Victor Damião Gontijo Mourão

Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)

Brasília

2022

Victor Damião Gontijo Mourão

Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)

Dissertação apresentada ao Curso de Mestrado Acadêmico em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Universidade de Brasília - UnB

Faculdade de Administração Contabilidade e Economia - FACE

Departamento de Economia - ECO

Programa de Pós-Graduação

Orientador: Prof. Dr. Daniel Oliveira Cajueiro

Brasília

2022

Victor Damião Gontijo Mourão
Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)/ Victor Damião Gontijo Mourão. – Brasília, 2022-
44p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Daniel Oliveira Cajueiro

Dissertação (Mestrado) – Universidade de Brasília - UnB
Faculdade de Administração Contabilidade e Economia - FACE
Departamento de Economia - ECO
Programa de Pós-Graduação, 2022.

1. Aprendizagem de máquinas. 2. Métodos de aprendizado supervisionado. 3. Risco de crédito. II. Universidade de Brasília. III. Faculdade de Administração, Contabilidade e Economia - FACE. IV. Departamento de Economia IV. Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)

Victor Damião Gontijo Mourão

Estudo Comparativo entre Técnicas de Machine Learning para Classificação do Tomador PJ – MPE (Micro e Pequenas Empresas)

Dissertação apresentada ao Curso de Mestrado Acadêmico em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Trabalho aprovado. Brasília, 30 de maio de 2022:

Prof. Dr. Daniel Oliveira Cajueiro
Orientador

Prof.ª Dra. Marina Delmondes de Carvalho Rossi
Membro interno

Prof. Dr. Herbert Kimura
Membro externo

Brasília
2022

Dedico este trabalho primeiramente à Deus,
à minha esposa Simone e aos meus pais José e Maria.

Agradecimentos

Agradeço ao meu orientador, Professor Doutor Daniel Oliveira Cajueiro, pela dedicação, disponibilidade, encorajamento e por todas as sugestões e conselhos, fundamentais para a conclusão deste trabalho.

Aos meus pais, por todo amor e por colocar a educação dos filhos como um dos principais pilares da vida.

À minha esposa, pelo apoio incondicional e por estar ao meu lado, dando a força necessária para enfrentar todos os desafios.

Aos amigos Taína, Sheilla, Sabrina, Jonathan, Blay, Allisson e Christian, pela ajuda na construção deste trabalho.

À todos os professores deste curso, pelo conhecimento repassado.

"Tudo na vida é gerenciamento de risco,
não sua eliminação"
(Walter Wriston)

Resumo

A literatura tem mostrado que as técnicas de *machine learning* são bastante adequadas no contexto de risco de crédito e muitos pesquisadores têm obtido bons resultados para previsão da inadimplência. Nesse sentido, esta dissertação teve como objetivo a realização de um estudo empírico, utilizando métodos de aprendizado supervisionado para a classificação de tomadores de crédito. O trabalho foi desenvolvido a partir de uma base de dados real, fornecida por um dos maiores bancos do Brasil, com informações de micro e pequenas empresas contratantes de empréstimos no ano de 2020, período em que a pandemia da COVID-19 impactava o cenário econômico. Utilizamos 3 técnicas de *machine learning*: Regressão Logística, *Random Forest* e *Gradient Boosting*. E para aumentar o desempenho dos algoritmos, diante de uma base de dados desbalanceada, utilizamos 3 técnicas de balanceamento: *NearMiss*, SMOTE - *Synthetic Minority Over-sampling Technique* e SMOTEENN - combinação do SMOTE com ENN - *Edited Nearest Neighbours*. Como medida de sucesso, buscamos o aumento da AUC - Área Sob a Curva ROC, e, de acordo com os resultados, o algoritmo que apresentou maior AUC foi o *Random Forest* com o balanceamento SMOTE, atingindo resultado de 79,16%, e a menor AUC foi do algoritmo Regressão Logística sem o balanceamento da base de treino, atingindo resultado de 67,99%.

Palavras-chave: aprendizagem de máquinas, métodos de aprendizado supervisionado e risco de crédito.

Abstract

The literature has shown that machine learning techniques are quite suitable in the context of credit risk and many researchers have obtained good results for default prediction. In this sense, this dissertation aimed to carry out an empirical study, using supervised learning methods to classify borrowers. The work was developed from a real database, provided by one of the largest banks in Brazil, with information on micro and small companies contracting loans in 2020, a period in which the COVID-19 pandemic impacted the economic scenario. We used three machine learning techniques: Logistic Regression, Random Forest and Gradient Boosting. And to increase the performance of the algorithms, in front of an unbalanced database, we used three balancing techniques: NearMiss, SMOTE - Synthetic Minority Over-sampling Technique and SMOTEENN - combination of SMOTE with ENN - Edited Nearest Neighbors. As a measure of success, we sought to increase the AUC - Area Under the ROC Curve, and, according to the results, the algorithm that presented the highest AUC was Random Forest with SMOTE balance, reaching a result of 79.16%, and the lowest AUC was from the Logistic Regression algorithm without balancing the training base, reaching a result of 67.99%.

Keywords: machine learning, supervised learning methods and credit risk.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

O mundo encontra-se em constante mudança, cada dia mais rápida em razão dos avanços tecnológicos, e de tempos em tempos vemos o cenário sendo impactado por crises, seja ela econômica, financeira ou até sanitária, como a que vivemos hoje: a pandemia da COVID-19. A ocorrência dessas crises provoca diversos impactos, muitas vezes negativos, na vida financeira das pessoas e das empresas, modificando a capacidade de pagamento e em casos extremos levando à falência. Esse lado negativo da crise pesa sobremaneira na concessão do crédito e no seu custo, em virtude do aumento do risco. Portanto, é necessário o aprimoramento contínuo dos modelos de avaliação do risco e concessão do crédito, buscando constantemente o aumento progressivo da acurácia.

As instituições financeiras têm observado um aumento na quantidade de informações disponíveis e melhorias nas técnicas de avaliação, cada vez mais acuradas para concessão e recuperação do crédito. Desta forma, a utilização de métodos mais sofisticados e de técnicas de *machine learning* têm sido mais comum para a classificação dos tomadores de crédito, a fim de proporcionar um ambiente dinâmico e contribuir para tomadas de decisões mais rápidas.

O objetivo deste trabalho é comparar alguns dos métodos de *machine learning* para classificação do tomador, a fim de verificar e apresentar o algoritmo com o melhor desempenho para classificar o risco de crédito com base nos dados de uma amostra de clientes MPE.

De forma resumida, as etapas do trabalho são: desenvolver os modelos de classificação com a utilização de técnicas de *machine learning*; comparar os modelos desenvolvidos, verificando a qualidade e capacidade de previsão; e propor um modelo para classificação do tomador.

Neste trabalho, utilizamos as seguintes técnicas de *machine learning*: Regressão Logística, *Random Forest* e *Gradient Boosting*. Outros trabalhos demonstram que estas técnicas são bastante adequadas para lidar com conjuntos de dados desbalanceados, situação presente na nossa base de dados. No trabalho de Brown e Mues (2012), realizado para comparar algoritmos de classificação para conjuntos de dados desbalanceados, teve como uma de suas conclusões que as técnicas de *Gradient Boosting* e *Random Forest* tiveram bons desempenhos ao lidar com amostras onde há grande desequilíbrio de classes, sugerindo que a capacidade do *Random Forest* e do *Gradient Boosting*, de concentrar-se em variáveis "locais", é útil para melhorar o desempenho. A regressão logística apresentou resultados razoáveis e os algoritmos árvore de decisão (C4.5), Análise Discriminante Quadrática (QDA) e LS-SVM, não obtiveram bons resultados em conjuntos de dados desbalanceados.

Para lidar com o problema de classes desbalanceadas, são propostas várias abordagens, por exemplo, técnicas de *oversampling*, como o SMOTE proposto por Chawla et al. (2002), técnicas de *undersampling*, como o NearMiss proposto por Mani e Zhang (2003) e a combinação de *oversampling* com *undersampling*, como o SMOTEENN proposto por Batista, Prati e Monard (2004). Contudo, essas soluções podem gerar maior custo computacional e maior período de treinamento.

O artigo de Goyal, Rathore e Sharma (2021) ressalta que o desequilibrio de classes tem sido um dos problemas mais complexos nos campos de *machine learning* e *data mining*, e para lidar com o desbalanceamento dos dados e melhorar os resultados, os autores utilizaram o algoritmo *Random Forest* combinado com a técnica de *oversampling* SMOTE.

