Detecção de Fraudes em Transações Financeiras: Uma Abordagem de Machine Learning para Ligia

Guilherme José
Universidade Federal de Pernambuco (UFPE)
Recife, PE, Brasil

***Resumo*—A detecção de fraudes em transações financeiras representa um desafio crítico devido à natureza altamente desbalanceada dos dados e à constante evolução das táticas fraudulentas. Este relatório descreve a jornada de pesquisa e desenvolvimento de uma solução de Machine Learning para o desafio técnico da Liga de IA da UFPE (Ligia). O estudo abrange desde a exploração inicial e tratamento de dados anônimos até a engenharia de atributos temporais e a avaliação de múltiplos algoritmos. O objetivo principal é maximizar a métrica ROC-AUC, garantindo alta detecção da classe minoritária sem comprometer a precisão global do sistema.**

***Index Terms*—Detecção de Fraude, Machine Learning, Dados Desbalanceados, Análise Exploratória, Redes Neurais.**

I. Introdução

O crescimento das transações financeiras digitais trouxe consigo um aumento proporcional nas tentativas de fraude, exigindo sistemas de segurança cada vez mais sofisticados. No contexto de cartões de crédito e pagamentos eletrônicos, o problema de detecção de fraudes é classicamente modelado como uma tarefa de classificação binária, onde o principal obstáculo é o extremo desbalanceamento das classes: transações legítimas ocorrem em volume massivamente superior às fraudulentas.

Este trabalho documenta a solução desenvolvida para o desafio de Machine Learning da Ligia. A relevância da tarefa reside na necessidade de criar um modelo capaz de identificar padrões anômalos sutis sem gerar um excesso de falsos positivos, o que prejudicaria a experiência do usuário com bloqueios indevidos de cartão. A abordagem foca na extração de características temporais relevantes e no treinamento de um modelo robusto, avaliado pela métrica ROC-AUC.

II. Metodologia

A estratégia de desenvolvimento dividiu-se em análise exploratória, engenharia de características e modelagem preditiva. O conjunto de dados utilizado compreende variáveis contínuas anonimizadas via Análise de Componentes Principais (*V1-V28*), *Time* (segundos), *Amount* (valor) e *Class* (variável alvo).

A. Inspeção Inicial e Tratamento de Outliers

A inspeção inicial confirmou a integridade da base, sem dados faltantes ou duplicados. Para a detecção de valores atípicos, empregou-se o método estatístico do Intervalo Interquartil (IQR). A análise revelou uma quantidade substancial de *outliers* nas componentes principais e no valor financeiro (*Amount*). Contudo, uma investigação aprofundada ao cruzar estes pontos fora da curva com a variável alvo revelou um padrão fundamental: a proporção de transações fraudulentas aninhadas nesses *outliers* era significativamente superior à proporção encontrada na distribuição normal da base.

Como fraudes são inerentemente anomalias comportamentais (frequentemente manifestadas como valores atípicos em horários incomuns), a remoção tradicional de *outliers* excluiria instâncias vitais da classe minoritária, destruindo o sinal preditivo. Assim, estabeleceu-se a premissa de manter 100% dos registros atípicos, utilizando algoritmos de escalonamento robustos posteriormente para mitigar distorções geométricas.

B. Análise Multivariada e Redução de Dimensionalidade

Para compreender a interação sistêmica entre as características, calculou-se a matriz de correlação com método de Pearson. Notou-se que as variáveis anonimizadas correlacionavam-se consideravelmente com o valor da transação, e isolou-se as *features* com maior correlação linear com a classe alvo (notavelmente *V17, V14* e *V12*).

A projeção de redução de dimensionalidade via algoritmo t-SNE (ver Figura 1 no Anexo) revelou que uma parcela significativa das fraudes pode ser separada em *clusters* definidos, mas expôs densas áreas de sobreposição com transações normais, justificando o uso posterior de modelos não-lineares.

Em relação à variável temporal, análises de distribuição KDE (ver Figura 2 no Anexo) demonstraram que as fraudes evitam valores altíssimos para burlar sistemas de limite, ocorrendo proporcionalmente mais durante o período da madrugada.

C. Engenharia de Características (Feature Engineering)

A variável linear *Time* foi convertida para a hora do dia (*Hour*). Para garantir que os algoritmos compreendessem a adjacência cronológica entre as extremidades do dia (23h e 00h), aplicaram-se transformações trigonométricas gerando variáveis cíclicas:

$$Hour\_sin = \sin\left(\frac{2\pi \times Hour}{24}\right) \tag{1}$$

$$Hour\_cos = \cos\left(\frac{2\pi \times Hour}{24}\right) \tag{2}$$

A representação linear (*Hour*) foi mantida junto às componentes cíclicas para fornecer tanto a progressão linear quanto a representação bidimensional do ciclo.

