distribution Bad'])
        print(data)
        iv = data['iv'].sum()
    return iv

def get_iv_df(df, features_to_drop, quantile, target, q = 10,  threshold_iv  = 0.05):
    df_drop = df.drop(df[features_to_drop], axis = 1)
    scaled_df = df_drop
    vars_iv = scaled_df.columns.tolist()
    for i in vars_iv:
        if str(scaled_df.dtypes[i]) == 'object' or str(scaled_df.dtypes[i]) == 'category':
            scaled_df[i] = scaled_df[i].astype('object')
    if target in vars_iv:
        vars_iv.remove(target)
    list_iv = []
    for i in vars_iv:
        if str(scaled_df.dtypes[i]) == 'object' or str(scaled_df.dtypes[i]) == 'category':
            scaled_df = scaled_df.fillna('Null')
            cat = True
        else:
            scaled_df = scaled_df.fillna(0.0)
            cat = False
        iv = get_IV(scaled_df, i, target, cat, q)
        list_iv.append(iv)
    data = {'Variáveis':  vars_iv,
        'iv': list_iv
```

```
        }
    df_iv = pd.DataFrame (data, columns = ['Variáveis', 'iv'])
    #df_iv = df_iv.sort_index(inplace=True)
    df_iv = df_iv.sort_values('iv', ascending = False).reset_index()
    df_iv = df_iv.drop(['index'], axis = 1)
    df_iv = df_iv[df_iv['iv'] >= threshold_iv]
    return df_iv

def correlation_iv(df, columns , threshold_corr = 0.995, target='target', q=10,
threshold_iv = 0.00, frac = 0.8, seed = 0):
    df_aux = df.filter(items=columns)
    if frac !=  1:
        df_aux = df_aux.sample(frac=frac, replace = False, random_state=seed)
    #get iv
    df_iv = get_iv_df(df_aux, [], 1.0, target, q,  threshold_iv)
    #get corr
    corr_matrix = df_aux.corr().abs()
    #del columns highly correlated
    col_corr = [] # Set deleted columns
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if (corr_matrix.iloc[i, j] != np.nan) or (corr_matrix.iloc[i, j] != None):
                if (corr_matrix.iloc[i, j] >= threshold_corr)
                and (corr_matrix.columns[j] not in col_corr)
                and (corr_matrix.columns[j] != target):
                    iv = df_iv[df_iv['Variáveis'].isin([corr_matrix.columns[j],
                    corr_matrix.columns[i]])]
                    iv = iv[iv['iv']==iv['iv'].min()] #get min iv and drop
                    iv.reset_index(inplace = True)
                    iv = iv.drop(['index'], axis = 1)
                    colname = str(iv.iloc[0, 0])
                    col_corr.append(colname)
                    if colname in df_aux.columns:
                        del df_aux[colname]
    return list(set(col_corr))
```

.3 Tratamento dos Dados

```
vars_modelagem = ['Vl Desembolsado', 'Vl Custo do Ativo', 'LTV', 'Idade', 'Tp Trabalho',
                  'Fl Celular', 'Fl Documento', 'Fl N Conta', 'Fl Titulo Eleitor',
                  'Fl CNH', 'Fl Passaporte', 'Score Bureau', 'Qt Emprestimos Tomados',
                  'Qt Emprestimos Ativos', 'Qt Restritivos Ativos',
                  'Vl Emprestimos Ativos', 'Vl Emprestimos Tomados',
                  'Qt Novos Emprestimos 6M', 'Qt Restritivos 6M', 'Prazo',
                  'Tempo Bancarizacao', 'Qt Consultas Feitas', 'Inadimplencia']

# Tirando duplicidade dos dados
```

```
df.drop_duplicates(subset=vars_modelagem,inplace=True)
df.reset_index(inplace=True, drop=True )
df.shape

# excluindo valor nulo na variável resposta
df.dropna(subset=['Inadimplencia'],inplace=True)
df.reset_index(inplace=True, drop=True )
df.shape

