

Modelos de previsão de inadimplência de crédito através de machine learning: um estudo abrangendo taxa básica de juros

Thiago Rigo Werminghoff^{1*}; Elton Gean Araújo²

¹ Economista. Rua dos Douradores, 134, 2D, 1100-207, Lisboa, Portugal.

² Professor do Magistério Superior. Avenida Pedro Pedrossian, 725 – Bairro Universitário II; 79500-000 Paranaíba, MS, Brasil.

*autor correspondente: twerminghoff@proton.me

Modelos de previsão de inadimplência de crédito através de machine learning: um estudo abrangendo taxa básica de juros

Resumo

Instituições financeiras realizam suas políticas de crédito baseando-se em sistemas de análise de crédito e inadimplência, ao mesmo tempo que são restritas por regulamentações estabelecidas por órgãos de supervisão acerca de boas práticas, metodologias de cálculo, reservas financeiras, entre outros. Este trabalho vale-se dos dados de contratos de crédito disponibilizados pela Lending Club Corporation juntamente com as taxas de juros básicas presentes e futuras dos EUA vigentes à época de cada contrato para realizar o estabelecimento de cinco diferentes modelos de previsão de inadimplência dos créditos propostos, a fim de analisar suas performances quanto à correta classificação e, assim, aprimorar a lucratividade do negócio, evitando clientes que acarretem em potenciais prejuízos e reduzir valores retidos a título de créditos de liquidez duvidosa. Nenhum dos modelos apresentou-se melhor do que os demais em todas as métricas analisadas (acurácia, sensibilidade, especificidade e precisão). O modelo de Gradient Boosting (CatBoost) obteve o melhor resultado em todas as métricas resultantes do banco de dados de teste, mesmo que marginalmente, citadamente Acurácia (0,998881), Sensibilidade (0,991376), Especificidade (0,999971) e Precisão (0,999801).

Palavras-chave: risco de crédito; finanças; machine learning; modelos de crédito.

Credit default prediction models through machine learning: a study approaching basic interest rates

Abstract

Financial institutions implement their credit policies based on credit and default analysis systems, while being constrained by regulations established by supervisory bodies regarding best practices, calculation methodologies, financial reserves and other factors. This study uses data from credit contracts provided by Lending Club Corporation, along with current and future U.S. base interest rates in effect at the time of each contract, aiming to develop five different models for predicting default rates on such proposed loans, in order to analyze their performance in terms of correct classification and, thus, improve business profitability by avoiding potential losses and reducing amounts retained as default risk. None of the models performed better than the others in every metric analyzed (accuracy, sensitivity, specificity, and precision). The Gradient Boosting (CatBoost) model achieved the best result in all metrics resulting from the test database, even if marginally, namely Accuracy (0.998881), Sensitivity (0.991376), Specificity (0.999971) and Precision (0.999801).

Keywords: credit risk, finance, machine learning, credit modelling

Introdução

Bancos comerciais determinam suas práticas de crédito baseados em sistemas de credit score, desenvolvidos de modo a estimar o risco de inadimplência dos tomadores (Zeidan et al., 2015). Os autores informam que novos padrões de regulamentações seguem os Acordos de Basileia III, no sentido de que os bancos são obrigados a estimar seus parâmetros de perda: probabilidade de inadimplência, inadimplência-base e exposição à inadimplência.

Não somente a preocupação em cumprir os normativos fazem parte do dia a dia dos bancos, o rendimento de sua atividade econômica principal também está diretamente ligado a uma boa manutenção do nível de riscos assumido. Enormes perdas econômicas podem ser evitadas com apenas uma leve melhora da accuracy de um modelo de concessão de credit scoring e de risco de inadimplência (He e Fan, 2021).

A utilização de extensivos bancos de dados e a evolução dos algoritmos de machine learning facilitaram uma melhor compreensão dos dados do tomadores, permitindo a predição de probabilidades de inadimplência e subsequente análise sobre o risco na concessão de crédito (Jiang et al., 2023). Nessa linha, Yan et al. (2025) caracterizam previsão de inadimplência como a avaliação da probabilidade que um tomador não honre seu débito dentro do cronograma no futuro. Isso ajuda na redução do risco de crédito das instituições financeiras através da análise de alguns dados do tomador, como situação econômica, histórico de crédito e padrões de comportamento. Os autores ainda indicam que a metodologia de análise de risco desse ramo é dada em duas categorias principais: métodos estatísticos e machine learning.

Zeidan et al. (2015) afirmam que historicamente existe uma preferência dentre as organizações financeiras e pesquisadores pela utilização de modelos estatísticos em análise de crédito, como o exemplo da aplicação de Z-score e modelos de regressão de Altman (1968). He e Fan (2021), Wu et al. (2022) e Yan et al. (2025) também mencionam Altman (1968) como precursor da análise de risco moderna, muito embora seja importante frisar que existem estudos anteriores (Beaver, 1966; Myers e Forgy, 1963). O paper de Altman (1968) dedica-se a estudar um grupo de indicadores financeiros e econômicos particulares de cada empresa com vistas a prever um contexto de falência utilizando uma metodologia estatística de discriminantes múltiplos.

Embora alguns estudos de referência tenham sido realizados com informações de empresas, é possível fazer um paralelo com a situação econômico-financeira de pessoas físicas e desse modo inferir sobre o potencial de risco de crédito dessa pessoa. Altman (1968) menciona que uma empresa com uma margem de lucro ruim e/ou histórico de baixa solvência (capacidade de honrar seus compromissos) pode ser considerada como um potencial de alto risco de crédito. O raciocínio às pessoas físicas é semelhante: um indivíduo com baixa renda e/ou histórico de atrasos pontuais ('solvência' baixa) também pode ser considerado um potencial de alto risco de crédito.

Como as técnicas de machine learning conseguem uma boa performance em classificação de dados, metodologias como redes neurais, support vector machines e árvores de classificação/regressão se tornaram mais comuns, especialmente em sistemas de risco de crédito cada vez mais altos (Shen et al., 2021). Mesmo com as técnicas de

machine learning performando melhor do que modelos estatísticos tradicionais, os autores ressaltam que não existe um modelo de classificação único que consiga obter os melhores resultados em todos os casos.

Diversos trabalhos já foram realizados confrontando métodos estatísticos e métodos de machine learning no tocante à análise e cálculo de risco de crédito (Liu et al., 2022; Yan et al., 2025, He e Fan, 2021; Zeidan et al., 2015; Shen et al., 2021; Barboza et al., 2017, Addo et al., 2018, Bayraci e Susuz, 2019), com a grande maioria atestando melhor performance por parte dos modelos de machine learning em detrimento aos modelos estatísticos.

A literatura atual contempla diversos estudos sobre risco de crédito (seja na antes ou após a efetivação do contrato) com a utilização de redes neurais (Ribeiro e Lopes, 2011; Kvamme et al., 2018; Mai et al., 2019; Georgiou e Yannacopoulos, 2024; Addo et al., 2018; Khashman, 2010) nas mais diversas áreas financeiras (cartões de crédito, empréstimos pessoais, hipotecas, créditos a empresas, risco de falência empresarial). O estudo de Dastile et al. (2020) informa que os modelos mais frequentes utilizados para análise e cálculo de risco de crédito são regressão linear, support vector machines e redes neurais artificiais - os classificadores de deep learning estão entre os menos frequentemente utilizados.

Nos últimos anos as redes neurais tem recebido bastante atenção, também no campo de finanças. Kellner et al. (2022) nomeiam duas possíveis razões: primeiro, o poder computacional aumentou massivamente nos anos recentes e soluções open-source foram amplamente desenvolvidas - o que permite algoritmos altamente complexos ficarem à disposição de uma audiência muito ampla. Segundo, a performance superior desses algoritmos é bem documentada na literatura.