Neste sentido, avaliamos o resultado da combinação do SMOTE, NearMiss e SMOTEEN com os 3 algoritmos selecionados.

Nosso trabalho contribui para a literatura com o desenvolvimento e análise de resultados obtidos a partir da aplicação de técnicas de *machine learning* em uma base de dados real, composta por micro e pequenas empresas de um dos maiores bancos do Brasil, no período em que a pandemia da Covid-19 causava forte impacto nas finanças das empresas brasileiras.

Nossos resultados mostram que as 3 técnicas investigadas são adequadas para o problema deste trabalho e o algoritmo *Random Forest* combinado com o balanceamento SMOTE apresentou o melhor desempenho.

O trabalho está dividido em 5 capítulos: o capítulo 1 traz a introdução; o capítulo 2 apresenta a composição da base de dados e o tratamento que realizamos nesta base; o capítulo 3 demonstra a metodologia que utilizamos para o desenvolvimento do trabalho e como os dados foram avaliados; o capítulo 4 traz os resultados que obtivemos com a aplicação dos algoritmos de *machine learnig*; e por fim, no capítulo 5 são apresentadas as conclusões e as sugestões de trabalhos futuros.

2 Dados

Os dados utilizados neste trabalho pertencem à uma instituição financeira do segmento S1[1], e com o objetivo de manter o sigilo das informações, não foram utilizados dados que possibilitem a identificação do cliente.

A base de dados é composta por 36.311 contratos (R$3,336 bilhões) pactuados em 2020 e vinculados à micro e pequenas empresas com empréstimo ativo junto à instituição financeira.

Destes, 33.671 contratos (R$3,178 bilhões) foram classificados como adimplentes e 2.640 contratos (R$157,9 milhões) foram classificados como inadimplentes. Foram considerados inadimplentes, os contratos com atraso a partir de 90 dias.

Realizamos diversas análises na base de dados para verificar as informações disponíveis e avaliar as possíveis variáveis para utilização nos modelos.

A fim de garantir a consistência dos dados, realizamos um pré-processamento com avaliação detalhada da base, para mitigar a ocorrência de problemas em razão de eventuais inconsistências nos dados.

Excluímos da base as variáveis com ausência de informações, com valores únicos (similar à uma constante) e com dados irrelevantes e/ou inconsistentes.

Após estes tratamentos, selecionamos, para compor os modelos, 9 variáveis preditoras capazes de explicar a variável dependente (Inadimplência). Ressaltamos que todas as informações das variáveis são do momento da contratação, para retratar a situação do contrato naquela época.

A tabela 1 apresenta o nome e a descrição das variáveis independentes, selecionadas para o trabalho.

Para cada variável foram criadas faixas de agrupamento, com o objetivo de caracterizar melhor a composição da base. A variável Segmento foi dividida em 3 faixas (EE, ES e EF). A faixa EE refere-se aos contratos vinculados às empresas MEI - Microempreendedor Individual, com faturamento de R$0,00 a R$81.000,00; a faixa ES refere-se às micro empresas com faturamento de R$0,00 a R$360.000,00 e a faixa EF refere-se às empresas de pequeno porte com faturamento de R$360.000,01 a R$4.800.000,00. A tabela 2 apresenta a distribuição nas faixas.

A variável Prazo foi dividida em 4 faixas (Curto, Médio, Longo e Muito Longo). A faixa Curto refere-se aos contratos pactuados com prazo de até 1.095 dias; a faixa Médio

[1] O segmento S1 é composto por instituições financeira que tenham porte maior ou igual a 10% do PIB ou atividade internacional relevante.

Tabela 1 – Variáveis selecionadas

Nome da variável	**Descrição**
Segmento	Segmento enquadrado a partir do faturamento
Prazo	Prazo do contrato
Valor_Contrato	Valor do contrato
Rating_Cliente	*Rating* do cliente na época da contratação
Rating_Contrato	*Rating* do contrato na época da contratação
Garantia_Real	Se o contrato possui garantia real
Renegociação	Se o cliente tinha renegociação na época da contratação
Idade_Empresa	Tempo de constituição da empresa
Dívida_Faturamento	Relação entre o valor do contrato e o faturamento (12 meses)

Fonte: Elaborado pelo autor (2022).

Tabela 2 – Variável Segmento

Faixa	**Qtd. Contrato**	**Qtd. Inadimplente**
EE	656	112
ES	11.350	1.167
EF	24.305	1.361

Fonte: Elaborado pelo autor (2022).

refere-se aos contratos pactuados com prazo de 1.096 a 1.460 dias; a faixa Longo refere-se aos contratos pactuados com prazo de 1.461 a 1.825 dias e a faixa Muito Longo refere-se aos contratos pactuados com prazo acima de 1.825 dias. A tabela 3 apresenta a distribuição nas faixas.

Tabela 3 – Variável Prazo

Faixa	**Qtd. Contrato**	**Qtd. Inadimplente**
Curto	5.148	899
Médio	23.583	1.144
Longo	7.240	469
Muito Longo	340	128

Fonte: Elaborado pelo autor (2022).

A variável Valor_Contrato foi dividida em 5 faixas (Muito Baixo, Baixo, Médio, Alto e Muito Alto). A faixa Muito Baixo refere-se aos contratos com valor de até R$15.000,00; a faixa Baixo refere-se aos contratos com valor de R$15.000,01 até R$30.000,00; a faixa Médio refere-se aos contratos com valor de R$30.000,01 até R$90.000,00; a faixa Alto refere-se aos contratos com valor de R$90.000,01 até R$200.000,00 e a faixa Muito Alto refere-se aos contratos com valor acima de R$200.000,01. A tabela 4 apresenta a distribuição nas faixas.

A variável Rating_Cliente foi dividida em 9 faixas, desde a categoria AA, melhor

Tabela 4 – Variável Valor_Contrato

Faixa	Qtd. Contrato	Qtd. Inadimplente
Muito Baixo	2.905	626
Baixo	4.667	555
Médio	17.435	923
Alto	7.999	435
Muito Alto	3.305	101

Fonte: Elaborado pelo autor (2022).

conceito, até a categoria H, pior conceito. A tabela 5 apresenta a distribuição nas faixas.

Tabela 5 – Variável Rating_Cliente

Faixa	Qtd. Contrato	Qtd. Inadimplente
AA	11.061	241
A	11.297	536
B	4.498	164
C	5.586	495
D	2.186	350
E	698	172
F	435	217
G	86	73
H	464	392

Fonte: Elaborado pelo autor (2022).

A variável Rating_Contrato foi dividada em 9 faixas, desde a categoria AA, melhor conceito, até a categoria H, pior conceito. A tabela 6 apresenta a distribuição nas faixas.

Tabela 6 – Variável Rating_Contrato

Faixa	Qtd. Contrato	Qtd. Inadimplente
AA	25.295	644
A	2.514	92
B	2.926	269
C	3.492	559
D	1.031	297
E	347	184
F	256	190
G	69	60
H	381	345

Fonte: Elaborado pelo autor (2022).

A variável Garantia_Real foi divida em 2 faixas (Sim e Não). A faixa Sim refere-se aos contratos que possuem garantia real e a faixa Não aos contratos que não possuem garantia real. A tabela 7 apresenta a distribuição nas faixas.

Tabela 7 – Variável Garantia_Real

Faixa	**Qtd. Contrato**	**Qtd. Inadimplente**
Sim	4.123	689
Não	32.188	1.951

Fonte: Elaborado pelo autor (2022).

A variável Renegociação foi dividida em 2 faixas (Sim e Não).A faixa Sim refere-se aos contratos cujo o cliente possuía renegociação ativa à época da contratação e a faixa Não refere-se aos contratos cujo o cliente não possuía renegociação ativa na data da contratação. A tabela 8 apresenta a distribuição nas faixas.

Tabela 8 – Variável Renegociação

Faixa	**Qtd. Contrato**	**Qtd. Inadimplente**
Sim	1.269	997
Não	35.042	1.643

Fonte: Elaborado pelo autor (2022).

A variável Idade_Empresa foi divida em 4 faixas (0 a 10, 11 a 15, 16 a 20, Acima de 20). A faixa 0 a 10 refere-se aos contratos vinculados às empresas com até 10 anos de constituição; a faixa 11 a 15 refere-se aos contratos vinculados às empresas de 11 a 15 anos de constituição; a faixa 16 a 20 refere-se aos contratos vinculados às empresas de 16 a 20 anos de constituição e a faixa Acima de 20 refere-se aos contratos vinculados às empresas com mais de 20 anos de constituição. A tabela 9 apresenta a distribuição nas faixas.