D. Preparação de Dados e Seleção de Atributos

Para garantir reprodutibilidade (*random_state*=42) e evitar vazamento de dados (*data leakage*), dividiu-se os dados em Treino (80%) e Validação (20%) via particionamento estratificado, preservando a proporção original de fraudes. Aplicou-se o *RobustScaler* (ajustado apenas no treino) para normalizar os dados, mitigando a distorção por *outliers*.

Adotou-se uma filtragem híbrida de atributos (ver Figura 3 no Anexo). As variáveis *Time* e *id* foram removidas. Embora *Hour_sin* tenha ficado abaixo da nota de corte linear, foi forçosamente mantida para preservar a integridade matemática do par vetorial temporal. A categorização manual de valores financeiros (*Amount_Risk_Level*) foi descartada por não agregar performance.

E. Modelagem Preditiva

Para garantir ampla cobertura analítica, seis algoritmos foram selecionados: Regressão Logística, Random Forest, XGBoost, LightGBM, CatBoost e Rede Neural (MLPClassifier). O treinamento ocorreu de forma padronizada via Validação Cruzada Estratificada (k=5).

A otimização dos modelos foi conduzida através do RandomizedSearchCV, com foco na maximização da área sob a curva ROC-AUC. Diversos hiperparâmetros foram testados, incluindo taxas de aprendizado, profundidade de árvores, regularizações do tipo ElasticNet e diferentes arquiteturas de rede neural, como camadas ocultas de 100 e 50 neurônios. Para atender às exigências de transparência e interpretabilidade em contexto corporativo, definiu-se posteriormente a aplicação da biblioteca SHAP para análise explicativa das predições.

III. Resultados e Discussão

A. Desempenho e Matriz de Decisão

A Tabela I avalia as previsões no conjunto de Validação inédito. A **Rede Neural** obteve o maior ROC-AUC (0.976), cumprindo o objetivo primário de ordenação probabilística do desafio. Sob o prisma de negócios, o **LightGBM** destacou-se com uma Precisão de 0.924, errando pouquíssimos Falsos Positivos, o que minimiza o bloqueio indevido de clientes legítimos.

Tabela I
Desempenho no Conjunto de Validação

Modelo	ROC-AUC	Precisão	Recall	F1
Neural Net	**0.976**	0.833	0.759	0.794
LightGBM	0.975	**0.924**	0.772	**0.841**
CatBoost	0.974	0.826	0.784	0.805
XGBoost	0.974	0.727	0.810	0.766
Random Forest	0.973	0.698	0.822	0.755
Reg. Logística	0.972	0.073	**0.848**	0.135

A avaliação detalhada das Curvas ROC (ver Figura 4 no Anexo) consolidou a capacidade superior de separação probabilística dos algoritmos não-lineares avaliados. Adicionalmente, as Matrizes de Confusão (ver Figura 5 no Anexo) evidenciam o impacto prático e o fracasso da borda linear: apesar do alto *Recall*, a Regressão Logística classificou erradamente milhares de transações lícitas como fraude (Precisão de 7%), tornando-a comercialmente inviável. Modelos não-lineares (árvores e MLP) equilibraram eficientemente essa relação.

B. Avaliação no Kaggle (Teste Público)

As probabilidades geradas pelos melhores modelos foram enviadas ao **leaderboard** oficial (*test.csv*). Surpreendentemente, o desempenho cego foi superior à validação interna, provando a ausência de *overfitting* e a robustez do escalonamento utilizado.

Tabela II
Desempenho (ROC-AUC) no Teste Público (Kaggle)

Modelo	ROC-AUC Público
Neural Net (MLP)	**0.99208**
XGBoost	**0.99103**
LightGBM	0.98615
CatBoost	0.98232

A **Rede Neural** liderou as submissões alcançando um **ROC-AUC Público de 0.99208**, seguida pelo **XGBoost** (0.99103). Estes dois algoritmos foram selecionados para a análise de explicabilidade.

IV. Interpretabilidade

Para garantir que a Rede Neural (campeã) e o XGBoost não atuassem como "caixas-pretas", aplicou-se a biblioteca SHAP (*SHapley Additive exPlanations*). O método baseia-se na teoria dos jogos cooperativos para calcular a contribuição marginal exata de cada variável na predição final de uma transação.