# Não há presença de valores nulos na variável resposta

# Excluíndo linhas que contém NA nas variáveis preditoras (Employment.Type)

df.dropna(subset=['Tp Trabalho'],inplace=True)
df.reset_index(inplace=True, drop=True )
df.shape
```

.4 Análise Descritiva

```
vars_corr = ['Vl Desembolsado', 'Vl Custo do Ativo', 'LTV', 'Idade',
             'Score Bureau', 'Qt Emprestimos Tomados',
             'Qt Emprestimos Ativos', 'Qt Restritivos Ativos',
             'Vl Emprestimos Ativos', 'Vl Emprestimos Tomados',
             'Qt Novos Emprestimos 6M', 'Qt Restritivos 6M', 'Prazo',
             'Tempo Bancarizacao', 'Qt Consultas Feitas']

def plot_corr(df):
    corr_ = df.corr()

    fig, ax = plt.subplots(figsize=(20, 16))

    mask = np.triu(np.ones_like(corr_, dtype=np.bool))

    mask = mask[1:, :-1]
    corr_cp = corr_.iloc[1:,:-1].copy()

    sns.heatmap(corr_cp,
                mask=mask,
                annot=True,
                fmt=".2f",
                linewidths=.6,
                cmap="Blues")

    plt.title("Correlação", size=25)
    plt.xticks(rotation=90, size=15)
    plt.yticks(rotation=0, size=15)
```

```
    return plt.show()

plot_corr(df[vars_corr])
plt.savefig('CORRELAÇÃO.png', dpi = 300)

plt.figure(figsize = ((12,8)))
plt.subplot(2,2,1)
sns.countplot(x ='Tp Trabalho',
              data = df,
              hue = "Inadimplencia",
              palette = "Blues",
              ec='black',
              lw = 1)

plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Tipo de Contrato de Trabalho", fontsize = 20)


plt.subplot(2,2,2)
ax = sns.countplot(x ='Fl Celular',
                   data = df,
                   hue = "Inadimplencia",
                   palette = "Blues",
                   ec='black',
                   lw = 1)
ax.set_xticks((1, 0))
ax.set_xticklabels(('$Não$', '$Sim$'))

#for p in ax.patches:
#        height = p.get_height()
#        ax.text(p.get_x()+p.get_width()/2,
#        height + 5,
#        '{:1.2f}'.format((height)),
#        ha='center')

plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Apresentou o Número do Celular", fontsize = 20)


plt.subplot(2,2,3)
ax = sns.countplot(x ='Fl Documento',
                   data = df,
                   hue = "Inadimplencia",
                   palette = "Blues",
                   ec='black',
                   lw = 1)
```

```
ax.set_xticks((0, 1))
ax.set_xticklabels(('$Não$', '$Sim$'))
plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Apresentou o Documento", fontsize = 20)

plt.subplot(2,2,4)
ax = sns.countplot(x ='Fl N Conta',
                   data = df,
                   hue = "Inadimplencia",
                   palette = "Blues",
                   ec='black',
                   lw = 1)
ax.set_xticks((0, 1))
ax.set_xticklabels(('$Não$', '$Sim$'))
plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Declarou o Número da Conta", fontsize = 20)

#plt.suptitle("Variáveis Categóricas Por Inadimplência", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.5)
#plt.savefig('VarCat.png', dpi = 300)
plt.show()

plt.figure(figsize = ((12,8)))

plt.subplot(2,2,1)
ax = sns.countplot(x ='Fl Titulo Eleitor',
                   data = df,
                   hue = "Inadimplencia",
                   palette = "Blues",
                   ec='black',
                   lw = 1)
ax.set_xticks((0, 1))
ax.set_xticklabels(('$Não$', '$Sim$'))
plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Apresentou o Título de Eleitor", fontsize = 20)


plt.subplot(2,2,2)
ax = sns.countplot(x ='Fl CNH',
                   data = df,
                   hue = "Inadimplencia",
                   palette = "Blues",
                   ec='black',
                   lw = 1)
ax.set_xticks((0, 1))
ax.set_xticklabels(('$Não$', '$Sim$'))
```