Tabela 9 – Variável Idade_Empresa

Faixa	**Qtd. Contrato**	**Qtd. Inadimplente**
0 a 10	11.607	1.486
11 a 15	9.733	761
16 a 20	7.281	225
Acima de 20	7.690	168

Fonte: Elaborado pelo autor (2022).

A variável Dívida_Faturamento foi dividida em 4 faixas (Muito Baixo, Baixo, Médio, Alto e Muito Alto). A faixa Muito Baixo refere-se aos contratos cuja a relação entre o valor do contrato e o faturamento da empresa nos últimos 12 meses, era de até 5%; a faixa Baixo refere-se à relação de 5,01% a 15%; a faixa Médio refere-se à relação de 15,01% a 30%; a faixa Alto refere-se à relação de 30,01% a 60% e a faixa Muito Alto quando esta relação for acima de 60%. A tabela 10 apresenta a distribuição nas faixas.

Para verificar a correlação entre as variáveis, geramos uma matriz de correlação, esta matriz mede o grau de relação entre cada par de variáveis, os valores de correlação

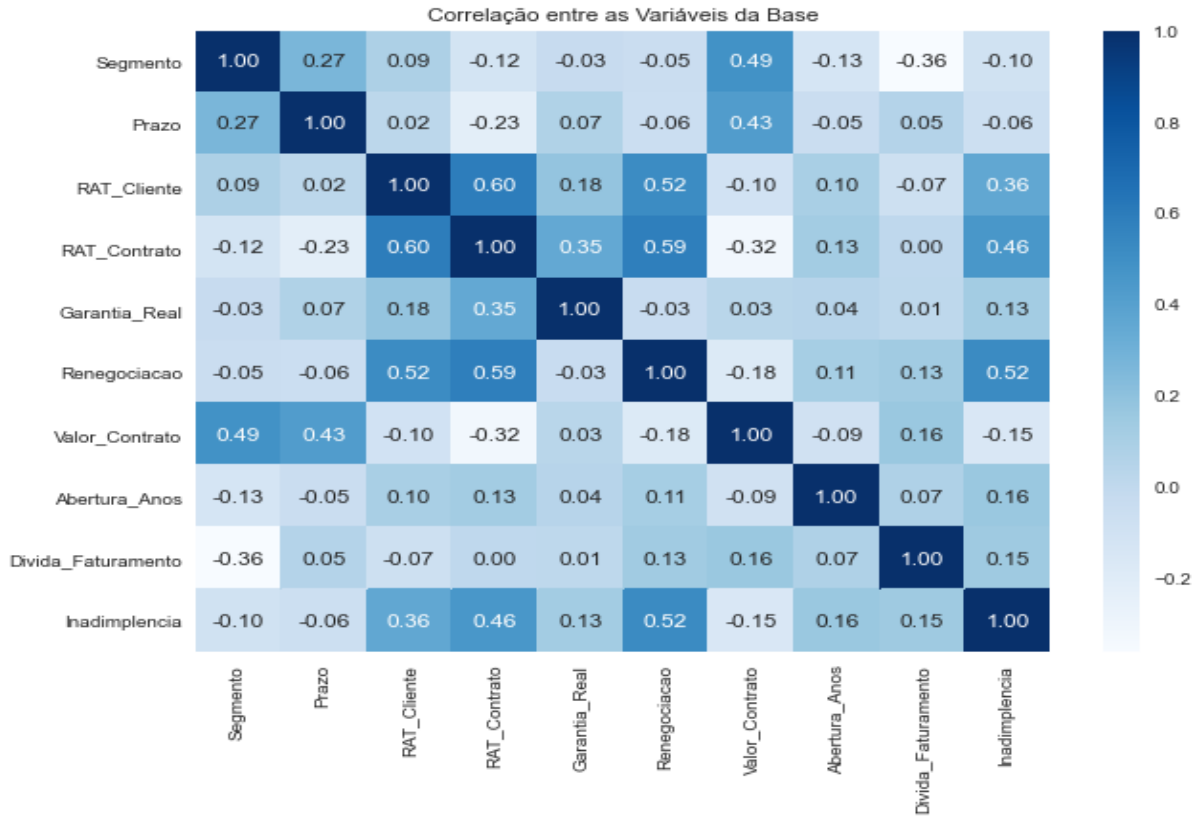
Tabela 10 – Variável Dívida_Faturamento

Faixa	Qtd. Contrato	Qtd. Inadimplente
Muito Baixo	7.561	305
Baixo	15.943	716
Médio	12.303	1.474
Alto	438	113
Muito Alto	66	32

Fonte: Elaborado pelo autor (2022).

podem variar de -1 a 1. A figura 1 apresenta o mapa de calor (matriz) com a correlação das variáveis.

Figura 1 – Correlação entre as variáveis



Fonte: Elaborado pelo autor (2022).

Realizamos a implementação dos modelos escolhidos em um software livre e como linguagem de programação utilizamos o Python e suas bibliotecas, alguns exemplos de bibliotecas que foram utilizadas são: Pandas, NumPy, Matplotlib, Seaborn e Scikit-Learn.

3 Metodologia

Este trabalho aborda problema de classificação com a utilização de métodos de aprendizado supervisionado. De acordo com Lee e Shin (2020), a classificação é o processo de identificação da categoria ou classe de uma observação. As categorias já são conhecidas para fins de treinamento e teste. Uma vez concluído o treino pode-se atribuir uma categoria a uma nova observação.

Apresentamos, neste capítulo, a metodologia utilizada para a construção de modelo classificador de *machine learning* que seja capaz de prever a classe do tomador de crédito. O capítulo está dividido em duas partes: mostramos na primeira parte o conjunto de modelos utilizados; e na segunda parte as ferramentas utilizadas para o balanceamento dos dados, escolha dos parâmetros e avaliação dos modelos.

3.1 Modelos

Para análise das técnicas de *machine learning* e suas aplicações no gerenciamento do risco de crédito, realizamos, primeiramente, pesquisas sobre o tema e selecionamos os principais artigos para análise, os quais serviram de base para o trabalho.

Para este trabalho selecionamos os modelos de Regressão Logística, *Random Forest* e o *Gradient Boosting*.

3.1.1 Regressão Logística

A regressão logística é uma abordagem estatística simplesmente paramétrica e tem sido considerada o padrão da indústria para *credit scoring*. Ela é usada para resolver problemas de classificação binária e problemas de regressão (BAO; LIANJU; YUE, 2019).

A regressão logística permite estimar a probabilidade de que um indivíduo E pertença ao grupo G1 (que denominaremos grupo evento). A probabilidade de que pertença ao grupo G2 é igual 1 menos esse valor. A escolha de G1 ou G2 como grupo evento é arbitrária e não afeta a classificação dos indivíduos. Neste texto, o grupo G1 é o grupo dos bons e o grupo G2 o grupo dos maus (SICSú, 2010).

Ainda de acordo com Sicsú (2010), o modelo logístico fundamenta-se na validade da relação:

$$\ln[\frac{P(bom)}{1-P(bom)}] = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + ... + \beta_p \times x_p \tag{3.1}$$

Denotando a função linear por Z, teremos:

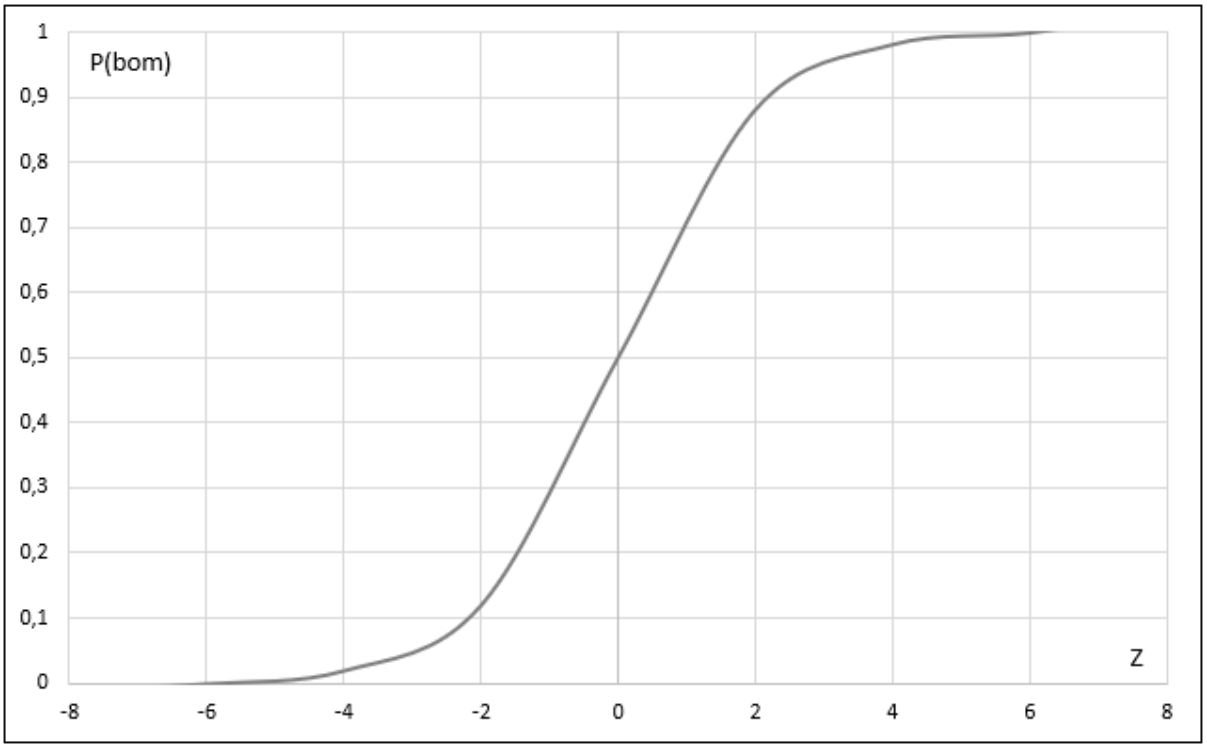
$$Z = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + ... + \beta_p \times x_p \Rightarrow \ln[\frac{P(bom)}{1 - P(bom)}] = Z \tag{3.2}$$

portanto,

$$P(bom) = \frac{e^Z}{1 + e^Z} = \frac{1}{1 + e^{-Z}} \tag{3.3}$$

A variação de P(bom) com Z pode ser vista na Figura 2.

Figura 2 – Curva logística



Fonte: Elaborado pelo autor (2022).

Note-se que:

- P(bom) é uma função de Z: quanto maior o valor de Z, maior será o valor de P(bom).
- O valor de P(bom) será utilizado para classificar um indivíduo como bom ou mau cliente. Se P(bom) >= k (em que k é um valor predeterminado), o indivíduo será classificado como bom; caso contrário, será classificado como mau.
- Para valores inferiores a Z = -6 ou superiores a Z = 6, a P(bom) é muito próxima de zero. Em particular, para Z = 6, temos P(bom) = 0,998 (99,8%) e para Z = -6 temo P(bom) = 0,002 (0,2%).

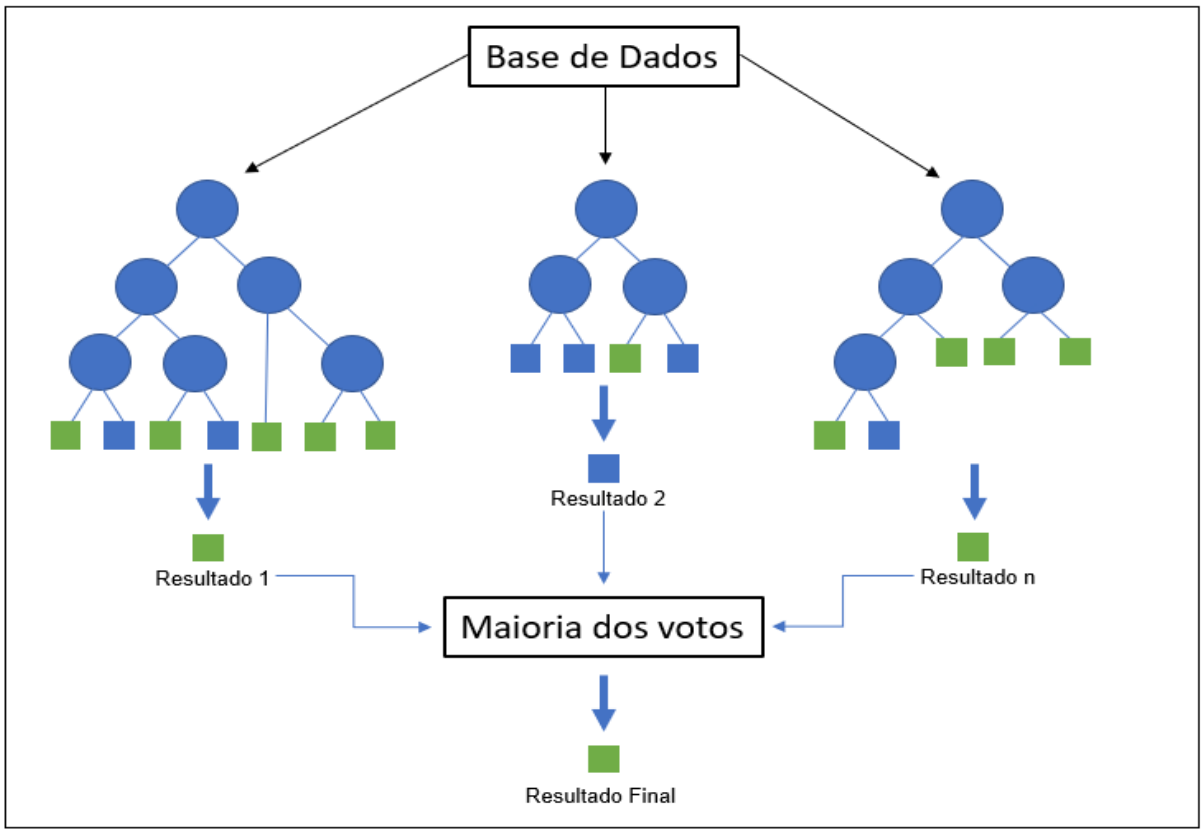
3.1.2 Random Forest

Random Forest é considerada uma técnica avançada de árvores de decisão, a ideia por trás desta técnica é combinar o *bagging* e a seleção de variáveis de forma aleatória para mesclar árvores de decisão individuais. O modelo de *Random Forest* usa a aleatoriedade em dois estágios: primeiro, seleciona aleatoriamente subconjuntos do conjunto de dados original; segundo, seleciona aleatoriamente, a partir das variáveis disponíveis, subconjuntos de variáveis. Desta forma, a correlação entre as árvores de decisão na floresta é reduzida. A decisão final é feita com base no procedimento de votação, onde a amostra de entrada será rotulada como a classe com a maioria dos votos (BAO; LIANJU; YUE, 2019).

De acordo com Breiman (2001), *Random Forest* é um classificador que consiste em uma coleção de classificadores estruturados em árvores $\{h(x, \Theta_k), k = 1, ...\}$ onde os $\{\Theta_k\}$ são vetores aleatórios independentes e distribuídos identicamente, e cada árvore lança um voto para a classe mais popular a partir do dado de entrada x.

Embora modelos baseados em árvores de decisão convencionais reduzam a possibilidade de *overfitting* ao escolher parâmetros que controlam sua profundidade e número de folhas, *Random Forest* também busca reduzir o *overfitting* de árvores tradicionais combinando diferentes árvores (ALBUQUERQUE; CAJUEIRO; ROSSI, 2022).