A. Impacto de Negócio e Eficiência Operacional

No conjunto de validação, a Rede Neural bloqueou com sucesso 60 transações fraudulentas reais. O seu grande diferencial competitivo comprovou-se na sua Precisão: o modelo gerou apenas 12 bloqueios indevidos. Em um cenário bancário real, isso garante um atrito mínimo na experiência dos clientes, poupando a instituição de altos custos operacionais com equipes de atendimento (*call center*) e mitigando o risco de *churn* por insatisfação.

B. Análise Global de Atributos e Direcionalidade (SHAP)

O *Summary Plot* ilustrando a topologia de decisão da Rede Neural (ver Figura 6 no Anexo) consolidou o triunfo empírico da variável *Hour_sin* no topo da hierarquia de importância, logo após fatores como *Amount* e *V17*. Isso fornece a prova visual de que a rede neural aprendeu ativamente a utilizar a trigonometria temporal para mapear anomalias da madrugada. O espectro de cores demonstra a direcionalidade: valores extremos (azuis para *V17*, vermelhos para *Hour_sin*) empurram a probabilidade ativamente para a fraude.

O modelo XGBoost (ver Figura 7 no Anexo) corrobora as mesmas variáveis-chave, atestando a integridade do sinal preditivo. A diferença metodológica baseia-se na transição da predição: enquanto o XGBoost agrupa cortes secos ortogonais

em blocos, a Rede Neural cria transições de gradiente muito mais suaves e contínuas, justificando sua ligeira vantagem de abstração matemática para generalizar dados complexos no Kaggle.

V. Limitações e Trabalhos Futuros

Apesar da altíssima acurácia, a abordagem possui limitações de desenvolvimento:

- **Falsos Negativos:** A evasão de 19 fraudes na validação (*Recall* de ∼76%) exige atenção. Trabalhos futuros devem focar na geração sintética de dados (SMOTE/ADASYN) exclusivamente para as fraudes atípicas não detectadas.
- **Custo Computacional:** A explicabilidade por *KernelExplainer* é lenta para inferência bancária em tempo real, exigindo otimizações de arquitetura.
- **Modelagem em Grafos:** Como passo evolutivo, a utilização de Redes Neurais em Grafo (GNNs) permitiria mapear a teia de conexões entre contas, atuando contra as transferências dinâmicas de quadrilhas.

VI. Conclusão

O estudo documentou uma solução altamente eficaz contra o severo desbalanceamento de fraudes financeiras do desafio Ligia. A vitória da arquitetura de Rede Neural sobressai-se aos algoritmos *Gradient Boosting*, não apenas pela capacidade do modelo, mas devido ao rigor na preparação dos dados: a manutenção justificada dos *outliers*, o uso do *RobustScaler* e a modelagem temporal cíclica. Alcançando a marca de 0.992 no teste cego com explicabilidade validada via matrizes de Shapley, o sistema desenvolvido prova-se comercialmente seguro, acurado e perfeitamente aplicável ao ecossistema financeiro.

Anexo: Figuras do Projeto

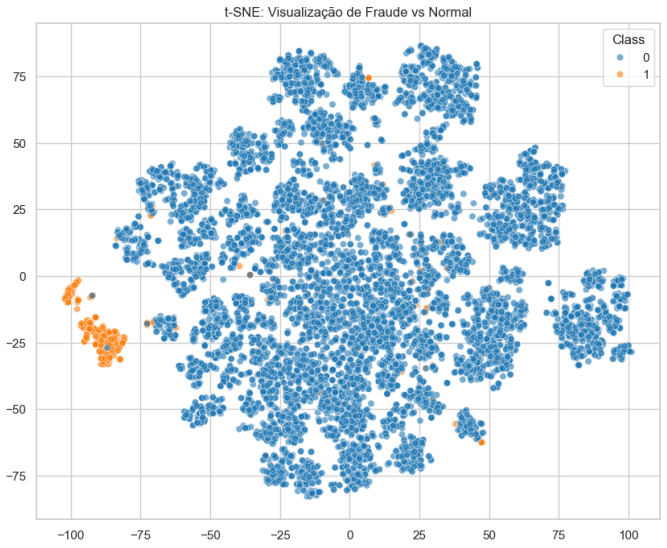

Figura 1. Projeção t-SNE: clusters de fraude formam-se em meio a uma forte sobreposição com transações legítimas.