A utilização de variáveis macroeconômicas em modelos de previsão de inadimplência se mostra importante dada a relação de forte influência de variáveis macroeconômicas (crescimento econômico, desemprego, inflação, taxas de juros...) sobre o risco de crédito bancário (CHEN, 2023). Em sua obra o autor explica a forte relação positiva que existe entre o crescimento do PIB e o aumento da qualidade do crédito, uma vez que a expansão econômica tipicamente reduz o risco de inadimplência dos tomadores, então reduzindo o risco geral de crédito para dos bancos.

De forma semelhante, Naili e Lahrichii (2022) escrevem que o crescimento do PIB, taxa de desemprego, nível de capitalização bancária, performance bancária, ineficiência operacional bancária, concentração bancária, inflação, dívida soberana e tamanho dos bancos são os principais determinantes de inadimplência sobre empréstimos. Neste contexto, a inflação representa uma variável macroeconômica a ser trabalhada pelo governo

nacional normalmente valendo-se de três instrumentos principais: operações de compra e venda de títulos do tesouro nacional em mercado aberto, aperto ou afrouxamento fiscal e, diretamente ligado a este estudo proposto, aumento ou redução da taxa básica de juros.

Este trabalho pretende averiguar empiricamente a eficácia de algoritmos de machine learning (random forest, gradient boosting, LDA, SVM e redes neurais) para análise de risco de crédito especificamente na tentativa de previsão de contrato de crédito em futura situação de prejuízo no âmbito de empréstimos pessoais, utilizando os dados disponibilizados ao público pela Lending Club, uma empresa pública americana de empréstimos pessoais sediada em San Francisco, Califórnia, realizando comparações entre os modelos treinados.

Material e Métodos

Foi realizado um estudo com base em banco de dados dos empréstimos concedidos pelo Lending Club Corporation, que concede empréstimos de até US\$40mil, sem contrapartida, através de modalidade peer-to-peer (investidores e tomadores com interação direta), disponibilizado no website Kaggle¹ com a inclusão de informações referentes à *Effective Federal Funds Rate* (EFFR) na época da solicitação do empréstimo, obtida através do website do *Federal Reserve Bank of St. Louis*², também incluindo expectativas de taxas futuras (períodos de 6 e 12 meses). Para estas taxas de juros de expectativas futuras é utilizada a própria taxa efetiva futura mais um ruído aleatório de até 1pp ao ano. Este ruído é incluído em função da impossibilidade de se conhecer a taxa futura efetiva a priori.

A EFFR é estabelecida pelo *Federal Open Market Committee* (FOMC), órgão americano equivalente ao Comitê de Política Monetária (Copom) no Brasil, responsável pelas diretrizes da política monetária e, particularmente, responsável pelo estabelecimento da taxa básica de juros da economia brasileira (Selic). A EFFR representa para os Estados Unidos o que a taxa DI (depósito interbancário) representa para o Brasil: ambas servem como proxy para a taxa de juros básica de suas respectivas economias.

Em relação a um paralelo preciso com a taxa Selic, diferente do Brasil os EUA não possuem uma taxa-alvo (como é comumente conhecida a nossa taxa básica de juros), eles trabalham com piso e teto de 'referência-alvo' desde 2008³. Por questões de reprodutibilidade, opta-se então por trabalhar com a EFFR como proxy de uma taxa básica de juros, tendo em vista essa taxa ser divulgada continuamente desde 1954 até hoje.

¹ <https://www.kaggle.com/datasets/ethon0426/lending-club-20072020q1/data>

² <https://fred.stlouisfed.org/series/EFFR>

³ <https://fred.stlouisfed.org/series/DFEDTAR>

O banco de dados compreende o período de janeiro de 2007 até o final de 2017, é composto de 107 variáveis numéricas e 35 variáveis não-numéricas. com 2,92 milhões de observações no total.

Os dados são tratados em relação a se o proponente dos contratos tem histórico de renegociações (*hardships*) e se as propostas são submetidas de forma individual (apenas um mutuário) ou conjunta (normalmente um casal). Após filtragem das variáveis desnecessárias e procedimentos como o One Hot Encoding e *binning* de categorias, o banco de dados finalizado compreendeu um total de 250 variáveis explicativas, todas numéricas, prontas para serem utilizadas no treinamento e verificação dos modelos propostos, contendo um total de 2,35 milhões de observações.

Anterior ao início de treino e verificação dos modelos, foi separado um grupo de observações (15% do total) com dois objetivos: (1) redução de dimensionalidade do banco de dados de treino e teste, possibilitando a computação pela máquina local em utilização (economias de CPU e RAM); e (2) servir como base de validação dos modelos em linha de produção, ou seja, sendo utilizados no dia a dia da instituição financeira. Dos 85% restantes (2 milhões de observações), separou-se dois agregados distintos: um grupo exclusivo para treinamento dos modelos (80% desse montante) e outro grupo exclusivo para teste dos modelos (os 20% restantes).

Como modelo de regressão de referência de modelos estatísticos para análise de risco será utilizada a análise de discriminante linear (LDA – *linear discriminant analysis*), que, segundo Moscatelli et al. (2020), permite uma análise da qualidade de crédito de uma instituição utilizando uma função linear que classifica os tomadores dessa instituição em grupos distintos (inadimplentes e não-inadimplentes) com base em uma série de características particulares desses tomadores. Dastile et al. (2020) informam que a LDA desenvolve uma combinação linear de variáveis independentes que resulta na maior diferença média entre as classes, proporcionando a solução de um problema de classificação binária. Sobre modelo LDA específico para classificações em apenas duas categorias, Duda et al. (2021) informam que este pode ser também chamado de Fisher Discriminant Analysis (FDA).

Uma função discriminante linear divide o espaço de variáveis por um hiperplano de limiar de decisão (Duda et al., 2021). Estes autores escrevem que essa função discriminante é uma combinação linear dos componentes de x que pode ser escrita conforme a eq. (1):

$$g(x) = w^T x + w_0 \quad (1)$$

onde w é um vetor de pesos e w_0 é a tendência ou o limiar de discriminação (*threshold weight*). Nesse sentido, os autores afirmam que um classificador linear de duas categorias implementa uma regra de decisão para os elementos do vetor w : decide w_1 se $g(x) > 0$ e w_2

se $g(x) < 0$. Assim, x é atribuído a w_1 se o produto interno $w^t x$ excede o limiar $-w_0$ e a w_2 caso não. Se $g(x) = 0$, então x pode ser atribuído ordinariamente a qualquer classe. Desse modo, a equação $g(x) = 0$ define o limiar para decisão que separa os pontos atribuídos a w_1 dos pontos atribuídos a w_2 . Quando $g(x)$ é linear, então esse limiar para decisão é um hiperplano. De acordo com os autores, se ambos x_1 e x_2 estão neste limiar para decisão, então $w^t x_1 + w_0 = w^t x_2 + w_0$, ou $w^t (x_1 - x_2) = 0$ e isso demonstra que w é normal a qualquer vetor que situa-se no hiperplano. Em regras gerais, o hiperplano H divide o espaço de variáveis em dois semiespaços, sendo eles região R_1 para w_1 e região R_2 para w_2 . Uma vez que $g(x) > 0$, se x encontra-se em R_1 , sucede-se que o vetor normal w aponta para R_1 . Ocasionalmente refere-se que qualquer x em R_1 encontra-se no *lado positivo* de H e qualquer x em R_2 encontra-se no *lado negativo* de H .

A orientação do limiar desse hiperplano é determinada pelo vetor normal w , e as localizações do limiar é determinado pelo viés w_0 . A função discriminante $g(x)$ é proporcional à distância sinalizada de x em direção ao hiperplano, com $g(x) > 0$ quando x encontra-se no lado positivo e $g(x) < 0$ quando x está no lado negativo (Duda et al., 2021).