Figura 3 – Random Forest



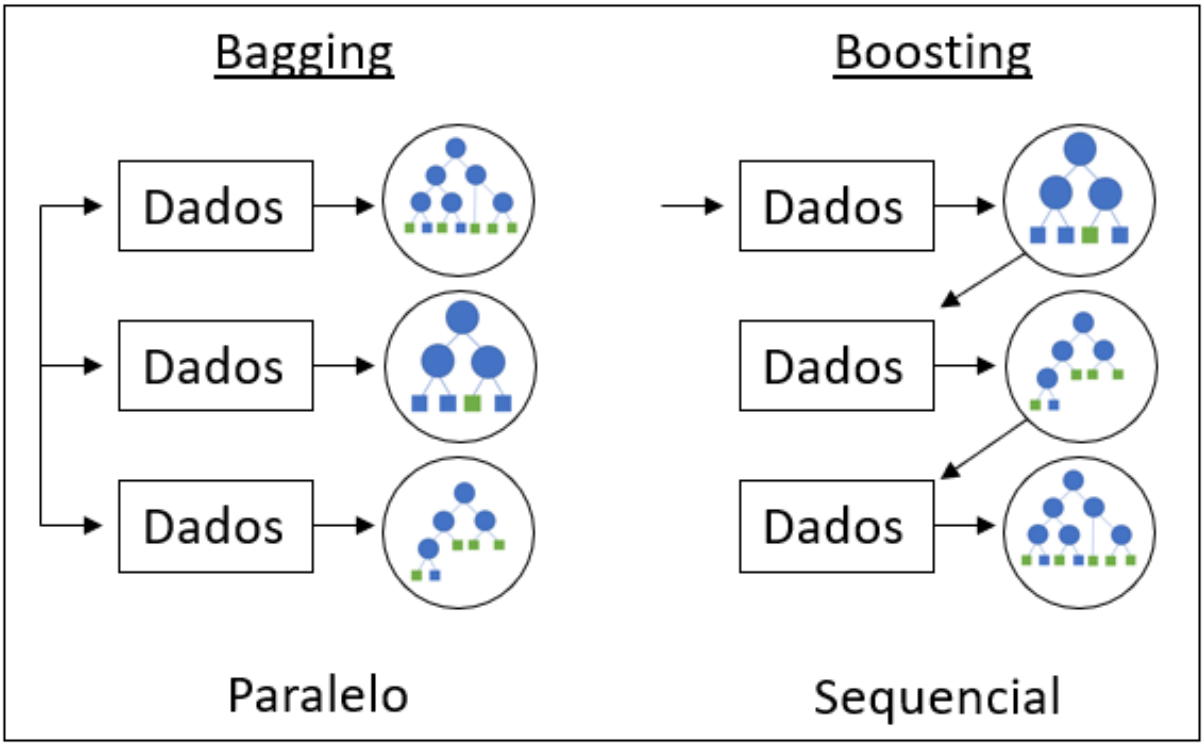
Fonte: Elaborado pelo autor (2022).

3.1.3 Gradient Boosting

Boosting é uma das ideias de aprendizagem mais poderosa introduzida nos últimos vinte anos. Ele foi originalmente projetado para problemas de classificação, mas, também pode ser estendido para regressão. A motivação para o *boosting* foi um processo que combinasse as saídas de vários classificadores "fracos" para produzir um poderoso "comitê" (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Em relação ao *bagging* (base para o algoritmo *Random Forest*), a principal diferença é que no *boosting* os modelos não são treinados de maneira independente, mas, são construídos especificamente para gerar aprendizados complementares, de forma sequencial, ou seja, não atribui o mesmo peso para todos os votos, pois depende do desempenho de cada modelo. A figura 4 dá a intuição dos 2 modelos.

Figura 4 – Bagging e Boosting



Fonte: Elaborado pelo autor (2022).

O *Gradient Boosting* visa minimizar a função de perda, calculando iterativamente o gradiente de acordo com o método padrão de gradiente descendente. Se falarmos sobre árvores de decisão como modelo base, então uma única árvore de decisão é construída em cada iteração para ajustar os gradientes negativos. O algoritmo minimiza a função de perda esperada usando árvores de decisão como modelo base e seus parâmetros incluem profundidade das árvores, a taxa de aprendizado, o número de iterações. Estes parâmetros são selecionados para fornecer uma alta generalização e precisão (KONSTANTINOV; UTKIN, 2021).

3.2 Avaliação dos Modelos

Para avaliação de um modelo preditivo é necessário ter um conjunto de dados independente, assim, inicialmente separamos de forma aleatória 70% dos dados da base para treinar o modelo e encontrar a melhor configuração de parâmetros e 30% para testar o desempenho do modelo. Essa proporção é amplamente utilizada na literatura, como ressalta García-Céspedes e Moreno (2022).

Utilizamos a mesma divisão dos dados para todos os modelos, a fim de garantir a comparabilidade entre os algoritmos.

Tendo em vista que os algoritmos, em geral, não apresentam bom desempenho para base de dados desbalanceadas, nós realizamos o balanceamento dos dados de treino utilizando 3 métodos diferentes, o *NearMiss*, o SMOTE e o SMOTEENN.

O *NearMiss*, referenciado no artigo de Mani e Zhang (2003), é um método de *undersampling*, que consiste em reduzir de forma aleatória a classe majoritária, porém este método realiza essa redução com base na distância, para diminuir a perda de variabilidade dos dados da classe.

O SMOTE, proposto por Chawla et al. (2002), refere-se a um método de *oversampling*, que consiste em duplicar dados aleatórios da classe minoritária, porém, ao invés de apenas duplicar os dados, o SMOTE gera dados sintéticos da classe minoritária a partir dos dados vizinhos, reduzindo assim o risco de *overfitting*.

O SMOTEENN, conforme Batista, Prati e Monard (2004), combina o SMOTE com o ENN, o SMOTE, como visto acima, gera dados sintéticos da classe minoritária e o ENN realiza uma limpeza, removendo dados de ambas as classes que tenham sido classificados incorretamente por seus três vizinhos mais próximos.

Além do balanceamento da base de treino, realizamos a otimização dos parâmetros. Para esta otimização, avaliamos 2 ferramentas, o GridSearchCV e o BayesSearchCV, ambas ferramentas realizam a buscar da melhor combinação de parâmetros por meio da validação cruzada.

A otimização de parâmetros é um processo para encontrar hiperparâmetros adequados para os modelos preditivos. Normalmente incorre em altos custos computacionais, devido à necessidade do processo de treinamento do modelo para determinar a eficácia de cada conjunto de possíveis valores de hiperparâmetros. A priori, não há garantia de que a otimização dos hiperparâmetros leve a um melhor desempenho (TRAN et al., 2020).

GridSearchCV é um método de otimização de parâmetro que divide os hiperparâmetros em grades com o mesmo comprimento em uma determinada faixa de um sistema de coordenadas. Cada ponto neste sistema de coordenadas representa um conjunto de hiperparâmetros, e então o GridSearchCV percorrer os pontos correspondentes a todas as

grades para verificar o desempenho do algoritmo e o ponto com o melhor desempenho é o que apresenta a melhor combinação de hiperparâmetros (TIAN et al., 2020).

A abordagem de Otimização Bayesiana, BayesSearchCV, acompanha os resultados de avaliações anteriores e usa estas avaliações para formar um modelo probabilístico de mapeamento de hiperparâmetros para melhorar a eficiência da busca. (ZHOU et al., 2021).

Portanto, observamos que o GridSearchCV faz uma busca exaustiva dos hiperparâmetros, o que demanda muito tempo de processamento e o BayesSearchCV em que pese não realizar esta busca exaustiva, realiza o mapeamento dos possíveis melhores pontos e utiliza as avaliações anteriores para melhorar a busca seguinte, entregando um bom resultado com menor tempo de processamento. Desta forma, optamos por utilizar o BayesSearchCV para otimização dos parâmetros. Configuramos a ferramenta para realizar a validação cruzada com k=10 e para garantir que cada *fold* tenha a mesma porcentagem de amostras para cada classe, utilizamos o StratifiedKFold. Buscamos o aumento da AUC como medida de sucesso. A AUC pode ter o valor máximo de 1 e quanto mais próximo de 1, significa que o modelo fornece um ajuste muito bom aos dados.

Medimos o desempenho dos classificadores com base nas seguintes métricas: *balanced accuracy*, *precision*, *recall*, *F-measure*, *specificity*, *G-mean* e na curva ROC e AUC.

Optamos por usar a métrica *balanced accuracy*, ao invés da *accuracy*, pois na presença de dados desbalanceados, a métrica *accuracy* pode não fornecer informações adequadas sobre o desempenho do classificador e acabar sendo enganosa. Como demonstrado no trabalho de Brodersen K. H.and Ong, Stephan e Buhmann (2010), a métrica *balanced accuracy* se torna mais adequada para trabalhos com bases desbalanceadas pois ela considera a média obtida nas duas classes de forma separada e igualitária. A fórmula da *balanced accuracy* é dada por $\frac{1}{2}(\frac{TP}{P} + \frac{TN}{N})$. Assim, se o classificador tiver desempenho igualmente bom nas duas classes, essa métrica será semelhante à *accuracy* convencional (número de previsões corretas dividido pelo número de previsões). Contudo, se a *accuracy* convencional for alta apenas porque o classificador fez uma boa classificação da classe majoritária e não desempenhou bem a previsão da classe minoritária, em um conjunto de dados desbalanceados, a *balanced accuracy* refletirá isso com um resultado baixo.

Outra métrica que utilizamos, e que é adequada para trabalhos com dados desbalanceados, é a *G-mean* - média geométrica, esta métrica segundo Zong, Huang e Chen (2013) fornece informações sobre a precisão obtida dentro de cada classe ao invés da precisão de todas as amostras. Para problema de classificação binária, a *G-mean* é dada por $\sqrt[2]{sensitivity * specificity}$.

