ISSN: 2317-0840

Adaptações do *Extreme Gradient Boosting* para base de dados desbalanceadas com aplicação em *Credit Scoring*

Gabriel Almeida Ferreira[1†], Adriano Kamimura Suzuki[2]

[1] *Aluno do curso de Bacharelado em Estatística e Ciências de Dados no Instituto de Ciências Matemáticas e de computação da Universidade de São Paulo (ICMC-USP)*
[2] *Professor no Departamento de Matemática Aplicada e Estatística no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP)*

Resumo: *O Credit Scoring pode ser visto como um problema de classificação binária, no qual o objetivo é aprender um modelo que classifique clientes como bons ou maus pagadores. Todavia, as bases de dados utilizadas no contexto de Credit Scoring possuem poucos exemplos de maus pagadores, o que pode levar ao erro de classificar um mau pagador como bom pagador e, portanto, gerar prejuízo ao credor. Nesse sentido, este trabalho apresenta o estudo de duas alternativas para lidar com o problema do desbalanceamento das classes: a adaptação dos algoritmos de aprendizado supervisionado, por meio do Extreme Gradient Boosting (XGBoost) utilizando a função de perda Weighted Focal Loss; e a utilização dos algoritmos de balanceamento artificial dos dados, por meio do oversampling e undersampling. Por fim, os resultados foram analisados, ponderações foram feitas sobre a utilização dos métodos propostos, e esses métodos foram aplicados em uma base de dados real. Como resultado, foram obtidos modelos com menor custo esperado, isso é, com menor prejuízo ao credor, porém também foi observada uma piora no Brier Score na abordagem baseada em balanceamento artificial dos dados.*

Palavras-chave: *Credit Scoring; XGBoost; Aprendizado de Máquina; Dados Desbalanceados; Balanceamento dos Dados.*

Adaptations of Extreme Gradient Boosting for Imbalanced Datasets with Application in Credit Scoring

Abstract: *Credit scoring can be seen as a binary classification problem, with the goal of developing a model that classifies customers as good or bad borrowers. However, databases used in credit scoring often have few examples of bad borrowers, which can result in misclassifying bad borrowers as good payers, leading to potential losses for the lender. In this study, two approaches for addressing the issue of class imbalance are explored: firstly, the adaptation of supervised learning algorithms, specifically Extreme Gradient Boosting (XGBoost), utilizing the Weighted Focal Loss function; and secondly, the utilization of artificial data balancing techniques through oversampling and undersampling. Finally, the obtained results are analyzed, considerations regarding the effectiveness of the proposed methods are discussed, and these methods are applied to a real-world database.As a result, models with a lower expected cost were obtained, i.e. with less damage to the creditor, but there was also a worsening in the Brier Score in the approach based on artificial data balancing.*

Keywords: *Credit Scoring; XGBoost; Machine learning; Umbalanced Data; Data Augmentation.*

[†]Autor correspondente: gabrielalmeidaferreira@usp.br

Manuscrito recebido em: 06/06/2024
Manuscrito revisado em: 26/09/2024
Manuscrito aceito em: 30/09/2024

O *Credit Scoring*

"O *Credit Scoring* é um conjunto de modelos de decisão e suas técnicas subjacentes que auxiliam os credores na concessão de crédito ao consumidor"Thomas *et al.* (2002). Nesse sentido, o *Credit Scoring* pode ser visto como um problema de classificação binária, no qual o objetivo é aprender um modelo que classifique clientes como bons ou maus pagadores.

As revisões sistemáticas de Louzada *et al.* (2016) e Dastile *et al.* (2020) indicam que os algoritmos de aprendizado de máquina, devido à sua alta capacidade preditiva, vêm ganhando relevância no contexto de *Credit Scoring*. Em particular, recentemente o XGBoost *(Extreme Gradient Boosting)* Chen e Guestrin (2016) vem sendo utilizado no contexto de *Credit Scoring*, como nos trabalhos de Chang *et al.* (2018) e Li *et al.* (2020).

No entanto, as base de dados utilizadas no contexto de *Credit Socring* tem mais exemplos de bom pagadores do que maus pagadores. Por exemplo, a seguinte tabela (Tabéla 1) reúne as principais base de dados utilizadas na literatura de *Credit Scoring*.

Table 1: Databases used in the Credit Scoring literature.

Conjunto de dados	Observações	Atributos	Proporção de maus pagadores
European Credit Bureau	186.574	324	0,038
UK	30.000	14	0,040
Barbados	21.117	20	0,024
Indonesia	14.700	31	0,300
Benelux 2	7.190	28	0,300
Brazilian	4.504	5	0,080
Benelux 1	3.123	27	0,666
UC San Diego	2.435	38	0,246
China	1.057	10	0,477
German	1.000	20	0,300
Iranian	1.000	27	0,050
Australian	690	14	0,555
Japanese	653	15	0,546
Polish	240	30	0,466
Texas Banks	162	19	0,500

Source: Adapted from Dastile *et al.* (2020).

Como pode ser observado (Tabela 1), em todas as bases de dados, com diferentes intensidades, há, em geral, mais exemplos de bons pagadores. Quando uma base de dados apresenta discrepância entre o número de exemplos das classes, ela é chamada de base de dados desbalanceada. Diante disso, é evidente que o problema de *Credit Scoring* pode ser caracterizado como um problema de classificação binária desbalanceada.