Os autores ainda esclarecem que a função discriminante $g(x)$ retorna uma medida algébrica da distância de x em relação ao hiperplano. Alternativamente, é possível expressar x como descrito na eq. (2):

$$x = x_p + r \frac{w}{\|w\|} \quad (2)$$

onde x_p é a projeção regular de x em H , e r é a distância algébrica desejada - positiva se x está no lado positivo e negativa se x encontra-se no lado negativo. Então, quando $g(x_p) = 0$, tem-se a eq. (3):

$$g(x) = w^t x + w_0 = r \|w\| \quad (3)$$

que, invertendo $\|w\|$ e $g(x)$, tem-se a eq. (4):

$$r = \frac{g(x)}{\|w\|} \quad (4)$$

Para os modelos de análise de risco com a utilização de algoritmos de machine learning, este trabalho propõe o desenvolvimento de modelos baseados em random forest, gradient boosting, support vector machines (SVM) e redes neurais.

No trabalho de Addo et al. (2018), os autores definem *random forest* como um agrupamento de árvores de decisão, sendo que cada árvore é desenvolvida através de um subconjunto de dados aleatoriamente escolhidos e a precisão de uma random forest

depende diretamente da força de cada árvore de decisão individual, bem como uma medida de dependência entre elas.

Cada resultado de uma decisão em um dado nó é chamado de *split* (separação, divisão), uma vez que pode ser representado utilizando apenas decisões binárias e corresponde à separação (*split*, literalmente) de um banco de dados em dois subgrupos. O primeiro nó (também chamado de nó raiz) separa a totalidade do dataset em uso, cada sucessiva decisão separa um subgrupo apropriado de dados (Duda et al., 2021). Via de regra, a quantidade máxima de splits é definida como hiperparâmetro, ou seja, definida pelo usuário, e pode variar ao longo da(s) árvore(s).

Wang et al. (2024) elencam três vantagens principais dos algoritmos de árvores de decisão/random forest: primeiro, que é possível utilizar essa modelagem tanto para regressão como para classificação, podendo se observar facilmente a importância relativa de cada variável utilizada; segundo, que os modelos de random forest são simples de usar e muito convenientes uma vez que produzam boas previsões valendo-se dos hiperparâmetros padrões; e terceiro, que os modelos de random forest raramente atingem o *overfit*, a adaptação perfeita ou quase-perfeita do algoritmo ao banco de dados de treino, uma vez que se possua uma quantidade significativa de árvores de decisão na floresta.

Contudo, o trabalho de Duda et al. (2021) informa que se os splits continuarem indefinidamente (e ao longo de todas as árvores distintas) até o ponto de menor impureza possível, então os dados sofreram *overfit*. Para estes autores, nesse caso extremo, cada folha (o nó final) da árvore corresponde a um único ponto do banco de dados de treino e a árvore como um todo não passa de apenas uma apresentação de uma tabela. Bem como se os splits forem interrompidos muito cedo então a perda (*loss*) resultante do treinamento não é suficientemente baixa e por consequência o modelo terá problemas de performance (*underfit*, a falha na adaptação do algoritmo ao banco de dados).

De acordo com Maimon e Rokach (2010), os tomadores de decisão naturalmente preferem árvores simples, menos complexas (o que corrobora com o princípio da Navalha de Occam, onde a explicação mais simples para um fenômeno é geralmente a mais correta), uma vez que essas árvores mais simples são consideradas mais compreensíveis. Segundo esses autores, a complexidade da árvore é explicitamente controlada pelo critério de interrupção e pelo método de poda empregados, ambos parâmetros estabelecidos pelo usuário nas definições dos hiperparâmetros do algoritmo utilizado. Acerca do crescimento e complexidade das árvores de decisão, Cappelli e Zhang (2007) afirmam que qualquer método baseado em árvores envolve duas etapas principais: o crescimento da árvore (desenvolvimento a partir dos splits) e a poda (*prunning*) de ramificações dessa árvore,

caracterizada como a remoção retroativa de alguns 'galhos' de forma a manter a árvore menor e mais precisa.

A etapa de crescimento é orientada pelo objetivo de formar um número de subsets homogêneos em relação à variável de resposta e, nesse sentido, formada através de splits. Maimon e Rokach (2010) informam os critérios de split de um nó, quando atestam que na maioria dos casos os critérios de split discretos são univariados, ou seja, um nó é seccionado (*split*) de acordo com o valor de um único atributo. Os critérios univariados são caracterizados de acordo com a origem da métrica utilizada (por exemplo o ganho de informação ou entropia) ou de acordo com a estrutura da métrica utilizada (o índice Gini, por exemplo, como critério de heterogeneidade ou impureza).

Considera-se um candidato a split s de um nó t em duas ramificações, t_l e t_r , e seja $p(t_l)$ e $p(t_r)$ as proporções de observações distribuídas por s a t_l e t_r , respectivamente (Cappelli e Zhang, 2007). O grau de impureza do nó t , denotado por $i(t)$, mede as impurezas baseado nas probabilidades intraclasse. Desse modo, a forma natural de se avaliar a performance de um candidato a split é a variação na impureza dado pela eq. (5):

$$\Delta i(s, t) = i(t) - [p(t_l)i(t_l) + p(t_r)i(t_r)] \quad (5)$$

a quantia $\Delta i(s, t)$ é utilizada então como o critério de partição (split).

Já para critérios multivariados para se realizar os splits, vários atributos podem participar em um único processo de split (Maimon e Rokach, 2010). De acordo com os autores, a maioria dos critérios multivariados se baseiam na combinação linear dos atributos de entrada. Nas palavras dos autores, "o processo de estimar essa combinação linear pode ser realizado utilizando *greedy search*, programação linear, LDA entre outros" (Maimon e Rokach, 2010, p. 157, tradução minha).

O trabalho de Cappelli e Zhang (2007) discorre sobre a poda das árvores de decisão, que geralmente segue-se ao crescimento da árvore em função de duas preocupações principais: (1) complexidade - a estrutura resultante tende a ser bem grande, especialmente no caso de árvores binárias uma vez que um atributo pode ter recorrência por várias vezes ao longo da árvore; e (2) overfitting - várias ramificações (galhos), especialmente as finais, refletem características particulares dos dados que surgem do processo de amostragem ao invés do treinamento.

Corroborando com essa informação, Maimon e Rokach (2010) escrevem que a fase de crescimento de uma árvore de decisão continua até encontrar um critério de interrupção. Para os autores, algumas condições são regras comuns de interrupção: (a) todas as folhas do banco de dados de treino pertencem a um único valor de y ; (b) a profundidade máxima da árvore foi atingida; (c) o número de casos no nó terminal é menor do que o número

mínimo de casos do nó anterior; (d) se o nó teve split, o número de casos em um ou mais nós-filhos é menor do que o número mínimo de casos para nós-filhos; (e) o melhor critério de split não é maior do que um dado limiar.

A utilização de critérios de interrupção de crescimento muito incisivos (apertados) tende a criar árvores de decisão pequeninas e sofrendo de underfit. O oposto também é verdadeiro: a utilização de critérios de interrupção de crescimento muito amplos (frouxos) tende a criar árvores de decisão exageradamente grandes e sofrendo de overfitting. Para esse impasse Breiman et al. (1984) sugeriram técnicas de poda (*prunning*), nas quais um critério frouxo de interrupção de crescimento é utilizado, de modo a que a árvore de decisão sofra de overfit. A partir disso a árvore então é cortada de modo a reduzir seu tamanho através da remoção de galhos que não contribuem para a acurácia de generalização.