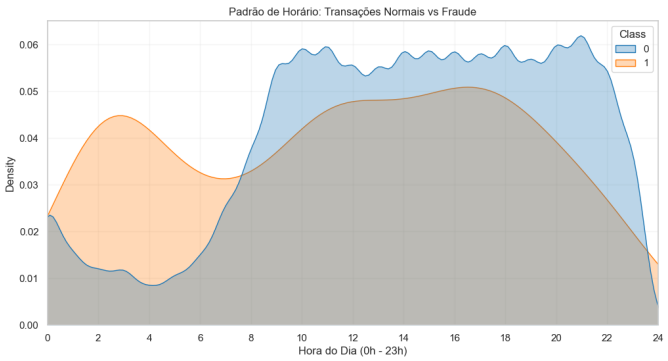

Figura 2. Estimativa de Densidade de Kernel (KDE) ilustrando a proporção relativa de fraudes superior durante a madrugada.

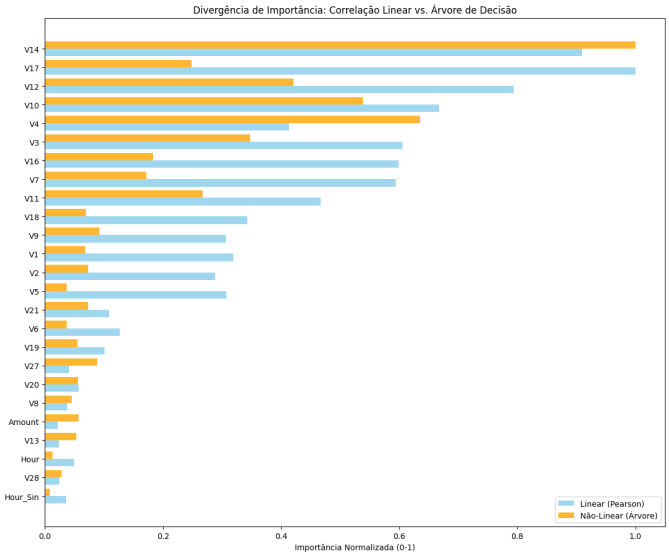

Figura 3. Seleção Híbrida: Correlação Linear (Pearson) vs Importância Não-Linear (Random Forest).

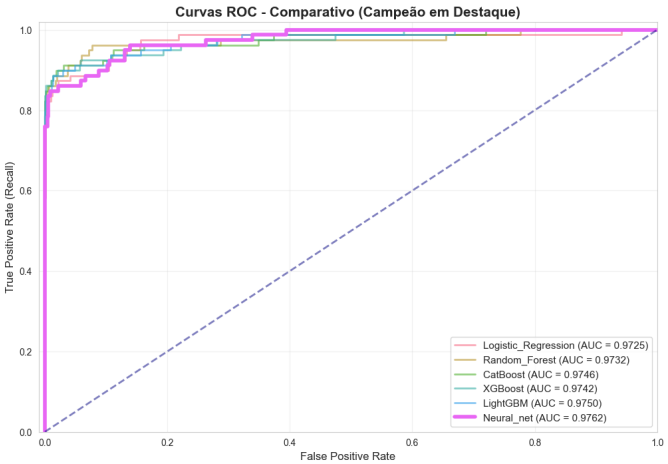

Figura 4. Curvas ROC indicando a convergência probabilística de excelência dos algoritmos avaliados.

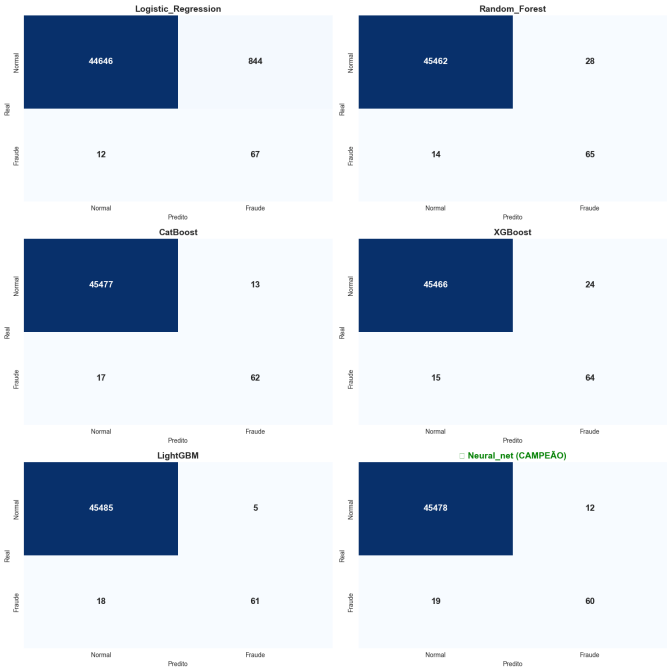

Figura 5. Matrizes de Confusão destacando a taxa massiva de Falsos Positivos no modelo linear de Regressão Logística.