Algoritmos baseados em *Gradient Boosting* Friedman (2001), como é o caso do XGBoost, vêm sendo utilizados no contexto de *Credit Scoring*. Por exemplo, o trabalho de Brown e Mues (2012) aponta que essa classe de algoritmos apresenta bons resultados, mesmo considerando o desbalanceamento dos dados.

Contudo, segundo Fernández *et al.* (2018), adaptações podem ser necessárias para que os algoritmos de aprendizado supervisionado apresentem bom desempenho em bases de dados desbalanceadas. Em particular, neste trabalho, foram consideradas duas abordagens: o balanceamento artificial dos dados e a modificação de algoritmos de aprendizado supervisionado. Essas adaptações são condizentes com o que já é empregado na literatura, por exemplo, no livro supracitado, a primeira adaptação é chamada de *Data Level* e a segunda de *Cost-sensitive learning*.

Em especial, os trabalhos de Batista *et al.* (2004) e More *et al.* (2016) indicam que a utilização de procedimentos de balanceamento artificial dos dados pode aumentar a capacidade preditiva dos algoritmos de aprendizado supervisionado. Por outro lado, apesar de haver evidências da eficácia dos algoritmos de balanceamento artificial dos dados, a revisão sistemática sobre as técnicas de aprendizado supervisionado no contexto de *Credit Scoring* de Dastile *et al.* (2020) aponta que, dentre todos os trabalhos considerados, apenas 18% utilizaram alguma técnica de balanceamento artificial dos dados. Essa baixa adoção, somada às evidências da eficácia dos algoritmos de balanceamento artificial dos dados, corrobora a adoção desse procedimento nos estudos de algoritmos de aprendizado de máquina aplicados ao *Credit Scoring*. Dessa forma, além de uma possível melhoria da capacidade preditiva dos algoritmos; terá se uma maneira de verificar as evidências encontradas nos estudos supracitados por meio da comparação com a abordagem mais usual - isto é, não utilizar o balanceamento artificial dos dados.

Li *et al.* (2017) introduziu e estudou a função de perda *Weighted Focal Loss* para lidar com dados desbalanceados no contexto de detecção de objetos em visão computacional, utilizando essa função de perda em uma rede neural. Posteriormente, o estudo de Wang *et al.* (2020) considerou a utilização da *Weighted Focal Loss* como função de perda no XGBoost. Nesse estudo, o autor verificou que, em comparação com o XGBoost com a função de perda padrão, o XGBoost com a função de perda *Weighted Focal Loss* obteve melhores resultados em bases de dados desbalanceadas. Recentemente, o trabalho de Mushava e Murray (2022) abordou a utilização do XGBoost com funções de perda adaptadas para dados desbalanceados em bases de dados de *Credit Scoring*, obtendo bons resultados. Isso posto, conclui-se que a utilização de funções de perda adequadas para o desbalanceamento de dados é recomendada, pois essa abordagem pode resultar na obtenção de um modelo com maior capacidade preditiva e, por consequência, evitando prejuízo ao credor.

Métricas para conjuntos de dados desbalanceados

Devido ao desbalanceamento dos dados, é necessário adaptar o processo de avaliação da capacidade preditiva de um classificador. Por exemplo, quando uma base de dados está desbalanceada, a acurácia, que é a métrica mais utilizada em problemas de classificação, não é recomendada, pois essa métrica favorece classificadores que menosprezam a classe minoritária. Uma revisão abrangente sobre as métricas adequadas para avaliar classificadores ajustados a dados desbalanceados pode ser encontrada no Capítulo 3 do livro de Fernández *et al.* (2018). Nesta subseção, descreveremos as métricas que serão utilizadas para avaliar o desempenho preditivo dos modelos.

Em um modelo de *Credit Scoring* há duas classes: bom pagador e mau pagador. Considerando que a classe bom pagador foi codificada como classe positiva (1) e mau pagador como classe negativa (0), temos a seguinte matriz de confusão (Quadro 1):

Chart 1: Confusion matrix.

VERDADE/PREDITO	Bom pagador	Mau pagador
Bom pagador	VP (Verdadeiro Positivo)	FN (Falso Negativo)
Mau pagador	FP (Falso Positivo)	VN (Verdadeiro Negativo)

Source: from the authors (2024).

Por meio desta matriz de confusão e considerando que P é a precisão, R a revocação e MCC o *Matthews's correlation coefficient*, é possível obter as seguintes métricas:

$$P = \frac{VP}{VP + FP}$$
$$R = \frac{VP}{VP + FN}$$
$$F1 = 2 \times \frac{P \times R}{P + R}$$
$$MCC = \frac{VP \cdot VN - FP \cdot FN}{\sqrt{(VP + FP) \cdot (VP + FN) \cdot (VN + FP) \cdot (VN + FN)}}$$

É importante notar que, dependendo da métrica escolhida, diferentes informações da matriz de confusão são utilizadas. Das métricas descritas, apenas o MCC considera todos os elementos da matriz de confusão. Observa-se também que todas as métricas levam em conta informações sobre pelo menos um dos tipos de erros (FP e FN), ao contrário da acurácia, que considera informações apenas sobre VP e VN. Por essa razão, é recomendado utilizar as métricas listadas Fernández et al. (2018, p. 49).