Maimon e Rokach (2010) argumentam que outra motivação chave para a poda das árvores de decisão é a troca de acurácia por simplicidade, uma vez que o objetivo é o desenvolvimento de um conceito descritivo suficientemente preciso. No processo de poda, um nó só é efetivamente podado se a operação melhorar um certo critério. Como exemplo, os autores citam dois exemplos de poda: poda de custo-complexidade (também chamada de 'poda do elo mais fraco') e poda de redução de erro.

A poda de custo-complexidade foi apresentada no trabalho de Breiman et al. (1984) e é composta de duas etapas. Primeiro, uma sequência de árvores T_0, T_1, \dots, T_k é construída baseada nos dados de treino, onde T_0 é a árvore original antes da poda e T_k representa apenas o tronco. Segundo, uma dessas árvores é escolhida como a árvore pós-poda através da estimação de seu erro de generalização. Uma árvore $T_{(i+1)}$ é obtida substituindo-se uma ou mais das árvores predecessoras T_i com folhas disponíveis para poda. As árvores a se podar são aquelas que resultam no menor aumento da taxa de erro aparente a cada folha podada, conforme a eq. (6):

$$\alpha = \frac{\varepsilon(\text{pruned}(T, t), S) - \varepsilon(T, S)}{|\text{leaves}(T)| - |\text{leaves}(\text{pruned}(T, t))|} \quad (6)$$

onde $\varepsilon(T, S)$ representa a taxa de erro da árvore T sobre a amostra S e $|\text{leaves}(T)|$ simboliza o número de folhas em T , $\text{pruned}(T, t)$ representa a árvore obtida através da substituição do nó t em T com outra folha. Na sequência desse processo então o erro de generalização é estimado para cada árvore podada T_0, T_1, \dots, T_k e a melhor árvore podada é escolhida.

Já a poda de redução de erro foi sugerida por Quinlan (1987) e define que, enquanto percorre os nós da base da árvore de decisão até o topo, o procedimento verifica para cada nó se a substituição desse nó com a moda da variável em questão reduz ou não a acurácia

da árvore. Se não reduzir, o nó é cortado. Esse processo continua até o momento em que qualquer poda adicional resultaria na queda da acurácia.

Duda et al. (2021) escrevem que uma random forest é particularmente útil em problemas binários, uma vez que impureza zero é desejável apenas quando o nó representa somente padrões de uma única categoria. A mais simples forma polinomial dá-se pela eq. (7):

$$i(N) = P(w_1)P(w_2) \quad (7)$$

onde $P(w)$ é a quantidade percentual de padrões no nó N que encontram-se na categoria w .o que pode ser interpretado como variância de impureza, uma vez que ela é relacionada à variância de uma distribuição associada com as duas categorias.

Dastile et al. (2020) informam que o *gradient boosting* é amplamente conhecido pela sua velocidade de processamento e performance, seu funcionamento é similar ao gradient boosting (que estima múltiplos modelos iterativamente e atribui pesos a cada instância de dados) porém ele cria árvores de decisão em paralelo ao invés de serializadas. De acordo com os autores, o algoritmo começa desenvolvendo um modelo fraco (por exemplo, uma árvore de decisão rasa), em sequência um modelo melhorado tenta resolver os problemas encontrados pelo modelo anterior. As instâncias em que as classificações foram incorretas pelo modelo anterior tem novos pesos atribuídos e o processo é repetido por um número X de vezes (definido pelo usuário), onde um terceiro modelo tenta resolver os problemas do segundo e assim sucessivamente, buscando minimizar a função de perda (*loss*) definida pelo usuário nos hiperparâmetros da função.

A respeito das SVM, ou *support vector machines*, Barboza et al. (2017) escrevem que esses modelos são similares à análise discriminante linear, porém não são sujeitas a uma série de premissas, o que por consequência faz com que as SVM sejam menos restritivas. Dastile et al. (2020) informam que uma SVM utiliza uma ideia de hiperplano (que representa uma fronteira de decisão) que separa as classes em um espaço de alta dimensão das variáveis; o SVM linear tem como objetivo maximizar a margem a entre os hiperplanos negativo e positivo., conforme a eq. (8):

$$\|\alpha\| = \sqrt{\sum_{i=1}^m \alpha_i^2} \quad (8)$$

A classe correta é então atribuída de acordo com a eq. (9):

$$y = \begin{cases} +1, & \text{se } b + \alpha^T x \geq +1 \\ -1, & \text{se } b + \alpha^T x \leq -1 \end{cases} \quad (9)$$

onde b representa a tendência ou viés. Desse modo, quanto maior a margem melhor é a generalização do classificador.

Por fim, redes neurais artificiais referem-se a um sistema baseado analogamente em um sistema neural biológico - uma rede neural artificial simula o modo como o cérebro processa informações através de neurônios interconectados (Dastile et al., 2020). Nesse aspecto, Bayraci e Susuz (2019) ilustram que a configuração dessa rede neural é determinada pela maneira pela qual as camadas de neurônios encontram-se conectadas.

Dastile et al. (2020) escrevem que uma rede neural consiste em três camadas (*layers*) distintas: uma camada de entrada (*input layer*), uma camada de processamento (*hidden layer*) e uma camada de saída (*output layer*). Nos casos em que o número de hidden layers é igual ou superior a dois, a rede neural é chamada de rede neural profunda (*Deep Neural Network – DNN*) (Bayraci e Susuz, 2019). O treino de uma rede neural envolve o processo de obter-se os pesos ótimos atribuídos que mapeiam as camadas input/output através de uma mecânica chamada de retropropagação (*backpropagation*) (Dastile et al., 2020).

Duda et al. (2021) argumentam que a retropropagação é um dos meios mais simples e gerais para o treinamento de modelos supervisionados de redes neurais. A maior utilidade da retropropagação é ela permitir o cálculo de um erro efetivo para cada neurônio, e assim resulta em uma regra de aprendizagem J para os pesos, conforme observado na eq. (10):

$$J(w) = 1/2 \sum_{k=1}^c (t_k - z_k)^2 = 1/2 (t - z)^2 \quad (10)$$

onde t é o target e z é o vetor de output da rede com comprimento c ; w representa os pesos arbitrados na rede neural.

Essa regra de aprendizagem é baseada em *gradient descent* (método do máximo declive): os pesos são inicializados com valores aleatórios e são alterados na direção em que o erro é reduzido seguindo a eq. (11), representada em sua forma componente vetorial pela eq. (12):

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (11)$$

$$\Delta w_{mn} = -\eta \frac{\partial J}{\partial w_{mn}} \quad (12)$$

onde η é a taxa de aprendizagem (*learning rate*) e indica o tamanho relativo das alterações nos pesos (Duda et al, 2021). Esse algoritmo iterativo utiliza então um vetor de peso em m e procede com sua atualização de acordo com a eq. (13):

$$w(m+1) = w(m) + \Delta w(m) \quad (13)$$

onde m representa o índice de iteração em uso.

De forma simplificada, Dastile et al. (2020) explicam que para um vetor de dados x qualquer, uma rede neural simples de três camadas calcula o resultado \hat{y} de acordo com a eq. (14):

$$\hat{y} = a_2 \left(a_1 \left(\alpha^{(1)} x + \alpha_0^{(1)} \right) \alpha^{(2)} + \alpha_0^{(2)} \right) \quad (14)$$

onde $\alpha_0^{(1)}$, $\alpha^{(1)}$, $\alpha_0^{(2)}$ e $\alpha^{(2)}$ são pesos; a_1 e a_2 são as funções de ativação entre a camada de entrada e a camada de processamento. Os parâmetros de decisão são aprendidos durante a fase de treino, para a entrega efetiva de um resultado a rede neural depende de ainda uma função de decisão (também referida como *função de ativação*), como, por exemplo, softmax ou sigmoide, que é registrada na camada de saída.

