Análise Comparativa de Algoritmos de Machine Learning para Classificação de Risco de Crédito

Shelda Azevedo Tenorio Ribeiro
Ciência da Computação
UNIMA/Afya
Maceió, Brasil
sheldaazevedo@gmail.com

Ramon Freire Alencar
Ciência da Computação
UNIMA/Afya
Maceió, Brasil
ramonfalencar@gmail.com

***Abstract*—This paper presents a comparative analysis of machine learning algorithms applied to credit risk classification, a critical task for financial stability. Using a real-world bank dataset ("p34"), we evaluated the performance of Random Forest and Multilayer Perceptron (MLP) models. The methodology incorporated robust data preprocessing and Stratified K-Fold Cross-Validation to ensure statistical reliability. Results indicate that the Random Forest model achieved superior performance with an accuracy of 88.24% and, crucially, a higher recall for the minority class compared to the MLP (87.76%). The study demonstrates that ensemble methods based on decision trees offer a more effective and stable solution for credit scoring in this domain compared to standard neural network architectures.**

Keywords— machine learning, credit scoring, random forest, neural networks, multilayer perceptron, cross-validation.

I. Introdução

A classificação de risco de crédito é uma tarefa crítica para instituições financeiras, permitindo a distinção entre clientes propensos a honrar seus compromissos ("bons pagadores") e aqueles com alto risco de inadimplência ("maus pagadores"). A precisão nessa classificação impacta diretamente a rentabilidade e a gestão de risco dos bancos.

Este trabalho tem como objetivo aplicar e comparar técnicas de Machine Learning para resolver esse problema de classificação binária. Utilizando uma base de dados real de financiamento de veículos com 35.331 instâncias, exploramos a eficácia de dois modelos distintos: Floresta Aleatória (Random Forest), um método de ensemble robusto, e uma Rede Neural MLP (Multilayer Perceptron). O estudo abrange desde o pré-processamento dos dados até a análise crítica das métricas de desempenho, como acurácia, precisão e matriz de confusão.

II. Fundamentação Teórica

A. Random Forest

O algoritmo Random Forest [1] é um método de aprendizado supervisionado baseado em ensemble, que constrói múltiplas árvores de decisão durante o treinamento. A decisão final é obtida pela moda das classes (classificação) ou média das previsões (regressão) das árvores individuais. Essa abordagem corrige o hábito das árvores de decisão de se ajustarem excessivamente ao conjunto de treinamento (overfitting), proporcionando maior generalização e robustez.

B. Rede Neural MLP

O Multilayer Perceptron (MLP) [3] é uma classe de rede neural artificial feedforward. Um MLP consiste em pelo menos três camadas de nós: uma camada de entrada, uma camada oculta e uma camada de saída. Exceto pelos nós de entrada, cada nó é um neurônio que utiliza uma função de ativação não linear. O MLP utiliza uma técnica de aprendizado supervisionado chamada backpropagation [2] para treinamento, sendo capaz de distinguir dados que não são linearmente separáveis.

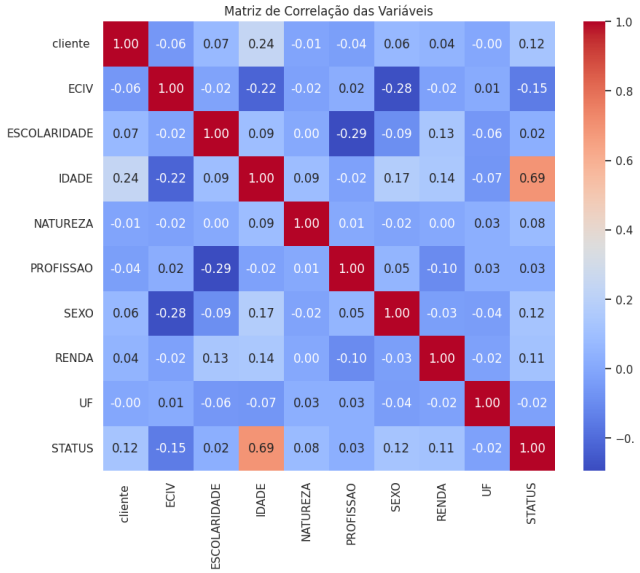
III. Metodologia

A metodologia adotada neste trabalho segue uma abordagem quantitativa experimental para a resolução do problema de classificação binária de risco de crédito. O fluxo de trabalho foi estruturado em quatro etapas principais: (1) aquisição e análise exploratória dos dados, para compreensão das variáveis e distribuição das classes; (2) pré-processamento, visando a limpeza e adequação dos dados para os algoritmos; (3) configuração e treinamento dos modelos de *Machine Learning*; e validação cruzada; e (4) avaliação de desempenho com base em métricas estatísticas. O estudo utilizou o *dataset* "p34", referente a clientes bancários, comparando a eficácia de modelos baseados em árvores de decisão (*ensemble*) e redes neurais artificiais.