É comum utilizar uma matriz de custo para ponderar a penalização dos Falsos Positivos ou Falsos Negativos. Uma matriz de custo é definida da seguinte forma (Quadro 2):

Chart 2: Cost matrix.

VERDADE/PREDITO	Bom pagador	Mau pagador
Bom pagador	C_{11}	C_{21}
Mau pagador	C_{21}	C_{22}

Source: from the authors (2024).

Então, utilizando essa matriz, de custo podemos definir o risco esperado como:

$$RE = \frac{VP \times C_{11} + FP \times C_{12} + FN \times C_{21} + VN \times C_{22}}{n}$$

Os autores que disponibilizaram a base de dados *Statlog (German Credit Data)* Hofmann (1994), que será analisada nesse trabalho, sugeriram a utilização da seguinte com as seguintes configurações: $C_{11}, C_{22} = 0, C_{21} = 5, C_{12} = 1$. Logo, nessas configurações, o risco esperado é dado por:

$$RE = \frac{-(5 \times FP + FN)}{n}$$

Convém ressaltar que os modelos empregados neste estudo geram uma pontuação, sendo a classificação nominal realizada após o ajuste do modelo, por meio da aplicação de um ponto de corte. É imediato que a escolha desse ponto de corte impacta diretamente as métricas que utilizam a matriz de confusão. Por isso, a definição desse ponto de corte e das métricas a serem otimizadas é uma etapa crucial na avaliação preditiva dos modelos de *Credit Scoring*. Nesse sentido, ao comparar as previsões nominais de diferentes modelos dentro do contexto de *Credit Scoring*, é aconselhável empregar diversas métricas e considerar que o erro de classificar um mau pagador como bom pagador é mais prejudicial ao credor.

Como mencionado anteriormente, a avaliação dos modelos por meio de métricas que consideram predições nominais depende do ponto de corte escolhido para designar as classes. Por isso, também serão utilizadas a área sob a curva ROC e o Brier Score, que são métricas calculadas independentemente do ponto de corte.

A curva ROC é um método de avaliação gráfica que consiste em ordenar as predições em ordem decrescente e, para cada uma das predições ordenadas, gerar um ponto no gráfico.

No eixo x, temos a taxa de falsos positivos, e no eixo y, a taxa de verdadeiros positivos. Uma linha de 45 graus é traçada no gráfico; um classificador abaixo dessa linha é pior do que um chute aleatório. Quanto mais próximo do canto superior esquerdo, melhor é o classificador. Uma métrica utilizada para resumir o desempenho de um classificador é a área sob a curva ROC, que varia de 0 a 1. Quanto mais próximo de 1, melhor Bradley (1997).

O Brier Score é calculado da seguinte forma:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2,$$

em que p_i é a pontuação obtida pelo modelo para i-ésima pontuação e o_i é classe da i-ésima observação, essa métrica varia de 0 a 1 e quanto mais próximo de 0 melhor.

Validação cruzada para dados desbalanceados e dados balanceados artificialmente

A validação cruzada é um método frequentemente empregado para avaliar o desempenho de um classificador. O esquema de validação cruzada mais comum é o *K-fold cross-validation*, no qual os dados são divididos em k partes independentes e disjuntas. Durante o processo, k-1 partes são utilizadas para treinamento, enquanto a parte restante é reservada para avaliar o desempenho em um conjunto de teste. As partes designadas para teste e treinamento são alternadas até que todas as partes tenham sido utilizadas para treino e teste. Ao final do processo, obtemos k métricas resultantes das k partes previamente divididas, e essas métricas são resumidas por meio da média e do desvio padrão Hastie et al. (2001).

Todavia, conforme destacado por Santos et al. (2018), é necessário tomar precauções ao aplicar a validação cruzada a conjuntos de dados desbalanceados, especialmente quando algoritmos de balanceamento artificial de dados são utilizados. Este autor observou que aplicar os métodos de balanceamento artificial antes de dividir os dados em conjuntos de treino e teste resulta em uma avaliação otimista dos modelos. Além disso, ele sugere que os conjuntos de treino e teste sejam divididos para manter a proporção de 0 e 1 nos conjuntos de treino e teste, reduzindo assim o impacto do desbalanceamento dos dados durante o procedimento da validação cruzada. As etapas desse procedimento são descritas a seguir:

1. Divida a base de dados em k partes, mantendo a proporção de 0 e 1.
2. Para cada uma das partes, reserve uma delas para teste e o restante para treinamento.
3. Utilize os algoritmos de balanceamento artificial apenas no conjunto de treinamento.
4. Avalie o desempenho no conjunto de teste.

Ressalta-se que o procedimento de validação cruzada mencionado anteriormente é aplicado em todo o trabalho, e foram tomadas precauções para evitar o vazamento de dados. O vazamento de dados ocorre quando há o uso acidental de informações do conjunto de treinamento para modificar o conjunto de teste, como, por exemplo, na padronização do conjunto de teste utilizando a média e o desvio padrão do conjunto de treinamento.

Os softwares que implementam os algoritmos de balanceamento artificial recomendam os procedimentos mencionados acima em seus manuais. Por exemplo, o manual do software a ser utilizado neste trabalho possui uma seção dedicada a erros comuns na avaliação preditiva quando se utiliza balanceamento artificial dos dados. Este manual pode ser encontrado em Lemaître et al. (2017).