Para fins de análise e comparação dos modelos, são utilizadas métricas baseadas na Matriz de Confusão, sendo elas acurácia (precisão simples de acertos), sensibilidade (*recall*), especificidade e precisão, além da Matriz de Confusão propriamente dita. Para todos os modelos treinados foi utilizado sempre o mesmo *cutoff* padrão (0,5): o *cutoff* é o valor limite de decisão em que o algoritmo classifica um problema binário em uma classe ou em outra.

Com base no estudo de classificação de Niu et al. (2020), a Figura 1 esboça a estrutura básica de uma Matriz de Confusão:

		Valores Observados	
		Verdadeiros	Falsos
Valores Estimados	Verdadeiros	Verdadeiros Positivos (TP)	Falsos Positivos (FP)
	Falsos	Falsos Negativos (FN)	Verdadeiros Negativos (TN)

Figura 1. Estrutura básica da Matriz de Confusão.

Fonte: elaboração do autor

Os cálculos das métricas utilizadas neste trabalho são definidos desde a eq. (15) até a eq.(18):

$$Precisão = \frac{TP}{TP + FP} \quad (15)$$

$$Sensibilidade = \frac{TP}{TP + FN} \quad (16)$$

$$Especificidade = \frac{TN}{TN + FP} \quad (17)$$

$$Acurácia = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

onde FP (*false positives*) são os falsos positivos; TP (*true positives*) são os verdadeiros positivos; FN (*false negatives*) são os falsos negativos; TN (*true negatives*) são os verdadeiros negativos.

Existem dois tipos básicos de erros possíveis quando da análise de uma Matriz de Confusão qualquer, definidos como erros de tipo I e erros de tipo II. O primeiro resulta do erro de classificação em que uma observação é classificada como positiva quando deveria ser corretamente classificada como negativa (falso positivo), o erro de tipo II se dá quando uma observação é classificada como negativa ao invés de ser corretamente classificada como positiva (falso negativo).

No contexto deste trabalho, então, um erro de tipo I se apresenta quando a uma proposta de crédito (observação) recebe indevidamente a classificação como provável inadimplente (que não se mostraria real), e um erro de tipo II percebe-se quando acontece a classificação como 'cliente bom', com baixo risco de inadimplência, e que se mostraria inadimplente no futuro. A depender da política econômica ou da própria governança empresarial, a instituição tem a liberdade de modificar o parâmetro de cutoff para valores diferentes do padrão, de modo que o modelo de crédito utilizado permita viés mais tendencioso a algum dos dois erros.

Outro ponto a se considerar na utilização de modelos de previsão de inadimplência e/ou de análise de risco de crédito se dá em função de regulamentações do setor financeiro. Dastile et al. (2020) fazem notar o Acordo de Basiléia II, a partir do qual os modelos de credit scoring são obrigados a serem transparentes, por exemplo. Os autores também relatam que a maior parte dos modelos de machine learning não são transparentes em sua natureza, são modelos caracterizados como *black-box*: não possuem uma explicação fácil, simples ou exata do seu funcionamento, tornando difícil a explicação de por quê um candidato tomador não teve seu empréstimo aceito. Observação que vem ao encontro de Kellner et al. (2022), que argumentam que modelos de machine learning seguidamente são tidos como problemáticos em função de sua característica *black-box*, especialmente dada a perspectiva de regulamentação das instituições.

Todo o desenvolvimento é realizado em Python, utilizando a versão 3.12.3, com o auxílio do editor de texto Visual Studio Code, desenvolvido pela Microsoft. Para o desenvolvimento em Python serão utilizados pacotes específicos de machine learning, visualização e manipulação de dados, como por exemplo (mas não restrito a) Pandas, NumPy, Matplotlib e Seaborn. Os modelos de risco de crédito serão treinados utilizando-se

os pacotes TensorFlow para redes neurais, CatBoost para gradient boosting, e Scikit-Learn para LDA, SVM e para random forest.

Após ser realizada a manipulação dos dados de modo a possibilitar o uso com os algoritmos de machine learning, as variáveis numéricas foram padronizadas através de Z-Score e as variáveis categóricas passaram pelo processo de One Hot Encoding utilizando o pacote scikit-learn.

Resultados e Discussão

Após o treino dos modelos utilizando a mesma base de dados, os resultados não indicam um melhor modelo em absoluto. Dependendo da métrica a ser considerada o melhor resultado pode pertencer a um ou outro modelo, com a exceção do modelo SVM, em que todas as métricas são inferiores aos demais resultados. Os valores obtidos em cada métrica para cada um dos modelos quando verificados sobre o banco de dados de teste podem ser verificados na Tabela 1, enquanto os valores obtidos sobre o banco de dados de validação podem ser verificados na Tabela 2.

Tabela 1. Métricas obtidas após o treinamento dos algoritmos com a utilização do banco de dados de teste

Modelo	Acurácia	Sensitividade	Especificidade	Precisão
Rede Neural	0.994564	0.966232	0.998679	0.990673
Gradient Boost	0.998881	0.991376	0.999971	0.999801
Random Forest	0.997471	0.980448	0.999943	0.999599
LDA	0.975838	0.818461	0.998693	0.989126
SVM	0.650703	0.906120	0.613610	0.254049

Fonte: Resultados originais da pesquisa

Tabela 2. Métricas obtidas após o treinamento dos algoritmos com a utilização do banco de dados de validação

Modelo	Acurácia	Sensitividade	Especificidade	Precisão
Rede Neural	0.994663	0.965931	0.998862	0.991999
Gradient Boost	0.998478	0.991566	0.999488	0.996476
Random Forest	0.997391	0.979936	0.999942	0.999592
LDA	0.975684	0.817982	0.998729	0.989476
SVM	0.649991	0.904363	0.612819	0.254468

Fonte: Resultados originais da pesquisa

A definição dos hiperparâmetros de treino para cada um dos modelo desenvolvidos neste trabalho podem ser verificadas no Apêndice A.

As melhores métricas resultantes da predição realizada sobre o banco de dados de teste correspondem todas às do modelo de Gradient Boost. Já em relação ao observado nos resultados com base no banco de dados de validação o melhor score para as métricas Acurácia e Sensitividade também correspondem ao modelo de Gradient Boost, ao passo que o modelo de Random Forest possui o melhor resultado das métricas Especificidade e Precisão.

Os valores previstos em cada campo da Matriz de Confusão de cada modelo para o banco de dados de teste encontram-se na Tabela 3. Os respectivos valores para o banco de dados de validação encontram-se na Tabela 4.

Tabela 3. Valores previstos na Matriz de Confusão para cada modelo, com base no banco de dados de teste

Modelo	Verdadeiros Positivos	Falsos Positivos	Falsos Negativos	Verdadeiros Negativos
Rede Neural	49.073	462	1.715	349.252
Gradient Boost	50.350	10	438	349.704
Random Forest	49.795	20	993	349.694
LDA	41.568	457	9.220	349.257
SVM	46.020	131.126	4.768	214.588

Fonte: Resultados originais da pesquisa

Tabela 4. Valores previstos na Matriz de Confusão para cada modelo, com base no banco de dados de validação

Modelo	Verdadeiros Positivos	Falsos Positivos	Falsos Negativos	Verdadeiros Negativos
Rede Neural	43.521	351	1.535	307.978
Gradient Boost	44.676	158	380	308.171
Random Forest	44.152	18	904	308.311
LDA	36.855	392	8.201	307.937
SVM	40.747	119.379	4.309	188.950

Fonte: Resultados originais da pesquisa

A análise da Tabela 4 permite que tiremos conclusões mais específicas sobre pequenas mas importantes variações entre os modelos que podem ser utilizadas de acordo

com políticas institucionais ou objetivos a serem seguidos por parte da instituição detentora dos modelos. Se a instituição tiver a liberdade de escolher um dentre vários modelos ao longo do tempo, pode preferir a utilização de um modelo ou outro em face ao aumento da base de clientes, de políticas de crédito expansivas ou restritivas, de controle mais acirrado dos níveis de inadimplência e assim por diante.