```
plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Apresentou a CNH", fontsize = 20)


plt.subplot(2,2,3)
ax = sns.countplot(x ='Fl Passaporte',
              data = df,
              hue = "Inadimplencia",
              palette = "Blues",
              ec='black',
              lw = 1)
ax.set_xticks((0, 1))
ax.set_xticklabels(('$Não$', '$Sim$'))
plt.ylabel("Contagem", fontsize = 15)
plt.xlabel("Categorias", fontsize = 15)
plt.title("Apresentou o Passaporte", fontsize = 20)

plt.subplot(2, 2, 4)
sns.kdeplot(data=df, x='Idade', hue="Inadimplencia", multiple="stack", ec='black', palette = "Blues", l
plt.title("Idade", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)


#plt.suptitle("Variáveis Categóricas Por Inadimplência", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.5)
plt.savefig('VarCat2.png', dpi = 300)
plt.show()

plt.figure(figsize = ((12,8)))
plt.subplot(2, 2, 1)
sns.kdeplot(data=df, x='LTV', hue="Inadimplencia", multiple="stack", ec='black', palette = "Blues", lw
plt.title("Porcentagem do Valor Financiado (LTV)", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)

plt.subplot(2, 2, 2)
sns.kdeplot(data=df, x='Prazo', hue="Inadimplencia", multiple="stack", ec='black', palette = "Blues", l
plt.title("Prazo do Financiamento", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)


plt.subplot(2, 2, 3)
sns.kdeplot(data=df, x='Vl Custo do Ativo', hue="Inadimplencia", multiple="stack", ec='black', palette
plt.title("Custo do Ativo", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)
```

```
plt.subplot(2, 2, 4)
sns.kdeplot(data=df, x='Vl Desembolsado', hue="Inadimplencia", multiple="stack", ec='black', palet
plt.title("Valor desembolsado", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)


#plt.suptitle("Variáveis Quantitativas Por Inadimplência", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.7)
#plt.savefig('Den.png', dpi = 300)
plt.show()

plt.figure(figsize = ((12,8)))

plt.subplot(1, 2, 1)
sns.kdeplot(data=df, x='Score Bureau' , hue="Inadimplencia", multiple="stack", ec='black', palette
plt.title("Score Bureau", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)


plt.subplot(1, 2, 2)
sns.kdeplot(data=df, x='Tempo Bancarizacao', hue="Inadimplencia", multiple="stack", ec='black', pa
plt.title("Tempo de Bancarização", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)


#plt.suptitle("Variáveis Quantitativas Por Inadimplência", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.7)
#plt.savefig('Den2.png', dpi = 300)
plt.show()

plt.figure(figsize = ((12,8)))

plt.subplot(2, 2, 1)
sns.countplot(data=df, x='Qt Consultas Feitas', hue="Inadimplencia", ec='black', palette = "Blues'
plt.title("Consultas Feitas Para Empréstimos", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
plt.ylabel("Contagem", fontsize = 15)
plt.legend(loc='upper right')

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='Qt Emprestimos Tomados', hue="Inadimplencia", ec='black', palette = "Blu
plt.title("Quantidade de Empréstimos Tomados", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
```

```
plt.ylabel("Contagem", fontsize = 15)
plt.legend(loc='upper right')

plt.subplot(2, 2, 3)
sns.countplot(data=df, x='Qt Novos Emprestimos 6M', hue="Inadimplencia", ec='black', palette = "Blues",
plt.title("Empréstimos Tomados nos Últimos 6 Meses", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
plt.ylabel("Contagem", fontsize = 15)
plt.legend(loc='upper right')

plt.subplot(2, 2, 4)
sns.countplot(data=df, x='Qt Restritivos 6M', hue="Inadimplencia", ec='black', palette = "Blues", lw =
plt.title("Restritivação nos Últimos 6 Meses", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
plt.ylabel("Contagem", fontsize = 15)
plt.legend(loc='upper right')