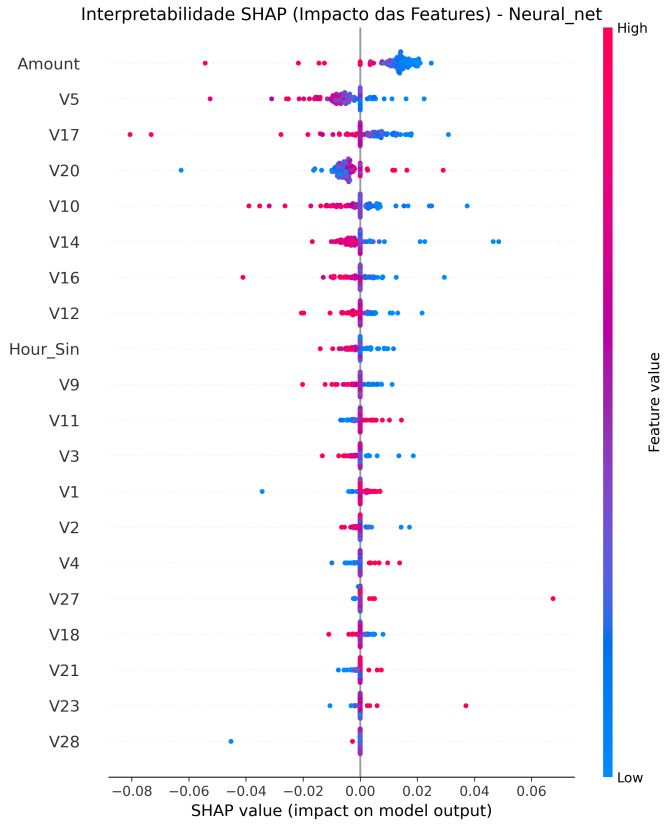

Figura 6. SHAP Summary Plot (Rede Neural). Cores quentes indicam valores altos da característica; o eixo X reflete o impacto em favor da fraude.

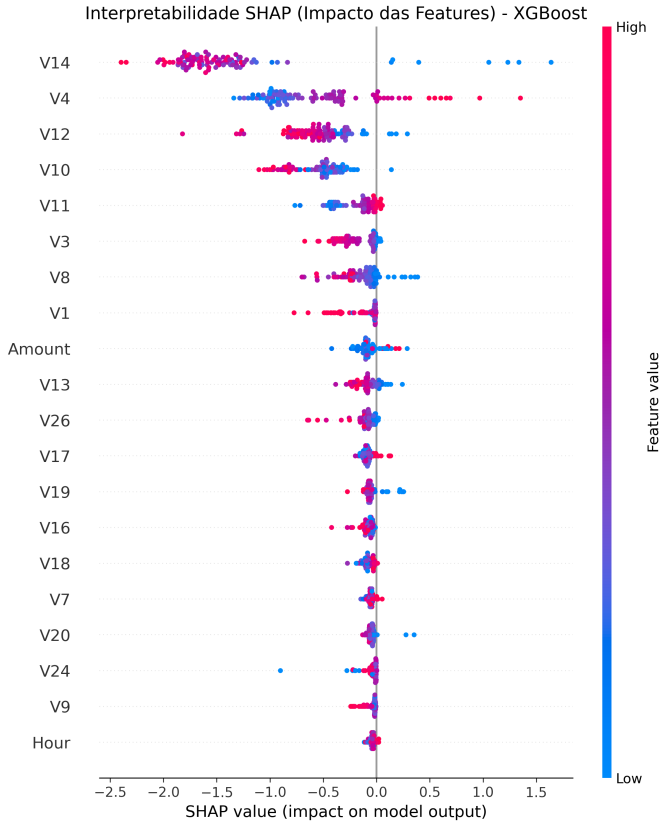

Figura 7. SHAP Summary Plot (XGBoost). Nota-se forte concordância na hierarquia estrutural das características mais importantes.

Referências

[1] V. D. Oliveira, "Métodos de machine learning na detecção de fraude em cartão de crédito: Um estudo comparado," *Revista Científica Acertte*, vol. 5, no. 9, p. e59265, 2025. DOI: 10.63026/acertte.v5i9.265.

[2] Kaggle, "Credit Card Fraud Detection Dataset / Desafio Ligia," Kaggle Inc., 2024. [Online]. Disponível: https://www.kaggle.com/competitions/ligia-machine-learning/overview

[3] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., 2017, pp. 4765–4774.

[4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[5] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3146–3154.

[6] L. Prokhorenkova *et al.*, "CatBoost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems 31*, 2018.

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