Por exemplo, uma maior quantidade de erros do tipo I se traduz em uma menor inadimplência geral, ao custo de abrir mão do lucro das operações erroneamente classificadas como inadimplências futuras, o que poderia ser estrategicamente utilizado em um cenário macroeconômico recessivo ou dado um índice de inadimplência já muito elevado da instituição em questão. Alternativamente, os tomadores de decisão podem preferir um viés voltado aos erro do tipo II, arriscando incorrer em um maior número de operações inadimplentes mas também aumentando sua carteira de crédito, além de potencialmente aumentar sua base de clientes em um cenário de expansão empresarial e/ou econômica.

No caso em que os normativos internos ou externos proíbam a utilização de modelos distintos ao longo do tempo (ou impossibilitem a alteração de um algoritmo de análise dentro de uma janela de tempo específica uma vez que tenha sido divulgado o uso desse algoritmo) então a instituição pode, por exemplo, redefinir os hiperparâmetros do modelo para um melhor fine-tuning de modo a que se encaixe melhor em sua necessidade imediata, parâmetros como, por exemplo, o tamanho das árvores de decisão, quantidade de iterações e nível de cut-off.

Envolvendo os níveis utilizados de cut-off, a Figura 2 a seguir exibe o gráfico das curvas ROC individuais de cada um dos cinco modelos treinados, bem como a reta de referência equivalente a uma classificação aleatória (a linha pontilhada).

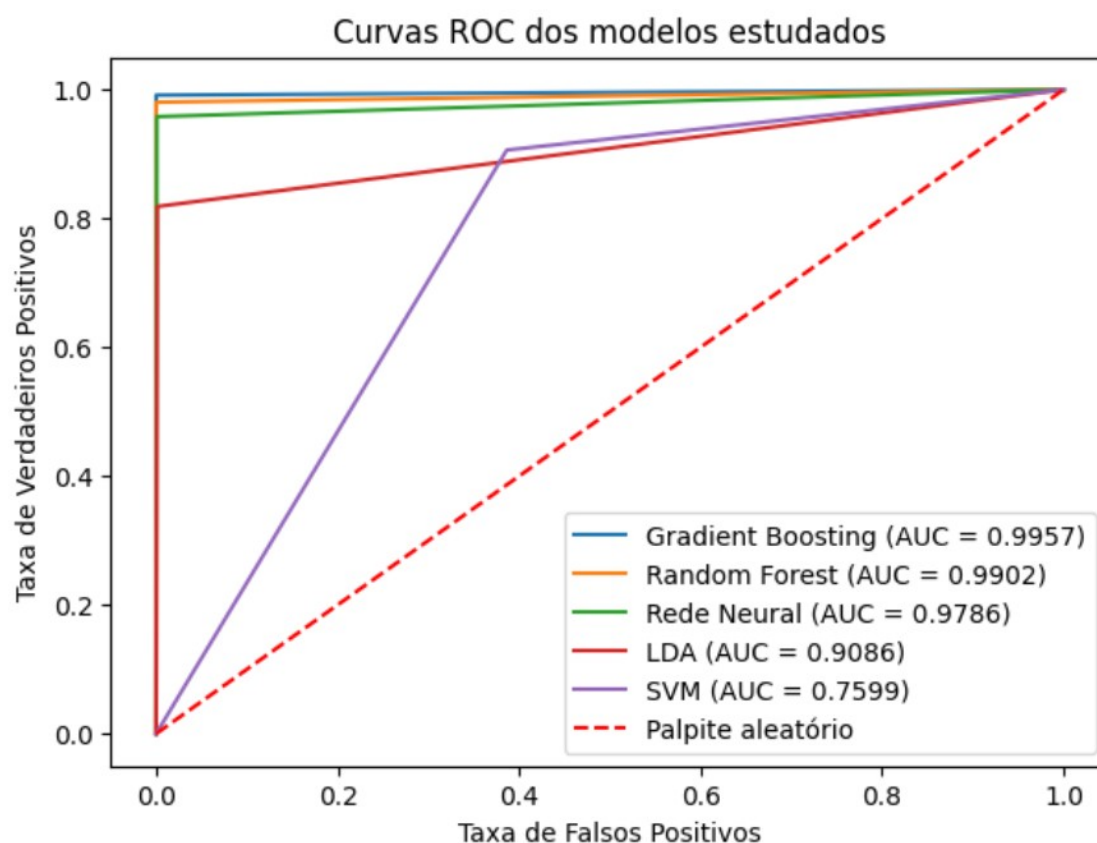


Figura 2. Gráfico das curvas ROC individuais dos modelos treinados.

Fonte: resultados originais da pesquisa

Os resultados deste trabalho corroboram parcialmente com o já apresentado em papers como Bayraci e Susuz (2019), He e Fan (2021), Jiang et al. (2023) e Kellner et al. (2022), onde redes neurais mostram-se superiores no poder de previsão de inadimplência. Contudo não foi observada uma superioridade unânime como nos resultados apresentados dos trabalhos referenciados, nas métricas tanto de Especificidade como de Precisão o modelo de gradient boost apresentou métricas mais altas neste trabalho.

Um melhor desempenho de redes neurais também é apresentado por Bayraci e Susuz (2019), que concluem que modelos de redes neurais profundas melhoram significativamente a performance de um sistema de credit scoring em relação a modelos de regressão linear e SVM no tocante às métricas de acurácia, erros do tipo I e do tipo II em bancos de dados com um número maior de observações. Contudo, o modelo de rede neural não performa melhor de forma significativa em relação aos outros modelos se o banco de dados forem pequenos e/ou restritos. Além dessa característica, os autores escrevem que os algoritmos de deep learning possuem uma implementação mais difícil e requerem um processo rigoroso de estabelecimento de seus hiperparâmetros.

Comparativamente, o resultado da métrica Acurácia deste trabalho pelo modelo de redes neurais obteve um valor 0.9945, enquanto no trabalho de Bayraci e Susuz (2019) os

autores obtiveram o valor 0.8569 para a mesma métrica e mesmo algoritmo (DNN), Barboza et al. (2017) obtiveram um resultado 0.6829 em Acurácia de redes neurais. Do mesmo modo, He e Fan (2021) aplicaram um modelo de redes neurais em dois bancos de dados distintos, obtendo os valores 0.8188 e 0.8704 para a métrica Acurácia, semelhante ao trabalho de Liu et al. (2020), que analisaram dois períodos distintos sob a ótica de um modelos de redes neurais e resultaram valores de 0.8573 e 0.7987 para a métrica Acurácia.

A possibilidade de resultados distintos é respaldada por Dastile et al. (2020), quando os autores relatam que vários estudos ao longo dos anos tem comparado e contrastado as performances de diferentes modelos estatísticos e de machine learning na seara de credit scoring mas que, entretanto, nenhum consenso foi atingido em relação à identificação de um modelo que possa performar melhor que os outros. Os autores escrevem que, de forma geral, um agregado de modelos classificadores performa melhor do que um modelo classificador único. Também que apesar das utilizações ínfimas de modelos de modelos de redes neurais na literatura de credit scoring, esses modelos apresentam melhores resultados se comparados a modelos estatísticos e modelos clássicos de machine learning (como SVM, RF ou boosting).