#plt.suptitle("Informações de Empréstimos e Restritivos", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.7)
#plt.savefig('Rest.png', dpi = 300)
plt.show()

plt.figure(figsize = ((12,8)))

plt.subplot(2, 2, 1)
sns.countplot(data=df, x='Qt Emprestimos Ativos', hue="Inadimplencia", ec='black', palette = "Blues", l
plt.title("Quantidade de Empréstimos Ativos", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
plt.ylabel("Contagem", fontsize = 15)

plt.subplot(2, 2, 2)
sns.countplot(data=df, x='Qt Restritivos Ativos', hue="Inadimplencia", ec='black', palette = "Blues", l
plt.title("Quantidade de Restritivos Ativos", fontsize = 20)
plt.xlabel("Quantidade", fontsize = 15)
plt.ylabel("Contagem", fontsize = 15)

plt.subplot(2, 2, 3)
sns.kdeplot(data=df, x='Vl Emprestimos Tomados', hue="Inadimplencia", multiple="stack", ec='black', pal
plt.title("Valor Total de Emprestimos Tomados", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)

plt.subplot(2, 2, 4)
sns.kdeplot(data=df, x='Vl Emprestimos Ativos', hue="Inadimplencia", multiple="stack", ec='black', pale
plt.title("Valor de Emprestimos Ativos", fontsize = 20)
plt.xlabel("Valor", fontsize = 15)
plt.ylabel("Distribuição", fontsize = 15)
```

```
#plt.suptitle("Informações de Empréstimos e Restritivos", fontsize = 25)
plt.subplots_adjust(left=0.05, right=1, top=0.9, wspace=0.5, hspace=0.7)
#plt.savefig('Rest2.png', dpi = 300)
plt.show()
```

.5 Treino e Teste

```
#Divindo os meus dados em treino e teste
y = df_model["Inadimplencia"] #var resp
X = df_model[vars_model] #vars preds

df_train, df_test, X_train, X_test, y_train, y_test = train_test_split(df_model[vars_model],
                                                                       X,
                                                                       y,
                                                                       test_size=0.30,
                                                                       random_state=42,
                                                                       stratify=y)

idx_train = df_train.index
idx_test = df_test.index
```

.6 Árvore de Decisão

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier(random_state = 42,
                             criterion= "gini",
                             max_depth = 3,
                             min_samples_leaf = 0.1,
                             class_weight = "balanced"
                            )

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)


print ('acc: ', accuracy_score(y_test,y_pred))
print ('recall: ', recall_score(y_test,y_pred))
print ('precision: ', precision_score(y_test,y_pred))
print ('roc: ', roc_auc_score(y_test,y_pred))
print ('sqrt: ', mean_squared_error(y_test,y_pred))
print ('confusion matrix: \n ',
```

```
        pd.DataFrame(confusion_matrix(y_test,y_pred)))


fig = plt.figure(figsize=(20,12))
tree.plot_tree(clf,
               feature_names = X_train.columns,
               class_names=['0', '1'],
               impurity = False,
               proportion = True,
               fontsize=10,
               filled = True
                )
```

.7 Padronização dos Dados

```
normalizador = StandardScaler()

df[['Vl Desembolsado', 'Vl Custo do Ativo', 'LTV', 'Idade',
    'Score Bureau', 'Qt Emprestimos Tomados',
    'Qt Emprestimos Ativos', 'Qt Restritivos Ativos',
    'Vl Emprestimos Ativos', 'Vl Emprestimos Tomados',
    'Qt Novos Emprestimos 6M', 'Qt Restritivos 6M', 'Prazo',
    'Tempo Bancarizacao', 'Qt Consultas Feitas']] = normalizador.fit_transform(df[['Vl Desembolsado', '
                                                                                   'Score Bureau', 'Qt
                                                                                   'Qt Emprestimos Ativ
                                                                                   'Vl Emprestimos Ativ
                                                                                   'Qt Novos Emprestimo
                                                                                   'Tempo Bancarizacao'
```