Objetivos

O objetivo desse trabalho é estudar a adaptação do XGBoost para conjuntos de dados desbalanceados, considerando uma aplicação em uma base de dados no contexto de *Credit Scoring*, em que o desbalanceamento dos dados é comum.

Considerar adaptações do XGBoost para dados desbalanceados é crucial para os problemas de *Credit Scoring*, pois um dos principais desafios nesse cenário é o desbalanceamento dos dados, conforme apontam os trabalhos de Dastile et al. (2020) e Louzada et al. (2016). Isso é ainda mais relevante quando se considera que o custo associado ao erro de classificar um mau pagador (classe minoritária) como bom pagador (classe majoritária) é mais grave do que classificar um bom pagador como mau pagador, veja, por exemplo, a Tabela 2. Portanto, um algoritmo que desempenha mal em conjuntos de dados desbalanceados, por exemplo, classificando muitos maus pagadores como bons pagadores, pode gerar muito prejuízo ao credor, evidenciando a necessidade de adaptá-los aos conjuntos de dados desbalanceados.

Por fim, também é objetivo desse trabalho, considerando um esquema de validação cruzada e métricas adequadas para o desbalanceamento dos dados, em uma base de dados real de *Credit Scoring*, comparar as adaptações do XGBoost para bases de dados desbalanceadas com as configurações padrões. Assim, será possível discutir sobre o efeito das adaptações do XGBoost estudadas no trabalho.

Materiais e métodos

XGBoost com função de perda *Weighted Focal Loss*

O XGBoost Chen e Guestrin (2016) é um algoritmo de aprendizado de máquina baseado no conceito de *Gradient Tree Boosting* Friedman (2001) que vem ganhando popularidade devido a seu ótimo desempenho em competições de aprendizado de máquina, em especial o XGBoost tem ganhado relevância no contexto de *Credit Scoring*. Por exemplo, Chang et al. (2018) comparou a desempenho de diferentes algoritmos de aprendizado de máquina para uma tarefa de classificação binária e concluiu que, ao menos empiricamente, o XGBoost teve melhor desempenho dentre os algoritmos selecionados.

Para um dado conjunto D de treinamento com m atributos e n exemplos representado da seguinte forma:

$$D = \{(\mathbf{x_i}, y_i)\} \ \ (|D| = n, \mathbf{x_i} \in R^m, y_i \in R)$$

O método XGBoost consiste em realizar uma previsão baseada na agregação dos resultados de várias árvores de decisões. Matematicamente, isso é representado por:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F},$$

em que k é a quantidade de árvores e $\mathcal{F}$ é o espaço das árvores de classificação e regressão (CART) Breiman (2017).O ajuste do modelo é feito minimizando a seguinte função:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$
$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

sendo que Ω penaliza a complexidade do modelo de modo a evitar o overfit e $l(\hat{y}_i, y_i)$ é uma função de perda.

Para minimizar $\mathcal{L}(\phi)$ utiliza-se o seguinte método iterativo:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Ou seja, as árvores são treinadas de modo que a cada iteração a função de perda é reduzida, sendo o classificador final formado pela combinação das predições de várias árvores de decisão. Na nomenclatura dos algoritmos de *Boosting*, utilizada em Schapire et al. (1999), dizemos que um classificador forte, nesse caso o XGBoost, é formado pela contribuição de vários classificadores fracos (CART).

Recentemente Wang et al. (2020) estudaram uma adaptação do XGBoost para base de dados desbalanceadas. Essa adaptação consiste em utilizar um função de perda que introduz hiperparâmetros que podem ser utilizados para dar enfasê ao aprendizado de uma das classes. A função de perda utilizada é a *Weighted Focal Loss*, que é descrita pela seguinte equação:

$$L = -\sum_{i=1}^{m} \left[\alpha y_i (1-\hat{y}_i)^{\gamma} \log(\hat{y}_i) + (1-y_i)\hat{y}_i^{\gamma} \log(1-\hat{y}_i)\right], \tag{1}$$

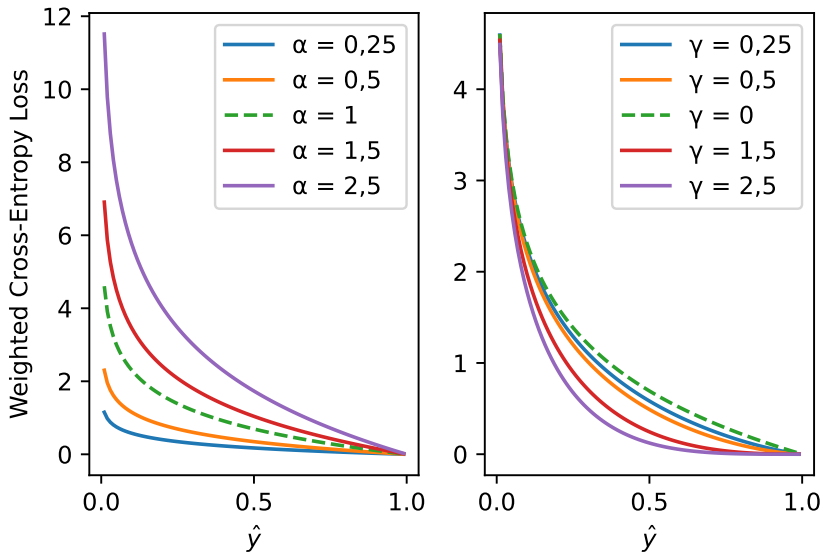
sendo que $y_i \in \{0, 1\}$ é a classe do i-ésimo exemplo, $\hat{y}$ é a probabilidade predita, e α e γ são os hiperparâmetros introduzidos pela função de perda. O hiperparâmetro α pode ser interpretado como um hiperparâmetro responsável por introduzir peso aos falsos positivos e aos falos negativos, o que pode ser útil no problema de *Credit Scoring*, já que o erro de classificar um mau pagador como bom pagador (falso positivo) geralmente tem mais peso do que classificar um bom pagador como mau pagador (falso negativo). Já o hiperparâmetro γ, segundo Lin et al. (2017), pode ser interpretado como um hiperparâmetro responsável por "direcionar o aprendizado aos exemplos difíceis de classificar". Quando $\alpha = 1$ e $\gamma = 0$, a função de perda se reduz à função de perda *Binary Cross-Entropy*, que é a função de perda mais comumente utilizada nos modelos XGBoost.

