

Relatório sobre o Dataset “Financial Transactions Dataset for Fraud Detection”

Descrição do Problema

Contexto

No cenário atual, o volume de transações financeiras digitais cresce exponencialmente devido ao aumento do uso de serviços online, bancos digitais, cartões virtuais e e-commerces. Esse crescimento torna os sistemas mais vulneráveis a fraudes, exigindo soluções automatizadas para detectar atividades suspeitas com rapidez e eficiência.

O Problema

A detecção de transações financeiras fraudulentas é um problema de classificação onde, dado um conjunto de informações sobre cada transação, é necessário identificar se aquela transação é legítima ou fraudulenta.

Detectar fraudes em tempo real é fundamental para impedir que transações fraudulentas sejam concluídas, minimizando prejuízos financeiros e danos à reputação da instituição.

Desafios

- Desequilíbrio entre classes: Transações fraudulentas são raras em comparação com as legítimas, dificultando o aprendizado do modelo.
- Dinâmica e adaptabilidade: Fraudadores mudam suas estratégias constantemente, exigindo modelos que consigam se adaptar rapidamente.
- Velocidade de decisão: Sistemas de detecção precisam ser eficientes para analisar milhões de transações em tempo real, sem impactar a experiência do usuário.
- Complexidade dos dados: Os padrões de fraude são difíceis de identificar manualmente.

Impactos

- Financeiros: Redução direta de perdas financeiras causadas por fraudes.
- Confiança: Manutenção da confiança dos clientes e parceiros.
- Operacionais: Otimização de recursos com sistemas automáticos, reduzindo a necessidade de análise manual.

Sobre o Dataset:

Origem: Kaggle - Financial Transactions Dataset for Fraud Detection

Tamanho: ~1 milhão de registros

Classes: isFraud (0 = legítima, 1 = fraude)

Principais atributos:

step: número da transação no tempo

type: tipo de transação (CASH_OUT, TRANSFER, etc.)

amount: valor da transação

oldbalanceOrg / newbalanceOrig: saldo da conta origem

oldbalanceDest / newbalanceDest: saldo da conta destino

Pré-processamento

- Remoção de colunas irrelevantes:
 - 'transaction_id',
 - 'timestamp',
 - 'sender_account',
 - 'receiver_account',
 - 'ip_address',
 - 'device_id'.
- One-hot encoding para type
- Normalização dos valores monetários
- Divisão em treino/teste: 70% treino / 30% teste
- Balanceamento da classe

Justificativa do Uso de Redes Neurais

As redes neurais são boas para identificar padrões complexos nos dados, mesmo quando esses padrões não são tão óbvios. Por isso, elas funcionam muito bem em tarefas como a detecção de fraudes, onde é preciso analisar muitos dados e encontrar sinais sutis de comportamento suspeito.

Descrição da arquitetura adotada

- Função de ativação ReLU
 - Ajuda a lidar com não linearidades.
 - Rápida convergência.
- Função de ativação Sigmoid na saída:
 - Retorna um valor entre 0 e 1, interpretado como probabilidade de fraude.
 - Permite aplicar um limiar (threshold) para classificar como fraude ou não.
- Função de perda: binary_crossentropy
 - Ideal para problemas de classificação binária.
- Otimizador: Adam
 - Adaptativo e eficiente para redes profundas.
- Épocas: 10

- Suficientes para aprendizado sem overfitting (em testes preliminares).
- Batch size: 32
 - Pequeno o suficiente para capturar variações e evitar overfitting.
 - Permite atualizações frequentes do modelo durante o treinamento
- Validação: 20% dos dados de treino separados automaticamente (validation_split=0.2)

Conclusão

O desempenho do nosso modelo foi ok — não chegamos a uma acurácia próxima do ideal, mas o resultado foi bom o suficiente. No final, conseguimos alcançar cerca de 90% de acurácia, o que consideramos satisfatório.

Algumas limitações atrapalharam a gente de chegar mais perto da solução ideal. A principal delas foi que os padrões de fraude mudavam com facilidade e frequência, o que dificultava bastante pro modelo conseguir identificá-las com precisão.

Aprimorar a engenharia de atributos:

- Criar variáveis derivadas, como:
- Diferença de saldo antes e depois;
- Número de transações por hora;
- Tempo desde a última transação do usuário;