.8 Seleção de Variáveis

```
#Transformando as variáveis em categoricas
categoricas = ['Tp Trabalho','Fl Celular',
               'Fl Documento','Fl N Conta','Fl Titulo Eleitor','Fl CNH',
               'Fl Passaporte'] # lista das vars que são categoricas

for col in categoricas:
  df[col] = df[col].astype('category')

  vars_model = ['Vl Desembolsado', 'Vl Custo do Ativo', 'LTV', 'Idade', 'Tp Trabalho',
              'Fl Celular', 'Fl Documento', 'Fl N Conta', 'Fl Titulo Eleitor',
              'Fl CNH', 'Fl Passaporte', 'Score Bureau', 'Qt Emprestimos Tomados',
              'Qt Emprestimos Ativos', 'Qt Restritivos Ativos',
              'Vl Emprestimos Ativos', 'Vl Emprestimos Tomados',
```

```
                'Qt Novos Emprestimos 6M', 'Qt Restritivos 6M', 'Prazo',
                'Tempo Bancarizacao', 'Qt Consultas Feitas', 'Inadimplencia']

vars_rest_corr_remov = correlation_iv(df,
                                      vars_model,
                                      0.80,
                                      'Inadimplencia',
                                      10,
                                      0.00,
                                      0.04,
                                      12)

number_of_elements = len(vars_rest_corr_remov)

print(f'Variáveis Eliminadas por correlação de 80% (com outras variáveis de IV maior): {number_of_

vars_model = ['Vl Desembolsado', 'Vl Custo do Ativo', 'LTV', 'Idade', 'Tp Trabalho',
                'Fl Celular', 'Fl Documento', 'Fl N Conta', 'Fl Titulo Eleitor',
                'Fl CNH', 'Fl Passaporte', 'Score Bureau',
                'Qt Emprestimos Ativos', 'Qt Restritivos Ativos',
                'Vl Emprestimos Ativos', 'Vl Emprestimos Tomados',
                'Qt Novos Emprestimos 6M', 'Qt Restritivos 6M', 'Prazo',
                'Tempo Bancarizacao', 'Qt Consultas Feitas', 'Inadimplencia']

vars_rest_corr_remov = correlation_iv(df,
                                      vars_model,
                                      0.80,
                                      'Inadimplencia',
                                      10,
                                      0.00,
                                      0.04,
                                      12)

number_of_elements = len(vars_rest_corr_remov)

print(f'Variáveis Eliminadas por correlação de 80% (com outras variáveis de IV maior): {number_of_

iv = get_iv_df(df[vars_model] ,
            [], 1.0, 'Inadimplencia', q=10, threshold_iv=0.0)

display(iv)
display(iv[(iv['iv']>0.004)])
len(iv['iv']>0.004)
```

.9 Ajuste do Modelo de Regressão Logística

```
logistic_regression= LogisticRegression()
logistic_regression.fit(X_train,y_train)
y_pred=logistic_regression.predict(X_test)

logistic_regression.class_weight

select_feature = logistic_regression.feature_names_in_
select_feature

prob_bom = logistic_regression.predict_proba(df[select_feature])[:,0]

x = prob_bom
x_minmax = (x - x.min())/(x.max()-x.min())

df['score_model_logistic'] = x_minmax * 1000

evaluate_model(df,
               X_train,
               X_test,
               y_train,
               y_test,
               idx_train,
               idx_test,
               'score_model_logistic',
               'Inadimplencia',
               logistic_regression,
               select_feature)
plt.savefig('ROC_Logit_reg.png', dpi = 300)

#Matriz de Confusão
conf_matrix(y_test, y_pred)
plt.xlabel('Classe Predita')
plt.ylabel('Classe Esperada')
plt.savefig('MC_Logit_reg.png', dpi = 300)
```