Uma visualização do efeito dos hiperparâmetros na função de perda pode ser obtida por meio de um gráfico em que um exemplo da classe positiva é considerada, ou seja, na equação 1 foi considerado que $y_i = 1$. Nesse gráfico, para diferentes configurações de hiperparâmetros, exibimos a função de perda (L) no eixo y e a probabilidade predita ($\hat{y}$) no eixo x. No gráfico à esquerda, γ foi fixado em 0 para avaliar separadamente o efeito do hiperparâmetro α, e no gráfico à direita, α foi fixado em 1 para avaliar separadamente o efeito do hiperparâmetro γ. Nos dois gráficos, a linha tracejada corresponde à função de perda *Binary Cross-Entropy*, que é a mais utilizada no XGBoost.

Note que, no gráfico à esquerda da Figura 1, é possível perceber que o hiperparâmetro α introduz uma flexibilidade em relação à *Binary Cross-Entropy*, no sentido de que é possível obter funções de perda acima ou abaixo dela, dependendo da escolha de α. É importante notar que, se $\hat{y} \leq 0.5$, o algoritmo cometeria o erro de classificar 1 como 0 (falso negativo), e pelo gráfico é possível observar que, conforme α aumenta, maior é a perda atribuída a esse erro.

Por outro lado, no gráfico à direita da Figura 1, é possível perceber que o hiperparâmetro γ introduz uma flexibilidade em relação à *Binary Cross-Entropy*, no sentido de que é possível obter curvas abaixo da função de *Binary Cross-Entropy*, sendo que, conforme γ aumenta, essas curvas ficam mais abaixo em relação à função de *Binary Cross-Entropy*. Lin et al. (2017) argumenta que o hiperparâmetro γ direciona o aprendizado para os exemplos que são mais difíceis de aprender. Para ele, um exemplo fácil de aprender, no caso em que $y_i = 1$, é um exemplo cuja probabilidade predita é próxima de um, ou seja, a parte em que $\hat{y} > 0.5$. Nesse sentido, conforme o exposto pelo gráfico à direita da Figura 1 ,percebe-se que, conforme γ aumenta, menor perda é atribuída a esses exemplos fáceis de aprender, justificando assim que o hiperparâmetro γ pode ser interpretado como um hiperparâmetro que direciona o aprendizado para os exemplos que são mais difíceis de aprender.

Figure 1: Graph of loss functions, considering different hyperparameters.



Source: from the authors (2024).

Métodos de balanceamento artificial dos dados

Os algoritmos de balanceamento artificial de dados abordam o problema do desbalanceamento de classes por meio da modificação do conjunto de treinamento. Essa alteração pode ser realizada de duas maneiras: aumentando o número de exemplos da classe minoritária (*OverSampling*) ou reduzindo o número de exemplos da classe majoritária (*UnderSampling*). Ao final desse processo, obtém-se um conjunto de dados artificialmente balanceado, e um algoritmo de classificação é treinado nesse novo conjunto. Geralmente, observa-se um aumento na capacidade preditiva quando esse procedimento é adotado, como visto nos trabalhos de Batista et al. (2004) e More (2016).

OverSampling

O algoritmo de *Oversampling* estudado neste trabalho é o *Adaptive synthetic sampling approach for imbalanced learning* ADASYN He et al. (2008). Esse algoritmo gera exemplos da classe minoritária de forma adaptativa, isso é, mais dados sintéticos são gerados para exemplos da classe minoritária que são 'difíceis' de aprender.

A quantidade de dados sintéticos gerados para um exemplo da classe minoritária é obtida da seguinte forma:

1. Defina m_s e m_l como o número de exemplos da classe minoritária e majoritária, respectivamente. Calcule $G = \beta(m_l - m_s)$, onde $\beta \in (0, 1)$ controla o nível de balanceamento. Se $\beta = 1$, ao final do processo, o conjunto de dados terá metade dos exemplos da classe positiva e metade da classe negativa.

2. Para cada exemplo da classe minoritária, calcule $r_i = \Delta_i / K$, em que Δ_i é o número de vizinhos mais próximos que pertencem à classe majoritária, e K é a quantidade de vizinhos mais próximos considerados.