A. Dataset

O estudo utilizou a base de dados "p34", composta por 35.331 instâncias referentes a clientes que receberam financiamento bancário. O dataset contém 10 colunas, incluindo atributos categóricos (Estado Civil, Escolaridade, Sexo, UF) e numéricos (Idade, Renda, Natureza da Ocupação, Profissão). A variável alvo é o "STATUS", indicando a classificação do cliente como bom ou mau pagador.

Adicionalmente, foi conduzida uma análise exploratória para compreender as relações lineares entre as variáveis numéricas do estudo. A matriz de correlação permitiu verificar a intensidade das associações entre os atributos. A análise preliminar indicou que não há multicolinearidade severa redundante entre as variáveis preditoras principais, justificando a manutenção das *features* para o treinamento dos modelos.



B. *Pré-processamento*

Os dados foram processados utilizando a linguagem Python e bibliotecas como *Pandas* e *Scikit-learn* [3]. As etapas incluíram:

1. *Limpeza:* Remoção de colunas irrelevantes ("cliente") e verificação de consistência.
2. *Codificação:* Transformação de variáveis categóricas em numéricas via *Label Encoding*. A variável alvo "STATUS" foi mapeada para binário (0/1).
3. *Normalização:* Aplicação de *StandardScaler* nas variáveis de entrada para otimizar o gradiente descendente da Rede Neural.

C. *Configuração do Experimento*

A estratégia de validação adotada consistiu na divisão dos dados em conjuntos de treino (80%) e teste (20%) via método *holdout* estratificado. Adicionalmente, durante a fase de treinamento, aplicou-se a Validação Cruzada Estratificada (*Stratified K-Fold*) com 5 dobras. Essa técnica garante a robustez dos estimadores, mitigando riscos de *overfitting* e assegurando que o desempenho do modelo não seja enviesado por uma divisão de dados específica.

Os algoritmos foram configurados com os seguintes hiperparâmetros, definidos para equilibrar desempenho e custo computacional:

- **Modelo 1 (Random Forest):** O modelo foi instanciado com 100 estimadores (n_estimators=100), utilizando o critério de Gini para as divisões das árvores e semente aleatória (*random_state=42*) para reprodutibilidade.
- **Modelo 2 (Rede Neural MLP):** Configurou-se um *Multilayer Perceptron* com duas camadas ocultas contendo 100 e 50 neurônios, respectivamente (hidden_layer_sizes=(100, 50)). O treinamento utilizou o otimizador Adam, função de ativação ReLU e um limite máximo de 500 iterações (max_miter=500) para garantir a convergência.

A avaliação de desempenho focou na Acurácia global, mas também priorizou o *Recall* e o *F1-Score*, métricas cruciais para cenários de risco de crédito onde a identificação correta dos "maus pagadores" (classe minoritária) é prioritária.

IV. PROPOSTA E IMPLEMENTAÇÃO

A. *Justificativa para a Seleção dos Modelos*

A escolha dos algoritmos foi guiada pela natureza específica dos dados financeiros tabulares.

Random Forest (RF): Selecionado por sua natureza de ensemble, conforme proposto por Breiman [1]. O RF reduz o risco de overfitting inerente a árvores de decisão individuais ao fazer a média de múltiplas árvores profundas treinadas em diferentes partes do mesmo conjunto de treinamento. É particularmente eficaz para dados de alta dimensionalidade e requer menos pré-processamento [2].

Multilayer Perceptron (MLP): Selecionado para representar modelos conexionistas baseados em Rumelhart et al. [3]. O MLP é um aproximador universal capaz de capturar relações complexas e não lineares entre variáveis por meio de retropropagação, embora seja sensível à normalização das variáveis.

B. *Detalhes de Implementação*

A implementação foi realizada em Python usando a biblioteca Scikit-Learn [3]. O pipeline experimental seguiu estas etapas:

Pré-processamento: A variável-alvo *STATUS* foi codificada (0/1). As variáveis categóricas foram convertidas usando *Label Encoding*. Para o modelo MLP especificamente, todas as variáveis de entrada foram normalizadas usando *StandardScaler* para facilitar a convergência do gradiente descendente.

Estratégia de Treinamento: Para mitigar viés de seleção, foi empregado *Stratified K-Fold Cross-Validation* (k = 5) durante a fase de treinamento. Isso garantiu que a distribuição das classes permanecesse consistente em todas as dobras.