4 Resultados

Neste capítulo apresentamos os resultados alcançados após o desenvolvimento das etapas descritas na metodologia.

Serão apresentadas 7 métricas de avaliação, contudo, por se tratar de estudo com base de dados desbalanceada, as principais métricas de avaliação foram a *Balanced Accuracy*, a *G-mean*, a *F-measure* - F1 e a AUC, ressaltamos que a métrica *Accuracy* foi substituída pela *Balanced Accuracy*, pois na presença de dados desbalanceados, esta métrica é mais adequada.

Inicialmente avaliamos o desempenho dos algoritmos com a base de dados de treino sem balanceamento. A tabela 11 apresenta o resultado alcançado.

Tabela 11 – Resultado sem o balanceamento da base de treino

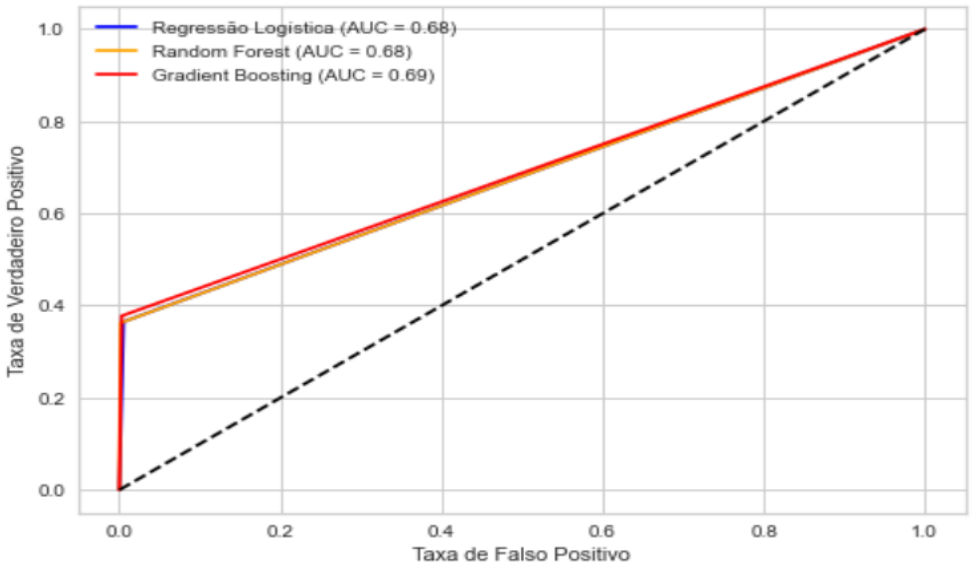
Algoritmo	B. Accuracy	Precision	Recall	F1	Specificity	G-mean	AUC
RL	67,99%	85,00%	36,49%	51,06%	99,49%	60,25%	67,99%
RF	68,04%	94,72%	36,24%	52,42%	99,84%	60,15%	68,04%
GB	68,73%	91,16%	37,75%	53,39%	99,72%	61,35%	68,73%

RL: Regressão Logística; RF: *Random Forest*; GB: *Gradient Boosting*

Fonte: Elaborado pelo autor (2022).

A figura 5 apresenta a curva ROC e a AUC a partir dos resultados obtidos sem o balanceamento da base de treino.

Figura 5 – Curva ROC - Resultado sem o balanceamento



Fonte: Elaborado pelo autor (2022).

Com a base de treino desbalanceada o algoritmo *Gradient Boosting* apresentou melhor desempenho nas métricas *Balanced Accuracy*, F1, *G-mean* e AUC, em segundo lugar ficou o algoritmo *Random Forest* e a Regressão Logística ficou um pouco abaixo e apresentou a menor AUC, atingindo 67,99%.

Observamos que os algoritmos não apresentaram um bom desempenho na métrica *Recall*, isso se dá em razão da base ser desbalanceada e o algoritmo aprender mais sobre a classe majoritária e acabar errando muito a classe minoritária.

Para mitigar esse problema realizamos o balanceamento da base de treino com 3 técnicas diferentes. A primeira avaliação de balanceamento foi com o *NearMiss*, uma técnica de *undersampling*, que consiste em reduzir de forma aleatória a classe majoritária, o *NearMiss* realiza essa redução com base na distância, para diminuir a perda de variabilidade dos dados da classe. A tabela 12 apresenta o resultado alcançado.

Tabela 12 – Resultado com o balanceamento NearMiss

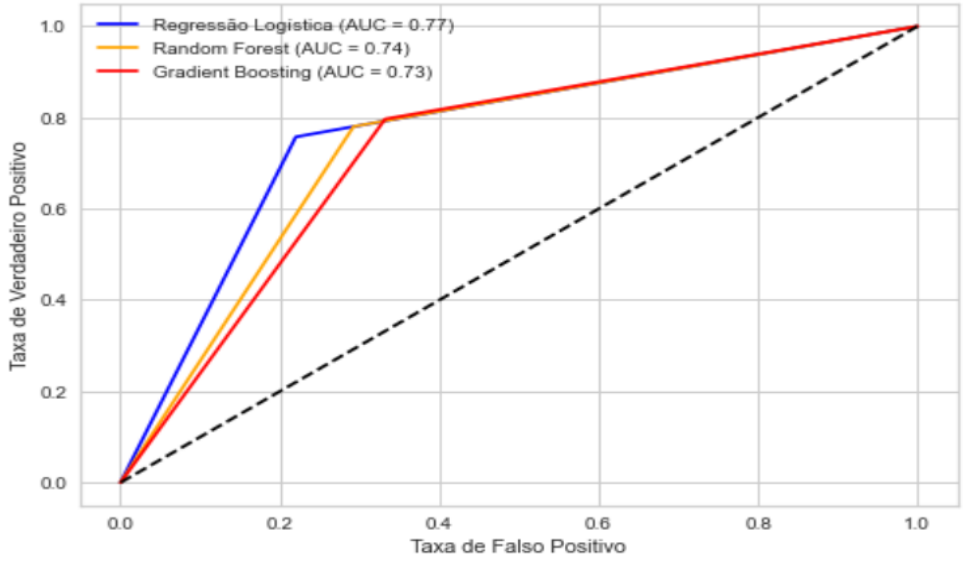
Algoritmo	B. Accuracy	Precision	Recall	F1	Specificity	G-mean	AUC
RL	76,88%	21,26%	75,76%	33,20%	78,00%	76,87%	76,88%
RF	74,39%	17,30%	78,03%	28,32%	70,76%	74,30%	74,39%
GB	73,23%	15,83%	79,67%	26,42%	66,80%	72,95%	73,23%

RL: Regressão Logística; RF: *Random Forest*; GB: *Gradient Boosting*

Fonte: Elaborado pelo autor (2022).

A figura 6 apresenta a curva ROC e a AUC a partir dos resultados obtidos com o balanceamento da base de treino com a técnica *NearMiss*.

Figura 6 – Curva ROC - Resultado com NearMiss



Fonte: Elaborado pelo autor (2022).

O balanceamento da base de treino utilizando a técnica *NearMiss* não apresentou resultados tão satisfatórios, em que pese ter obtido *Recall* bem superior ao observado no resultado sem balanceamento, a *precision* apresentou baixo desempenho. O algoritmo de Regressão Logística foi o que teve melhor desempenho nas 4 métricas principais, porém todos os algoritmos ficaram com baixo desempenho na métrica F1.

O desempenho ruim na métrica F1 se deu em razão da baixa precisão dos algoritmos após a aplicação da técnica de balanceamento de *undersampling*. Uma provável causa é que com a redução das amostras majoritária, mesmo o *NearMiss* utilizando a redução com base na distância, os algoritmos não conseguiram aprender o suficiente desta classe.

A segunda avaliação de balanceamento foi com o SMOTE, uma técnica de *oversampling*, que consiste em duplicar dados aleatórios da classe minoritária, o SMOTE gera dados sintéticos da classe minoritária a partir dos dados vizinhos, reduzindo o risco de *overfitting*. A tabela 13 apresenta o resultado alcançado.