3. Normalize $\hat{r}_i = r_i / \sum_i^{m_s} r_i$ e, em seguida, calcule o número de exemplos a serem gerados para o i-ésimo exemplo da classe minoritária: $g_i = \hat{r}_i G$.

He et al. (2008) argumenta que $\hat{r_i}$ pode ser utilizado como uma medida para determinar a dificuldade de aprendizado de um exemplo. Note que a quantidade de exemplos sintéticos gerados é proporcional a $\hat{r_i}$, o que significa que, além de balancear artificialmente os dados, o ADASYN gera mais exemplos em regiões que são mais difíceis de aprender.

Para gerar um dado sintético o seguinte procedimento é adotado:

1. Seja $\mathbf{x_i}$ o exemplo de interesse. Dentre os K vizinhos mais próximos, escolha aleatoriamente um elemento da classe minoritária, denotado por $\mathbf{x_{zi}}$.

2. Gere um exemplo sintético da seguinte forma: $\mathbf{s_i} = \mathbf{x_i} + \lambda(\mathbf{x_{zi}} - \mathbf{x_i})$. em que λ é um número aleatório no intervalo [0,1].

Esse procedimento permite que exemplos semelhantes, mas não duplicados, sejam gerados a partir de exemplos da classe minoritária. Para cada um dos $i = 1 \dots m_s$ exemplos da classe minoritária, esse procedimento é repetido g_i vezes.

Uma vez aplicado o ADASYN, obtém-se um conjunto de dados balanceado artificialmente e com mais exemplos em regiões de difícil aprendizado, o que, em geral, levará ao aumento do poder preditivo de algoritmos aplicados posteriormente ao ADASYN.

UnderSampling

O algoritmo de *UnderSampling* estudado neste trabalho é o *Edited Nearest Neighbors*EEN. Para remover elementos da classe majoritária, o seguinte procedimento é adotado:

1. Para cada exemplo da classe majoritária, encontre os k vizinhos mais próximos.

2. Encontre a classe mais frequente entre os k vizinhos mais próximos.

3. Descarte todos os exemplos em que a classe mais frequente é a classe minoritária.

Ao final desse procedimento, o problema do desbalanceamento dos dados será atenuado. A estratégia adotada pelo EEN é útil, pois tende a remover ruído, eliminando instâncias da classe majoritária que não concordam com a maioria dos seus k vizinhos mais próximos. Intuitivamente, essa heurística "limpa"a classe majoritária, removendo exemplos que dificultariam o aprendizado dos exemplos da classe minoritária. Mais detalhes sobre essa heurística podem ser encontrados em Wilson (1972).

Base de dados

A base de dados analisada neste trabalho foi a *Statlog (German Credit Data)* Hofmann (1994), que é apontada pelas revisões sistemáticas de Dastile et al. (2020) e Louzada et al. (2016) como uma das bases de dados mais utilizadas para avaliar o desempenho de algoritmos de aprendizado supervisionado no contexto de *Credit Scoring*. Trata-se de uma base de dados com 20 atributos, sendo 13 categóricos e 7 numéricos, e 1000 exemplos, em que o objetivo é predizer se o cliente do banco será um bom (1) ou mau pagador (0). Nessa base de dados, há 300 maus pagadores e 700 bons pagadores, indicando um desbalanceamento dos dados.

O esquema de pré-processamento envolveu a padronização dos atributos numéricos para que eles tenham média 0 e desvio padrão 1, o que é importante, já que os algoritmos de balanceamento artificial dos dados irão utilizar o conceito de vizinhos mais próximos em seus algoritmos. Além disso, foi utilizado o método *One-Hot Encoding* para os atributos categóricos.

O autor da base de dados sugere que seja adotada a seguinte matriz de custos (Tabela 2):

Table 2: Cost matrix.

VERDADE/PREDITO	Bom pagador	Mau pagador
Bom pagador	0	1
Mau pagador	5	0

Source: from the authors (2024).

Ou seja, classificar mau pagador como bom pagador tem 5 vezes mais custo que classificar bom pagador como mau pagador, refletindo assim os diferentes graus de prejuízo ao credor. Além disso, utilizando a Tabela 2, o custo esperado para a classe positiva é:

$$0 \times P(Bom\ pagador) + 5 \times P(Mau\ pagador) = 5 \times P(Mau\ pagador) = 5 \times (1 - P(Bom\ pagador))$$

E o custo esperado para a classe negativa é

$$1 \times P(Bom\ pagador) + 0 \times P(Mau\ Pagador) = P(Bom\ pagador)$$

Um exemplo é classificado como classe positiva se o risco esperado da classe positiva é menor do que o risco esperado da classe negativa. Isso é: $5 - P(Bom\ pagador) \leq P(Bom\ pagador)$, resolvendo para $P(Bom\ pagador)$ obtemos o ponto de corte ótimo, segundo a matriz de custo apresentada, dado por: $P(Bom\ Pagador) \geq \frac{5}{6}$. Dessa forma, o ponto de corte que sera utilizado é $\frac{5}{6}$, ou seja um exemplo é predito como da classe positiva (bom pagador) se a sua probabilidade predita for maior ou igual $\frac{5}{6}$.