Addo et al. (2018) concluem que é importante a consideração de variados modelos que se encaixem com o problema de negócios e com os dados disponíveis, o que é claramente demonstrado através da apresentação da diferença nas métricas de análise. Os autores recomendam que as métricas de análise não fiquem restritas a um único critério, como por exemplo a métrica AUC (*area under the curve* - área abaixo da curva). Também notam a importância de critérios padrões (como AIC, BIC, R^2) nem sempre são apropriados para a comparação da performance de modelos distintos, dadas as diferenças das classes destes modelos (regressão/classificação).

A respeito das métricas de análise, o trabalho de Barboza et al. (2017) indica que os modelos tradicionais (regressão linear, análise discriminante) tem menor capacidade de previsão se comparados aos modelos de machine learning, os autores também sugerem novas abordagens através da adaptação das técnicas de machine learning para outros estudos de risco de crédito, não exclusivamente aos estudos relativo a falências.

No principal ensaio tido como base para o desenvolvimento de modelos de predição de inadimplência, Altman (1968) utilizou apenas um modelo LDA, relatando esse modelo como 'extremamente preciso' (Altman, 1968, pg. 609), com a correta classificação de 95% dos casos, ao mesmo tempo que sugere várias aplicações teóricas e práticas para o uso da modelagem LDA: análise de crédito, procedimentos de controles internos, diretrizes de investimento, seleção de portfólios...

Barboza et al. (2017) escrevem que embora modelos de machine learning tenham tido muito sucesso no campo das finanças, o modelo de Altman (1968) segue sendo importante, não somente pelo poder de previsão bem como pela sua estrutura simples, prática e consistente. E afirmam que embora algoritmos com base em bagging, boosting e random forest incorporem procedimentos similares, os algoritmos de random forest geralmente produzem uma melhor acurácia e taxa de erros. Mesmo assim os autores declaram que os trabalhos sobre risco de crédito - especificamente previsão de inadimplência - devem ser investigados mais a fundo, particularmente em esforços a se obter modelos relacionados a variáveis macroeconômicas, como este trabalho se propôs. Este ponto vem ao encontro do relato de Dastile et al. (2020), que informam sobre a literatura de credit scoring ignorar com frequência os processos de EDA (*exploratory data analysis*, 'análise exploratória de dados'), a inclusão de variáveis macroeconômicas e não determinam as correlações entre a variável dependente e as independentes, apenas para citar algumas limitações encontradas pelos autores.

Consoante à utilização de variáveis macroeconômicas, o trabalho de Kellner et al. (2022) indica que o comportamento sistemático de uma economia, expresso então por essas variáveis, é diferente nos Estados Unidos e na Europa, e essas variáveis macroeconômicas expressam fortes efeitos conjuntos com as demais variáveis utilizadas, reforçando a superioridade de um modelo baseado em redes neurais (o trabalho dos autores utiliza um modelo de redes neurais de regressão quantílica, *quantile regression neural networks* – QRNN).

Considerações Finais

A previsão de inadimplência é associada com o risco de crédito, que cada dia que passa exige mais atenção por parte das instituições de modo a resguardarem-se de eventuais prejuízos e, ao mesmo tempo, evitar imobilizações de capital de garantia na forma de provisão contra créditos de liquidez duvidosa. Novos modelos de machine learning se mostram muito úteis em utilizações no campo das finanças, com papers e trabalhos acadêmicos verificando sua eficácia no tocante à previsão de inadimplência, conforme observado ao longo deste trabalho. Não menos importantes, os modelos de previsão que valem-se da estatística clássica (regressão linear, LDA) compõe o dia-a-dia das instituições financeiras no que tange o desenvolvimento de modelos preditivos relacionados à sua atividade financeira. Por exemplo, o modelo sugerido por Altman (1968) há quase seis décadas ainda segue relevante, não somente pelo seu poder preditivo como também pela sua estrutura simples, prática e consistente.

Abordagens como as exercitadas ao longo deste trabalho podem ser de grande valia para instituições financeiras nascentes e/ou em consolidação, uma vez que nessas empresas a estratégia de concessão de crédito nem sempre é baseada em dados concretos e muitas vezes depende da análise subjetiva de um gerente ou responsável pela área/carteira. Esse tipo de abordagem pode potencialmente acabar em grandes prejuízos para a instituição durante o tempo de maturação dessas operações de crédito concedidas indiscriminadamente, sem uma decisão baseada em indicadores e modelos de previsão com alta efetividade. Grandes instituições (os grandes bancos comerciais, por exemplo) já possuem em sua cultura corporativa o gerenciamento de risco baseado em dados, através da utilização de modelos semelhantes aos apresentados nesta obra.

As melhores métricas de referência consideradas neste trabalho foram resultado de um mesmo modelo, Gradient Boosting. Contudo, isso não invalida os demais modelos, tendo em vista, inclusive, que os modelos de redes neurais, random forest e LDA exibiram métricas apenas marginalmente inferiores àquelas do modelo de gradient boosting. O ajuste individual dos hiperparâmetros de cada algoritmo pode ser a solução para elevar os resultados desses dois modelos que mostraram métricas levemente inferiores. O modelo SVM, dadas as condicionantes deste estudo, apresentou métricas insatisfatórias.

Deste modo, percebe-se que pode não existir um modelo ótimo quando estuda-se modelos de previsão de risco de crédito mantendo-se fixos os bancos de dados de observações para treino, teste e validação. Cada banco de dados possui características próprias em função de seu tamanho, quantidade de variáveis, variabilidade dos dados, o que invariavelmente resultará em métricas diferentes durante a fase de análise dos modelos, bem como cada algoritmo possui características intrínsecas (seu método de cálculo) e extrínsecas (seus hiperparâmetros) que se traduzirão em modelos individuais após treino.

Cada modelo, em função da métrica utilizada como referência, pode se comportar de forma mais otimizada do que outro. Existem diferenças não só nas métricas utilizadas como potencialmente também ocorrerão diferenças ao longo do tempo, uma vez que novas referências, novos valores e novos eventos são observados nos bancos de dados das instituições financeiras no passar do tempo, tornando indispensável o monitoramento e a melhora contínua dos modelos de risco no decurso do tempo e das diversas situações econômico-financeiras que se apresentarem.

Agradecimento

Meus sinceros agradecimentos ao meu orientador, prof. Elton Gean Araújo, que durante nosso pequeno percurso juntos sempre me estimulou a produzir mais conhecimento e sempre conhecimento mais acessível.

Referências

Addo, P.M.; Guegan, D.; Hassani, B. 2018. Credit risk analysis using machine and deep learning models. Working Papers 2018(08). Department of Economics, University of Venice "Ca' Foscari". Disponível em: <<https://ideas.repec.org/p/ven/wpaper/201808.html>>. Acesso em: 27 fev. 2025.

Altman, E.I. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance 23(4): 589-734. Disponível em: <<https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1540-6261.1968.tb00843.x>>. Acesso em: 26 fev. 2025.

Barboza, F.; Kimura, H.; Altman, E.I. 2017. Machine learning models and bankruptcy prediction. Expert Systems with Applications (83). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417417302415>>. Acesso em: 24 fev. 2025.

Bayraci, X.; Susuz, C. 2019. A deep neural network (DNN) based classification model in application to loan default prediction. Theoretical and Applied Economics 26(4): 75-84. Disponível em: <https://www.ebsco.ectap.ro/Theoretical_&_Applied_Economics_2019_Winter.pdf#page=75>. Acesso em: 24 fev. 2025.

Beaver, W.H. 1966. Financial ratios as predictors of failure. Journal of Accounting Research 4: 71-111. Disponível em: <<https://www.jstor.org/stable/2490171>>. Acesso em: 28 fev. 2025.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. 1984. Classification and Regression Trees. 2a Edição. Editora CRC Press, Boca Raton, Flórida, Estados Unidos da América.