Tabela 13 – Resultado com o balanceamento SMOTE

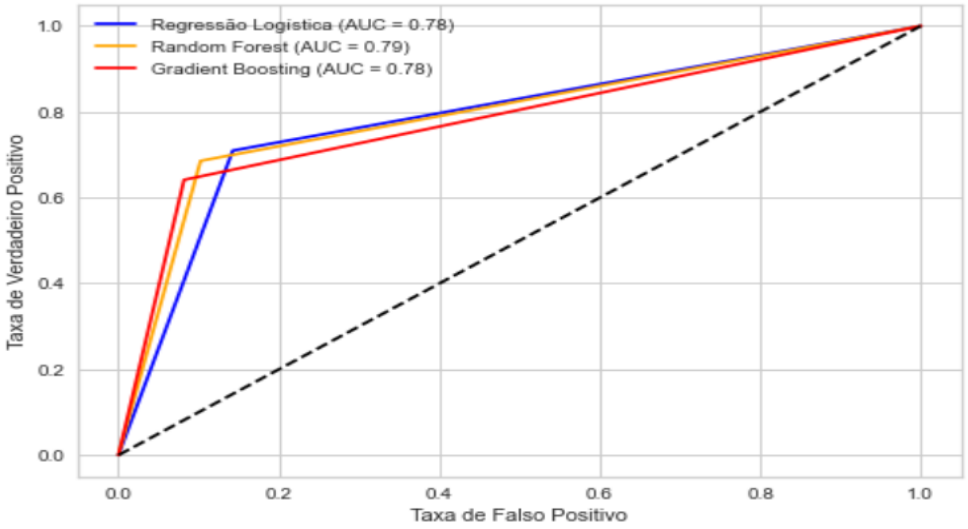
Algoritmo	B. Accuracy	Precision	Recall	F1	Specificity	G-mean	AUC
RL	78,36%	28,08%	70,96%	40,24%	85,75%	78,01%	78,36%
RF	79,16%	34,43%	68,56%	45,84%	89,76%	78,45%	79,16%
GB	77,98%	38,08%	64,14%	47,79%	91,82%	76,74%	77,98%

RL: Regressão Logística; RF: *Random Forest*; GB: *Gradient Boosting*

Fonte: Elaborado pelo autor (2022).

A figura 7 apresenta a curva ROC e a AUC a partir dos resultados obtidos com o balanceamento da base de treino com a técnica SMOTE.

Figura 7 – Curva ROC - Resultado com SMOTE



Fonte: Elaborado pelo autor (2022).

O balanceamento da base de treino utilizando a técnica SMOTE apresentou resultados superiores aos alcançados com a técnica NearMiss, além disso, apresentou melhor desempenho nas métricas *Balanced Accuracy*, *G-mean* e AUC comparado com a base sem balanceamento.

O algoritmo *Random Forest* apresentou os melhores resultados nas métricas *Balanced Accuracy*, *G-mean* e AUC, e, o *Gradient Boosting* apresentou o melhor desempenho na métrica F1. A Regressão Logística apresentou o melhor *recall* e desempenho próximo aos demais algoritmos nas demais métricas.

A terceira avaliação de balanceamento foi com o SMOTEENN, uma técnica combinada de *oversampling* (usando o SMOTE) e *undersampling* (usando o ENN - *Edited Nearest Neighbours*), que consiste em duplicar dados aleatórios da classe minoritária e reduzir dados de ambas as classes que tenham sido classificados incorretamente por seus três vizinhos mais próximos, a fim de obter uma amostra mais limpa. A tabela 14 apresenta o resultado alcançado.

Tabela 14 – Resultado com o balanceamento SMOTEENN

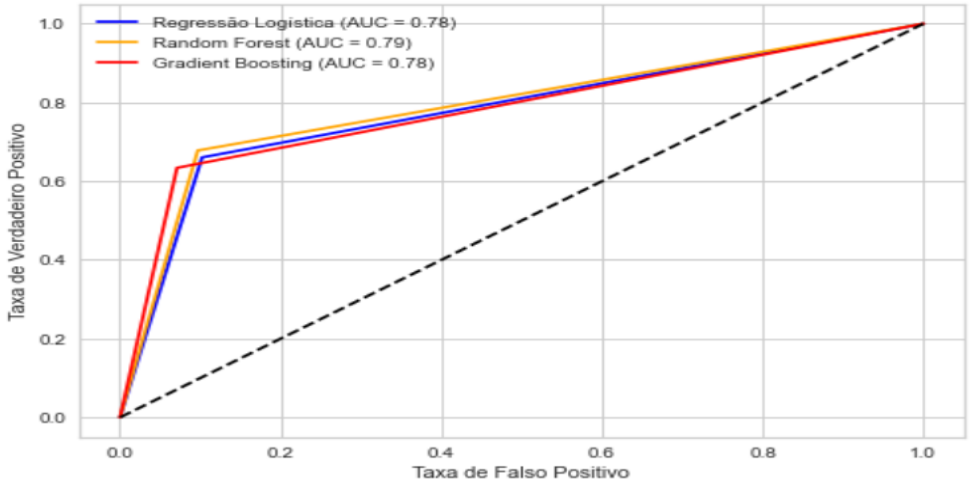
Algoritmo	B. Accuracy	Precision	Recall	F1	Specificity	G-mean	AUC
RL	77,91%	33,63%	66,03%	44,57%	89,78%	77,00%	77,91%
RF	79,06%	35,45%	67,80%	46,55%	90,32%	78,25%	79,06%
GB	78,14%	41,18%	63,38%	49,92%	92,90%	76,74%	78,14%

RL: Regressão Logística; RF: *Random Forest*; GB: *Gradient Boosting*

Fonte: Elaborado pelo autor (2022).

A figura 8 apresenta a curva ROC e a AUC a partir dos resultados obtidos com o balanceamento da base de treino com a técnica SMOTEENN.

Figura 8 – Curva ROC - Resultado com SMOTEENN



Fonte: Elaborado pelo autor (2022).

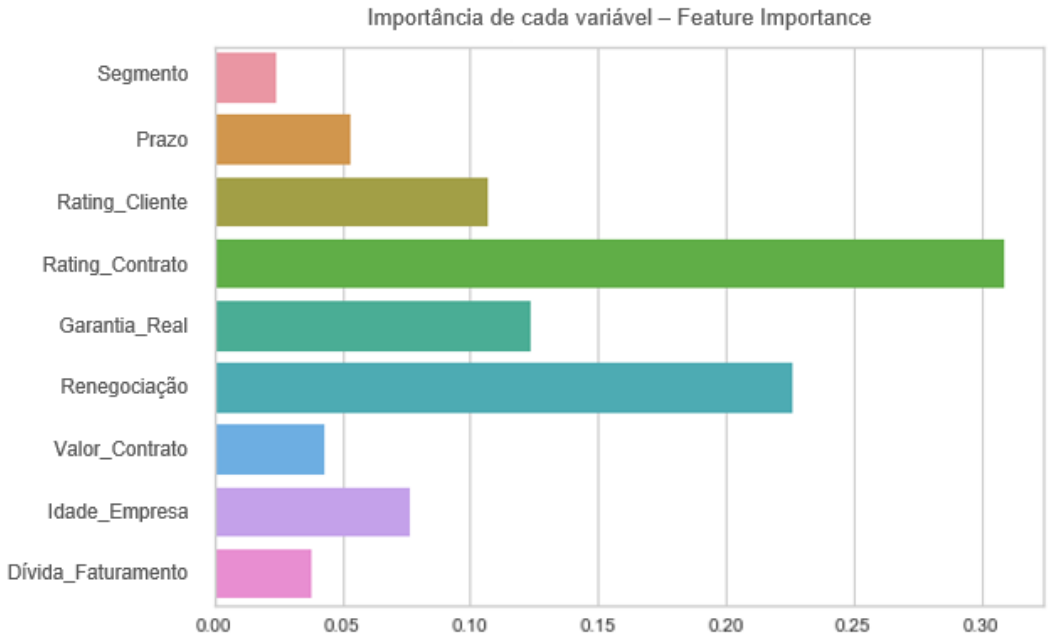
O balanceamento da base de treino com a técnica SMOTEENN apresentou resultados bem similares aos alcançados com a técnica SMOTE nas métricas *Balanced Accuracy*, *G-mean* e AUC, contudo, apresentou melhor desempenho na métrica F1.

O algoritmo *Random Forest* apresentou melhor desempenho nas métricas *Balanced Accuracy*, *G-mean* e AUC, o *Gradient Boosting* apresentou o melhor desempenho na métrica F1. A Regressão Logística apresentou desempenho próximo aos demais algoritmos, mas no geral foi o pior modelo com esse tipo de balanceamento.

Observamos que o balanceamento da base de treino com as técnicas SMOTE e SMOTEEN apresentaram os melhores desempenhos e resultados bem similares, contudo, a combinação que alcançou a maior AUC foi do *Random Forest* com o SMOTE, atingindo resultado de 79,16%.