.10 Ajuste do Modelo de Regressão Logística com Balanceamento

```
lr = LogisticRegression(solver='newton-cg')

#Setting the range for class weights
weights = np.linspace(0.0,0.99,200)
```

```
#Creating a dictionary grid for grid search
param_grid = {'class_weight': [{0:x, 1:1.0-x} for x in weights]}

#Fitting grid search to the train data with 5 folds
gridsearch = GridSearchCV(estimator= lr,
                          param_grid= param_grid,
                          cv=StratifiedKFold(),
                          n_jobs=-1,
                          scoring='f1',
                          verbose=2).fit(X_train, y_train)

#Ploting the score for different values of weight
sns.set_style('whitegrid')
plt.figure(figsize=(12,8))
weigh_data = pd.DataFrame({ 'score': gridsearch.cv_results_['mean_test_score'], 'weight': (1- weig
sns.lineplot(weigh_data['weight'], weigh_data['score'])
plt.xlabel('Peso para a classe inadimplente')
plt.ylabel('F1 score')
plt.xticks([round(i/10,1) for i in range(0,11,1)])
plt.title('Pontuação para diferentes pesos de classe', fontsize=25)
plt.savefig('PontuaçãoF1', dpi = 300)

#importing and training the model
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(solver='newton-cg', class_weight={0: 0.23, 1: 0.77})
lr.fit(X_train, y_train)

# Predicting on the test data
pred_test = lr.predict(X_test)

#Calculating and printing the f1 score
f1_test = f1_score(y_test, pred_test)
print('The f1 score for the testing data:', f1_test)

#Ploting the confusion matrix
conf_matrix(y_test, y_pred)
plt.xlabel('Classe Esperada')
plt.ylabel('Classe Predita')
plt.savefig('MC_Logit_reg_LR.png', dpi = 300)

prob_bom = lr.predict_proba(df[select_feature])[:,0]

x = prob_bom
x_minmax = (x - x.min())/(x.max()-x.min())

df['score_model_lr'] = x_minmax * 1000

evaluate_model(df,
               X_train,
```

```
                X_test,
                y_train,
                y_test,
                idx_train,
                idx_test,
                'score_model_lr',
                'Inadimplencia',
                lr,
                select_feature)
plt.savefig('ROC_Logit_reg_LR.png', dpi = 300)
```

.11 Árvore de Regressão Para Quebras do Score Obtido

```
def fx_decision_tree(df, feature, target, min_samples_leaf=1):
    X_tree = df[feature].to_numpy().reshape(-1, 1)
    y_tree = df[target]

    clf = DecisionTreeClassifier(criterion='gini',
#                                  max_depth=max_depth,
                                 min_samples_leaf=min_samples_leaf,
                                 random_state = 42)
    clf.fit(X_tree, y_tree)

    plt.figure(figsize=(20,12))
    plot_tree(clf, filled=True, proportion=True)

    return plt.show()

fx_decision_tree(df.dropna(subset = ['score_model_lr']),'score_model_lr','Inadimplencia',
                 int(df.dropna(subset=['score_model_lr']).shape[0]*0.1))

fx_tree = [0,
           372,
           432,
           462,
           502,
           554,
           614,
           692,
           #np.inf,
           1000]

df['fx_score_model_lr'] = pd.cut(df['score_model_lr'], bins=fx_tree)
```

```
def tab_freq_abs_lin(df, var_1, var_2):
    df_tab_freq_abs_lin = round(pd.crosstab(df[var_1],
                                    df[var_2],
                                    margins=True,
                                    normalize='index')*100,2)
    return df_tab_freq_abs_lin


def tab_freq_abs_col(df, var_1, var_2):
    df_tab_freq_abs_col = round(pd.crosstab(df[var_1],
                                    df[var_2],
                                    margins=True,
                                    normalize='columns')*100,2)
    return df_tab_freq_abs_col
def plot_bin_bad(df, var_bin, target):
    fig, ax1 = plt.subplots(figsize=(12,6))
    sns.set()
    sns.set_theme(style="white")
    sns.set_style("white")
    tab_freq_abs_col(df, var_bin, target)['All'].plot(kind='bar',
                                                      color=['#87CEEB'], ax=ax1,
                                                      ylabel ='Volume por Categorias - Porcentagem d
                                                       xlabel = 'Pontuação Obtida no Ajuste do Model
    ax2 = ax1.twinx()
    tab_freq_abs_lin(df, var_bin, target)[:-1][1].plot(color ='black',
                                                      ax=ax2,
                                                      ylabel='Probabilidade de Inadimplência')

    plt.setp(ax1.xaxis.get_majorticklabels(), rotation=45)

    return plt.show()
```