Ambiente computacional

Todo o trabalho foi realizado no ambiente Google Colaboratory. Esse recurso possibilita a escrita e execução de códigos em Python diretamente no navegador. Para obter informações mais detalhadas sobre essa ferramenta, recomenda-se consultar ⟨https://colab.research.google.com/⟩. Para o ajuste dos algoritmos de balanceamento dos dados, foi utilizada a biblioteca *imbalanced-learn* Lemaître et al. (2017), e para o processamento dos dados e ajuste do XGBoost, foi considerada a biblioteca *scikit-learn* Pedregosa et al. (2011). Além disso, para garantir a reprodutibilidade dos resultados apresentados, uma semente foi fixada.

Resultados

Considerando um esquema de validação cruzada com 10 *folds* estratificados e tomando os cuidados necessários para evitar o vazamento dos dados, foram considerados os seguintes modelos:

- XGBoost com as configurações padrões do *scikit-learn* (XGB).
- XGBoost após o balanceamento por meio do ADASYN considerando $\beta = 1$, ou seja, considerando que o conjunto de treino, após o balanceamento dos dados, tenha a mesma quantidade de zeros e uns (XGB ADASYN).
- XGBoost após o balanceamento por meio do EEN (XGB EEN).
- XGBoost com a função de perda *Weighted Focal Loss* (XGB FOCAL).

Os hiperparâmetros γ e α do XGBoost com a função de perda *Weighted Focal Loss* foram escolhidos por meio de validação cruzada de acordo com o recomendado por Lin et al. (2017).

A Tabela 3 exibe o resultado dos modelos ajustados para as métricas independentes do ponto de corte. Fora dos parênteses, está a média nos 10 *folds* da validação cruzada e entre parênteses o desvio padrão.

Table 3: Cut-off point independent metrics.

Modelo	Área sob a curva ROC	Brier Score
XGB ADASYN	0,773 (0,052)	0,190 (0,026)
XGB EEN	0,785 (0,060)	0.239 (0,048)
XGB FOCAL	0,784 (0.051)	0,165 (0,020)
XGB	0,783 (0,047)	0,179 (0,026)

Source: from the authors (2024).

Note que dentre os modelos estudados, aqueles que têm a área sob a curva ROC maior do que o XGB são o XGB EEN e o XGB FOCAL, embora essa diferença não seja tão pronunciada. Contudo, o modelo XGB ADASYN teve área sob a curva ROC menor do que o XGB. Em relação ao Brier Score, o único modelo que teve desempenho melhor do que o XGB foi o XGB FOCAL.

Note que o XGB teve as melhores métricas se considerarmos o MCC, Revocação e F1 Score. No entanto, considerando o custo esperado e a precisão, todos os modelos estudados neste trabalho apresentam desempenho melhor do que o XGB. Em particular, o XGB EEN e o XGB FOCAL apresentaram menor custo esperado em comparação com o XGBoost, sendo que essa diferença foi maior que 0,1, indicando que a utilização desses modelos pode contribuir para reduzir o prejuízo gerado ao credor (Tabela 4).

Table 4: Cut-off point dependent metrics.

Modelo	MCC	Precisão	Revocação	Custo Esperado	F1
XGB ADASYN	0,378 (0,079)	0,852 (0,036)	0,690 (0,044)	0,642 (0,116)	0,761 (0,031)
XGB EEN	0,349 (0,101)	0,892 (0,055)	0,520 (0,061)	0,556 (0,129)	0,655 (0,057)
XGB FOCALL	0,343 (0,110)	0,886 (0,053)	0,521 (0,067)	0,565 (0,136)	0,655 (0,066)
XGB	0,384 (0,081)	0,849 (0,040)	0,709 (0,035)	0,649 (0,126)	0,772 (0,027)

Source: from the authors (2024).

Discussão

Semelhante ao observado em Batista et al. (2004), foi observada uma melhora, embora não pronunciada, na área sob a curva ROC quando um algoritmo de aprendizado com balanceamento artificial dos dados foi utilizado, no caso deste trabalho, o EEN. Em relação à precisão e à revocação, verificou-se que, em comparação com o XGB sem balanceamento artificial de dados (XGB), houve um aumento na precisão, mas uma diminuição na revocação. O aumento na precisão e a diminuição da revocação são mais pronunciados no EEN. Essa mudança na precisão e na revocação também foi observada em More (2016).

O trabalho Mushava e Murray (2022) mostrou que a utilização de funções de perda adequadas para conjuntos de dados desbalanceados pode melhorar a capacidade preditiva do XGBoost em conjuntos de dados de *Credit Scoring*. Os resultados desse trabalho corroboram com o que foi observado no estudo desse autor, principalmente quando é considerado o *Brier Score* e o custo esperado. Contudo, uma melhora expressiva não foi observada na área sob a curva ROC e houve piora no MCC e no F1, isso indica que em estudos futuros, semelhante ao considerado Mushava e Murray (2022), podem ser consideradas diferentes funções de perda, de modo a tentar melhorar as métricas citadas.

Por fim, quando comparamos o balanceamento artificial dos dados versus o XGBoost com a função de perda *Weighted Focal Loss*, é possível notar uma piora no *Brier Score* para as abordagens que envolvem o balanceamento artificial dos dados, o que é um indicio da piora da probabilidade estimada Fernández et al. (2018, p. 57). Na literatura, o estudo Goorbergh et al. (2022) observou a piora da probabilidade estimada por modelos de regressão logística quando utilizados em conjunto com o balanceamento artificial dos dados. Uma consequência disso é uma dificuldade para determinar um ponto de corte para gerar predições nominais, principalmente quando uma matriz de custos é fornecida, como é o caso da base de dados utilizada nesse trabalho.