Hiperparâmetros:

- RF: Configurado com *n_estimators = 100* e critério de impureza de Gini, equilibrando custo computacional e estabilidade.
- MLP: Configurado com duas camadas ocultas (100, 50), função de ativação ReLU e otimizador Adam, com máximo de 500 iterações.

V. Resultados

O desempenho dos modelos foi avaliado em um conjunto de teste separado, composto por 20% dos dados originais (7.067 instâncias). A Tabela I resume as principais métricas de desempenho.

>>> TABLE I. PERFORMANCE COMPARISON OF ALGORITHMS

Modelo	Accuracy	Precision (Weighted)	Recall (Bad Payer)	F1-Score (Weighted)
Random Forest	0.8824	0.8800	0.8200	0.8800
Rede Neural MLP	0.8776	0.8800	0.7800	0.8800

O modelo Random Forest obteve uma acurácia ligeiramente maior e demonstrou melhor capacidade de identificar a classe positiva (inadimplentes), evidenciada pelo Recall de 0.82, comparado a 0.78 do MLP.

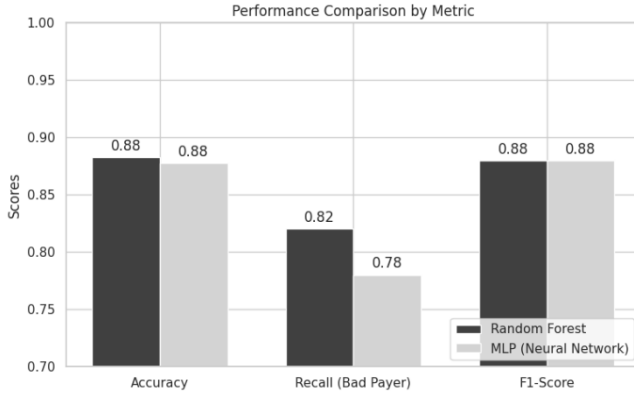
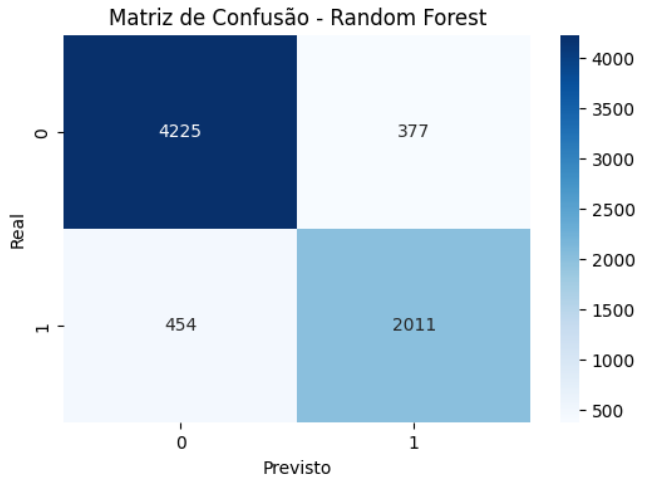


Fig. 3. Comparative metrics between Random Forest and MLP.

A. *Random Forest*

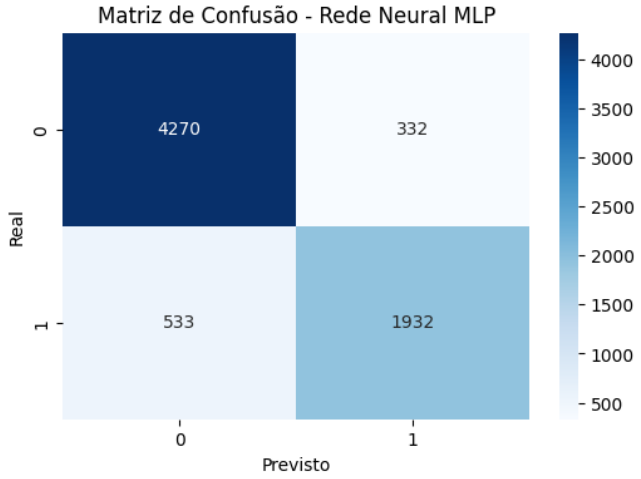
O modelo *Random Forest* apresentou o melhor desempenho global, alcançando uma Acurácia de 88,24%. Conforme o relatório de classificação, o modelo obteve um *Recall* de 0.92 para a classe majoritária (0 - Bons pagadores) e 0.82 para a classe minoritária (1 - Maus pagadores). O *F1-Score* balanceado foi de 0.88.