Chen, Y. 2023. The Impact of Macroeconomic Factors on Bank Credit Risk. Advances in Economics Management and Political Sciences 53(1): 79-84. Disponível em:

<https://www.researchgate.net/publication/376132265_The_Impact_of_Macroeconomic_Factors_on_Bank_Credit_Risk>. Acesso em: 15 ago. 2025.

Dastile, X.; Celik, T.; Potsane, M. 2020. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing Journal* 91. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494620302039?via%3Dihub>>. Acesso em: 27 fev. 2025.

Duda, R. O.; Hart, P. E.; Stork, D. G.. 2001. *Pattern Classification*. 2a Edição. Editora Wiley-Interscience, Hoboken, Nova Jersey, Estados Unidos da América.

Georgiou, K; Yannacopoulos, A.N. 2024. Deep neural networks for probability of default modelling. *Journal of Industrial and Management Optimization* 20(12). Disponível em: <https://www.researchgate.net/publication/379054682_DEEP_NEURAL_NETWORKS_FOR_PROBABILITY_OF_DEFAULT_MODELLING>. Acesso em: 26 fev. 2025.

He, H.; Fan, Y. 2021. A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction. *Expert Systems with Applications* 176. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417421003407>>. Acesso em: 24 fev. 2025.

Jiang, Z.; Su, J.; Zhou, L. 2023. Credit default prediction based on genetic algorithm improved neural network. In: 9th International Conference on Systems and Informatics (ICSAI) 2023. Anais... Disponível em: <<https://ieeexplore.ieee.org/abstract/document/10423366>>. Acesso em: 24 fev. 2025.

Kellner, R.; Nagl, M.; Rosch, D. 2022. Opening the black box - quantile neural networks for loss given default prediction. *Journal of Banking & Finance* (134). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378426621002855>>. Acesso em: 24 fev. 2025.

Khashman, A. 2010. Neural networks for credit risk evaluation: investigation of different neural models and learning schemes. *Expert Systems with Applications* 37(9). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417410001405>>. Acesso em: 27 fev. 2025.

Kvamme, H.; Sellereite, N.; Aas, K.; Sjørusen, S. Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications* 102: 207-217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417418301179>>. Acesso em: 28 fev. 2025.

Liu, J.; Zhang, S.; Fan, H. 2022. A two-stage hybrid credit risk prediction model based on XGBoost and graph-based deep neural network. *Expert Systems with Applications* (195). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417422001142>>. Acesso em: 25 fev. 2025.

Mai, F.; Tian, S.; Lee, C.; Ma, L. 2019. Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research* 274(2): 743-758. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221718308774>>. Acesso em: 28 fev. 2025.

Maimon, O.; Rokach, L. 2010. *Data Mining and Knowledge Discovery Handbook*. 2a Edição. Editora Springer Science+Business Media, Nova York, Nova York, Estados Unidos da América.

Myers, J.H.; Forgy, E.W. 1963. The development of numerical credit evaluation systems. *Journal of the American Statistical Association* 58(303): 799-806. Disponível em: <<https://pages.ucsd.edu/~aronatas/project/academic/Comparison%20of%20Discriminant%20and%20Regression%20analysis%20for%20cred.pdf>>. Acesso em: 28 fev. 2025.

Naili, M.; Lahrichi, Y. 2022. Banks' credit risk, systematic determinants and specific factors: recent evidence from emerging markets. *Heliyon* 8(2). Disponível em: <<https://pmc.ncbi.nlm.nih.gov/articles/PMC8886134/>>. Acesso em: 15 ago. 2025.

Niu, J.; An, G.; Gu, Z.; Li, P.; Liu, Q.; Bai, R.; Sun, J.; Du, Q. 2020. Analysis of sensitivity and specificity: precise recognition of neutrophils during regeneration of contused skeletal muscle in rats. *Forensic Sciences Research* 7: 1-10. Disponível em: <https://www.researchgate.net/publication/340034692_Analysis_of_sensitivity_and_specificity_precise_recognition_of_neutrophils_during_regeneration_of_contused_skeletal_muscle_in_rats>. Acesso em 12 jun. 2025.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27(3): 221-234. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0020737387800536>>. Acesso em 16 jun. 2025.

Ribeiro, B.; Lopes, N. 2011. Deep belief networks for financial prediction. In: *Neural Information Processing (ICONIP) 2011. Anais...* p.766-773. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-24965-5_86>. Acesso em: 26 fev. 2025.

Shen, F.; Zhao, X; Kou, G; Alsaadi, F.E. 2021. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing Journal* 98. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494620307900>>. Acesso em: 27 fev. 2025.

Wang, J.; Xu, J.; Cheng, Q.; Kumar, R. 2024. Research on finance Credit Risk Quantification Model Based on Machine Learning Algorithm. *Academic Journal of Science and Technology* 10(1). Disponível em: <<https://pdfs.semanticscholar.org/07ac/c995c9288498666608b157362d8be2df7123.pdf>>. Acesso em: 02/06/2025.

Wu, D.; Ma, X.; Olson, D. 2022. Financial distress prediction using integrated z-score and multilayer perceptron neural networks. *Decision Support Systems* (159). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167923622000859>>. Acesso em: 24 fev. 2025.

Yan, Z.; Qu, H.; Chen, C.; Lv, X.; Zuo, E.; Wang, K.; Cai, X. 2025. WIGNN: An adaptive graph-structured reasoning model for credit default prediction. *Engineering Applications of Artificial Intelligence* 139(B). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095219762401755X>>. Acesso em: 25 fev. 2025.

Zeidan, R.; Boechat, C.; Fleury, A. 2015. Developing a sustainability credit score system. *Journal of Business Ethics* 127: 283-196. Disponível em: <<https://link.springer.com/article/10.1007/s10551-013-2034-2>>. Acesso em: 26 fev. 2025.

Apêndice A. Hiperparâmetros de Treinamento dos Modelos e Algoritmos Utilizados.

Modelo de Rede Neural (TensorFlow):

```
classif_nn = tf.keras.Sequential()
classif_nn.add(tf.keras.layers.Dense(64, activation='relu', name='Input_Dense64',
input_shape=(treino_x.shape[1],)))
classif_nn.add(tf.keras.layers.Dropout(0.3, seed=1))
classif_nn.add(tf.keras.layers.Dense(32, activation='relu', name='Hidden1_Dense64',
input_shape=(treino_x.shape[1],)))
classif_nn.add(tf.keras.layers.Dense(1, activation = 'sigmoid', name =
'Output_Dense1'))
classif_nn.compile(loss = categorical_hinge, optimizer = 'Adam', metrics=
['categorical_hinge'])
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
```

Modelo de Gradient Boosting (CatBoost):

```
CatBoostClassifier(loss_function='Logloss', eval_metric = 'Logloss', iterations = 1000,
learning_rate = 0.01, random_seed = 1, bootstrap_type = 'MVS', bagging_temperature = 7,
depth = 10, early_stopping_rounds = 500, thread_count = 12, task_type = 'CPU',
gpu_ram_part = 0.2, target_border = 0.5, grow_policy = 'Lossguide', min_child_samples =
15, max_leaves = 20, boosting_type = 'Plain', score_function = 'L2')
```

Modelo de Random Forest (scikit-learn):

```
RandomForestClassifier(n_estimators = 1000, criterion = 'entropy', max_depth = 21,
min_samples_split = 6, min_samples_leaf = 1, max_features='sqrt', bootstrap = True,
oob_score = False, max_samples = 0.4, n_jobs = 12, random_state = 1, verbose = 2,
warm_start = False)
```

Modelo de LDA (scikit-learn, LinearDiscriminantAnalysis)

```
LinearDiscriminantAnalysis()
```

Modelo de SVM (scikit-learn, SVC)

```
SVC(verbose=True, max_iter=1000, random_state=1)
```