Em suma, diante do que foi exposto acima, recomenda-se que as adaptações do XGBoost estudadas nesse trabalho sejam consideradas, principalmente pela melhora no custo esperado. No entanto, é necessário cautela ao utilizar algoritmos de balanceamento artificial dos dados, devido à piora do *Brier Score*, o que pode indicar uma piora na probabilidade estimada e, por conseguinte, a melhora no custo esperado pode não ser tão expressiva, como é o caso do ADAYSN.

Considerações Finais

Foi argumentado que o problema de *Credit Scoring* pode ser abordado como um problema de classificação desbalanceada. Nesse sentido, foram estudados métodos para lidar com o desbalanceamento dos dados. Esses métodos foram avaliados em uma base frequentemente utilizada na literatura de *Credit Scoring*, tomando as precauções necessárias para evitar o vazamento dos dados ao aplicar o balanceamento artificial dos dados. O principal resultado é a redução do custo esperado. Essa redução no custo é mais expressiva para a combinação do balanceamento artificial por meio do EEN e do XGBoost com função de perda *Weighted Focal Loss*. Isso indica que os métodos estudados neste trabalho podem ser utilizados como alternativa ao XGBoost, no sentido de que o prejuízo gerado ao credor pode ser reduzido.

Todavia, os métodos de balanceamento artificial dos dados apresentaram uma piora no Brier Score, o que pode indicar que a utilização desses métodos pode resultar em uma estimativa de probabilidade menos precisa. Isso pode ser prejudicial no contexto de *Credit Scoring*, principalmente porque dificulta a escolha de um ponto de corte para classificação.

Por fim, um próximo passo pode ser estudar métodos de aprendizado de máquina interpretáveis, para que as previsões do XGBoost sejam explicadas de forma a garantir a transparência no modelo de *Credit Scoring*. Além disso, outra alternativa é propor funções de perda diferentes da *Weighted Focal Loss*, semelhante ao que foi feito no trabalho de Mushava e Murray (2022).

Agradecimentos

Agradecimento à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro, por meio do processo nº 2023/06883-3.

Referências

BATISTA, G. E.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, v. 6, p. 20–29, 2004.

BRADLEY, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, v. 30, n. 7, p. 1145–1159, 1997.

BREIMAN, L. *Classification and regression trees*. [S.l.]: Routledge, 2017.

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.

CHANG, Y.-C.; CHANG, K.-H.; WU, G.-J. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing*, v. 73, p. 914–920, 2018.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 2016. (KDD '16), p. 785–794.

DASTILE, X.; CELIK, T.; POTSANE, M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, v. 91, p. 106263, 2020.

FERNÁNDEZ, A.; GARCÍA, S.; GALAR, M.; PRATI, R. C.; KRAWCZYK, B.; HERRERA, F. *Learning from imbalanced data sets*. [S.l.]: Springer, 2018. v. 10.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, p. 1189–1232, 2001.

GOORBERGH, R. van den; SMEDEN, M. van; TIMMERMAN, D.; CALSTER, B. V. The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *Journal of the American Medical Informatics Association*, v. 29, n. 9, p. 1525–1534, 2022.

HASTIE, T.; FRIEDMAN, J.; TIBSHIRANI, R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. [S.l.]: Springer New York, 2001. 193–224 p.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. [S.l.: s.n.], 2008. p. 1322–1328.

HOFMANN, H. *Statlog (German Credit Data)*. 1994. UCI Machine Learning Repository. Disponível em: ⟨https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data⟩.

LEMAîTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. *Common pitfalls and recommended practices*. 2017. Disponível em: ⟨https://imbalanced-learn.org/stable/common_pitfalls.html⟩.

LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, v. 18, p. 559–563, 2017.

LI, H.; CAO, Y.; LI, S.; ZHAO, J.; SUN, Y. Xgboost model and its application to personal credit evaluation. *IEEE Intelligent Systems*, v. 35, p. 52–61, 2020.

LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLÁR, P. Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 2980–2988.

LOUZADA, F.; ARA, A.; FERNANDES, G. B. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science*, v. 21, p. 117–134, 2016.

MORE, A. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

MUSHAVA, J.; MURRAY, M. A novel xgboost extension for credit scoring class-imbalanced data combining a generalized extreme value link and a modified focal loss function. *Expert Systems with Applications*, v. 202, p. 117233, 2022.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

SANTOS, M. S.; SOARES, J. P.; ABREU, P. H.; ARAUJO, H.; SANTOS, J. Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [research frontier]. *IEEE Computational Intelligence Magazine*, v. 13, n. 4, p. 59–76, 2018.

SCHAPIRE, R. E. et al. A brief introduction to boosting. In: *Ijcai*. [S.l.: s.n.], 1999. v. 99, n. 999, p. 1401–1406.

THOMAS, L. C.; EDELMAN, D. B.; CROOK, J. N. *Credit Scoring and Its Applications*. [S.l.]: Society for Industrial and Applied Mathematics, 2002.

WANG, C.; DENG, C.; WANG, S. Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost. *Pattern Recognition Letters*, v. 136, p. 190–197, 2020.

WILSON, D. L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2, p. 408–421, 1972.