Isso indica que o modelo é bom não apenas em acertar a média, mais especificamente em identificar corretamente os maus pagadores, o que é crucial para a mitigação de risco de crédito.



B. *Rede Neural MLP*

A Rede Neural obteve uma Acurácia de 87,76%, um resultado muito próximo, porém ligeiramente inferior ao Random Forest. Embora tenha tido um desempenho excelente na classe majoritária (*Recall* de 0.93), seu desempenho na identificação de maus pagadores foi inferior, com um *Recall* de 0.78 (contra 0.82 do Random Forest).



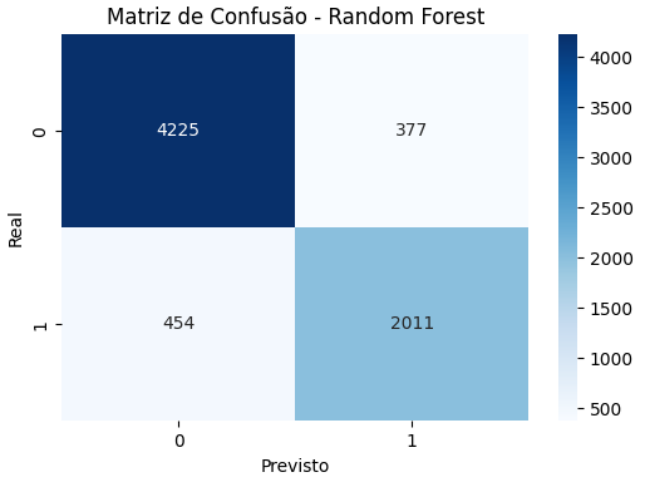
VI. Discussão

.

A. *Interpretação dos Resultados*

Ambos os modelos demonstraram alta competência na tarefa de classificação, com acurácias superiores a 87%. Entretanto, o Random Forest [1] mostrou-se mais adequado para este conjunto de dados específico. O Recall superior (0.82 vs 0.78) indica que o conjunto de árvores foi mais eficaz em capturar os padrões dos “maus pagadores” do que a rede neural.

No contexto de risco de crédito, deixar de identificar um mau pagador (Falso Negativo) é financeiramente mais prejudicial do que negar crédito a um bom pagador (Falso Positivo); portanto, o maior recall do Random Forest o torna o modelo operacionalmente preferível.



B. *Análise Crítica e Limitações*

Embora o MLP [3] tenha apresentado bom desempenho, seu resultado ligeiramente inferior pode ser atribuído à natureza dos dados tabulares, nos quais modelos baseados em árvores frequentemente superam redes neurais, a menos que seja aplicada engenharia de atributos extensa.

Uma limitação deste estudo é o desbalanceamento de classes inerente a bases de dados de crédito. Embora tenha sido utilizada Validação Cruzada Estratificada para mitigar isso durante o treinamento, as métricas sugerem que a classe "mau pagador" é mais difícil de prever. Trabalhos futuros devem investigar técnicas de amostragem como SMOTE para melhorar o recall da classe minoritária.

Além disso, quanto à deployabilidade, o Random Forest oferece interpretabilidade por meio da importância das variáveis, um requisito regulatório em muitos sistemas bancários, enquanto o MLP funciona como uma caixa-preta.

VII. Conclusão

A análise comparativa demonstrou que ambos os algoritmos são viáveis para a classificação de risco de crédito na base de dados "p34". O modelo Random Forest destacou-se como a melhor opção, superando a Rede Neural tanto em acurácia global (88,24% contra 87,76%) quanto, mais importante, na capacidade de recuperar instâncias da classe de inadimplentes (*Recall* de 82%).

Essa diferença sugere que, para este problema específico, a estrutura de *ensemble* baseada em árvores lidou melhor com as características dos dados do que a topologia da rede neural testada. Para trabalhos futuros, sugere-se a aplicação de técnicas de balanceamento de classes para tentar elevar ainda mais a detecção da classe minoritária.

VIII. Referencias

[1] L. Breiman. "Random Forests". Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[2] D. E. Rumelhart, G. E. Hinton, e R. J. Williams. "Learning representations by back-propagating errors". Nature, vol. 323, pp. 533–536, 1986.

[3] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[4] IBM. "O que é random forest?". Disponível em: https://www.ibm.com/br-pt/think/topics/random-forest. Acesso em: 29 nov. 2025.

[5] IBM. "O que é regressão logística?". Disponível em: https://www.ibm.com/br-pt/think/topics/logistic-regression. Acesso em: 29 nov. 2025.

[6] P. C. T. Gomes. "Pré-Processamento de Dados: Técnicas e Etapas de Tratamento". DataGeeks, 2019. [Online].