Para entender quais variáveis foram mais utilizadas pelo modelo que alcançou a maior AUC, utilizamos a ferramenta *feature_importances_*. A figura 9 apresenta o resultado.

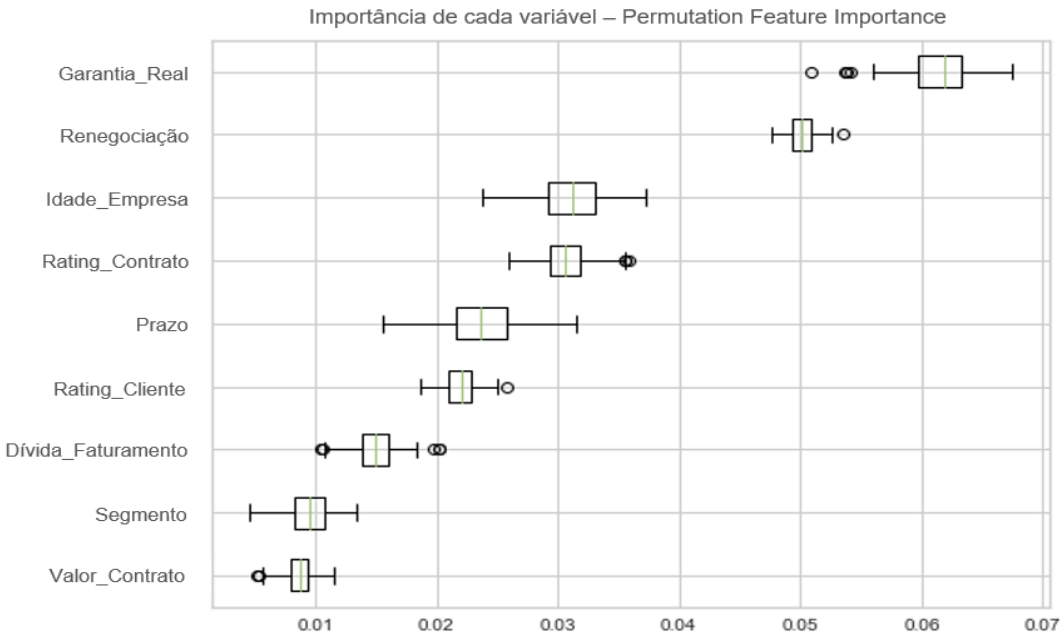
Figura 9 – Feature Importance



Fonte: Elaborado pelo autor (2022).

Também verificamos a relevância das variáveis por meio da ferramenta *permutation feature importance*. Esta ferramenta serve para verificar a redução na pontuação de um modelo quando os valores de uma variável são embaralhados aleatoriamente. Esse procedimento quebra a relação entre a variável independente e a variável dependente, portanto, a queda na pontuação do modelo é indicativo de quanto o modelo depende daquela variável. Configuramos a ferramenta para utilizar a AUC como *scoring* e realizar o embaralhamento aleatório 150 vezes. O resultado consta na figura 10.

Figura 10 – Permutation Feature Importance



Fonte: Elaborado pelo autor (2022).

E para visualizar, de forma quantitativa, os acertos e os erros do modelo, na base de teste, utilizamos a matriz de confusão e o resultado consta na tabela 15.

Tabela 15 – Matriz de confusão

False	9.068	1.034
True	249	543
	False	True

Fonte: Elaborado pelo autor (2022).

5 Conclusões

Avaliamos neste trabalho 3 técnicas de *machine learning*: Regressão Logística, *Random Forest* e *Gradient Boosting*. E para aumentar o desempenho dos algoritmos, diante de uma base de dados desbalanceada, combinamos com 3 técnicas de balanceamento: *NearMiss*, SMOTE e SMOTEENN.

Como medida de sucesso, buscamos o aumento da AUC e, de acordo com os resultados, o algoritmo que apresentou maior AUC foi o *Random Forest* com o balanceamento SMOTE, atingindo resultado de 79,16%, e de maneira oposta, o pior desempenho foi apresentado pela Regressão Logística sem o balanceamento da base de treino, atingido o resultado de 67,99%.

Foram também comparadas outras métricas importantes para avaliação do resultado: *Balanced Accuracy*, *G-mean* e F1. O algoritmo *Random Forest* combinado com o balanceamento SMOTE também apresentou o melhor desempenho nas métricas *G-mean* e *Balanced Accuracy* e o algoritmo *Gradient Boosting* sem balanceamento da base de treino apresentou o melhor F1 score.

Observamos que de modo geral, os 3 algoritmos analisados apresentaram desempenhos parecidos, contudo, o *Random Forest* com SMOTE mostrou leve vantagem em 3 das 4 principais métricas avaliadas, sendo, portanto, considerado o melhor modelo para este trabalho.

Supomos que os modelos implementados não alcançaram melhores desempenhos devido a utilização de uma base de dados referente a um período pandêmico, portanto, uma sugestão de trabalho futuro é a comparação do desempenho obtido neste trabalho com uma avaliação em período de estabilidade na economia do país.

Outra sugestão para trabalhos futuros, diante da complexidade de trabalhar com dados desbalanceados, seria a investigação de outras técnicas de *machine learning* para classificação de tomadores de crédito, em especial as Redes Neurais, que em vários trabalhos apresentou bons resultados, e as variações dos algoritmos *Random Forest* e *Gradient Boosting*.

Por fim, concluímos que a utilização das técnicas de *Machine Learning*, no contexto do risco de crédito, apresenta bons resultados para classificação de micro e pequenas empresas.

Referências

ALBUQUERQUE, P. C.; CAJUEIRO, D. O.; ROSSI, M. D. Machine learning models for forecasting power electricity consumption using a high dimensional dataset. *Expert Systems with Applications*, v. 187, p. 115917, 2022. Citado na página 31.

BAO, W.; LIANJU, N.; YUE, K. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Systems with Applications*, v. 128, p. 301–315, 2019. Citado 2 vezes nas páginas 29 e 31.

BATISTA, G.; PRATI, R.; MONARD, M.-C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, v. 6, p. 20–29, 2004. Citado 2 vezes nas páginas 22 e 33.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, p. 5–32, 2001. Citado na página 31.

BRODERSEN K. H.AND ONG, C. S.; STEPHAN, K. E.; BUHMANN, J. M. The balanced accuracy and its posterior distribution. *20th International Conference on Pattern Recognition*, p. 3121–3124, 2010. Citado na página 34.

BROWN, I.; MUES, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, v. 39, n. 3, p. 3446–3453, 2012. Citado na página 21.

CHAWLA, N. et al. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, v. 16, p. 321–357, 06 2002. Citado 2 vezes nas páginas 22 e 33.

GARCíA-CéSPEDES, R.; MORENO, M. The generalized vasicek credit risk model: A machine learning approach. *Finance Research Letters*, p. 102669, 2022. Citado na página 33.

GOYAL, A.; RATHORE, L.; SHARMA, A. Smo-rf:a machine learning approach by random forest for predicting class imbalancing followed by smote. *Materials Today: Proceedings*, 2021. Citado na página 22.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The elements of statistical learning: Data Mining, Inference, and Prediction*. [S.l.]: Springer, 2009. v. 2ª Edição. Citado na página 32.

KONSTANTINOV, A. V.; UTKIN, L. V. Interpretable machine learning with an ensemble of gradient boosting machines. *Knowledge-Based Systems*, v. 222, p. 106993, 2021. Citado na página 32.

LEE, I.; SHIN, Y. J. Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, v. 63, n. 2, p. 157–170, 2020. Citado na página 29.

MANI, I.; ZHANG, J. knn approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of Workshop on Learning from Imbalanced Datasets*, v. 126, 2003. Citado 2 vezes nas páginas 22 e 33.

SICSú, A. L. *Credit Scoring: Desenvolvimento, Implantação, Acompanhamento*. [S.l.]: Blucher, 2010. v. 1ª Edição. Citado na página 29.

TIAN, Z. et al. Credit risk assessment based on gradient boosting decision tree. *Procedia Computer Science*, v. 174, p. 150–160, 2020. Citado na página 34.

TRAN, N. et al. Hyper-parameter optimization in classification: To-do or not-to-do. *Pattern Recognition*, v. 103, p. 107245, 2020. Citado na página 33.

ZHOU, L. et al. Credit risk modeling on data with two timestamps in peer-to-peer lending by gradient boosting. *Applied Soft Computing*, v. 110, p. 107672, 2021. Citado na página 34.

ZONG, W.; HUANG, G.-B.; CHEN, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing*, v. 101, p. 229–242, 2013. Citado na página 34.