UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Aprendizado de Máquina

Mini Trabalho 7

Apresentação e documentação da solução

Grupo - 1

Ana Clara Barbosa Borges
André Emanuel Bispo da Silva
Artur Handow Krauspenhar
Gabriel Moura dos Santos
João Artur Leles Ferreira Pinheiro
João Pedro Anacleto Ferreira Machado

Brasília, DF
2025

Introdução

Umas das áreas onde machine learning é mais aplicada é na identificação de fraudes e transações ilícitas em finanças. Algoritmos de machine learning são especialmente aptos para este tipo de tarefa pois agentes ilícitos estão constantemente e rapidamente inovando as suas técnicas e métodos, algo que, na escala em que se passam transações cotidianas, não pode ser acompanhado manualmente. Dessa forma, estes algoritmos são capazes de filtrar essas grandes massas de dados e efetuar a “deteção de outliers” ou transações suspeitas que são “anormais” ou diferentes das outras.

Com isso em mente, o nosso trabalho visa treinar um modelo capaz de identificar transações fraudulentas, ou fazer “deteção de *outliers*” em conjuntos de dados de transações bancárias. Para este fim nos baseamos em um *dataset* público disponível na plataforma *Kaggle* de transações financeiras reais, foi colocada ênfase em dados reais para obter resultados reais. E utilizamos isto como ponto de partida para o trabalho.

Metodologia

Foram feitos testes simples preliminares a fim de evitar gastos grandes de tempo vários modelos diferentes, a fim de se filtrar candidatos para otimização. a base para a análise desses resultados foi o recall, pois o foco é na identificação de outliers, e como as classes são extremamente desbalanceadas no dataset o modelo estaria deveras enviesado.

A metodologia adotada para o desenvolvimento da solução de detecção de fraude seguiu um pipeline estruturado, compreendendo a preparação e pré-processamento dos dados, a seleção de modelos de aprendizado de máquina, a otimização e ajuste fino dos algoritmos mais promissores, e a definição de métricas de avaliação apropriadas para o problema.

Escolha do Dataset

O estudo foi conduzido utilizando o conjunto de dados "Credit Card Fraud Detection", obtido da plataforma Kaggle. Ele consiste em transações de cartão de crédito realizadas por titulares europeus durante um período de dois dias em setembro de 2013.

O conjunto de dados compreende um total de 284.807 transações, das quais apenas 492 foram classificadas como fraudulentas.

As features do dataset incluem:

- 'Time': Uma variável numérica (float64) que representa o tempo em segundos decorrido entre cada transação e a primeira transação no conjunto de dados.
- 'Amount': Uma variável numérica (float64) que representa o valor da transação.
- V1-V28: Vinte e oito variáveis numéricas (float64) que são o resultado de uma transformação de Análise de Componentes Principais (PCA) aplicada às features originais. Esta anonimização foi realizada para proteger a confidencialidade dos dados dos titulares dos cartões.
- 'Class': A variável alvo (int64), binária, que indica se uma transação é fraudulenta (1) ou não fraudulenta (0).

Análise Exploratória dos Dados (EDA)

Uma análise exploratória detalhada dos dados (EDA) foi realizada para compreender a estrutura, as características e os padrões presentes no conjunto de dados.

- **Dimensões e Tipos de Dados**: Confirmou-se que o dataset possui 284.807 transações e 31 colunas. Trinta colunas são do tipo float64 e uma ('Class') é do tipo int64.
- **Valores Ausentes**: Percebemos a ausência de valores nulos (missing values) em todas as colunas do dataset.
- **Linhas Duplicadas**: Foram identificadas 1.854 linhas duplicadas no conjunto de dados. Optamos por manter essas duplicatas porque transações aparentemente idênticas podem, em alguns cenários, representar um padrão de atividade fraudulenta ou simplesmente transações legítimas que compartilham características. A remoção indiscriminada poderia levar à perda de informações potencialmente relevantes.

Preparação e Pré-processamento dos Dados

A fase de preparação dos dados foi crucial para garantir a qualidade e a consistência das informações fornecidas aos modelos.

- **Limpeza de Dados**: Como identificado na Análise Exploratória de Dados, o conjunto de dados não apresentava valores ausentes (missing values), simplificando esta etapa. A decisão de manter as 1.854 linhas duplicadas e os outliers observados foi reiterada, baseada na premissa de que tais instâncias poderiam conter sinais relevantes de atividade fraudulenta e sua remoção poderia prejudicar a capacidade do modelo de detectar esses padrões.
- **Normalização/Padronização**: As features V1-V28, sendo resultado de uma transformação PCA, já se encontram padronizadas (média 0 e desvio padrão 1). No entanto, as features 'Time' e 'Amount' possuem escalas muito distintas.
- **Divisão dos Dados em Conjuntos de Treinamento e Teste**: Para garantir

uma avaliação justa e consistente do desempenho dos modelos, o conjunto de dados foi dividido em duas partes: 80% para treinamento e 20% para teste. Essa divisão foi realizada utilizando a função train_test_split da biblioteca Scikit-learn, e os conjuntos resultantes foram salvos e reutilizados em todos os experimentos de modelagem. Isso assegura que todos os modelos fossem treinados e avaliados exatamente sobre os mesmos dados, permitindo comparações diretas de desempenho.

A manutenção de outliers e a subsequente escolha da técnica de validação cruzada StratifiedKFold representam uma combinação metodológica particularmente robusta para este problema. O StratifiedKFold assegura que a proporção de classes, incluindo a rara classe de fraude (que pode ser vista como um tipo de outlier comportamental), seja preservada. Em um dataset tão desbalanceado, o uso de uma validação cruzada padrão (não estratificada) poderia resultar em alguns folds contendo pouquíssimas ou nenhuma instância da classe minoritária (fraude). Isso comprometeria seriamente o treinamento e a avaliação do modelo nesses folds específicos. Ao garantir que cada fold seja uma representação proporcional do dataset completo, o StratifiedKFold torna o processo de validação cruzada mais confiável e os resultados de desempenho mais estáveis e representativos da capacidade do modelo em generalizar para dados não vistos, especialmente no que tange à detecção da classe de interesse (fraude).

Seleção de Modelos de Aprendizado de Máquina

Uma variedade de algoritmos de classificação foi testada inicialmente para identificar os candidatos mais promissores para a detecção de fraude neste conjunto de dados. A seleção foi guiada pelo desempenho em métricas críticas para problemas de classificação desbalanceada

- **Algoritmos Testados**: Os modelos avaliados na fase inicial incluíram:
  - K-Nearest Neighbors (KNN)
  - XGBoost
  - Random Forest
  - Support Vector Machine (SVM)
  - Categorical Boosting (CatBoost)
  - AdaBoost.
- **Critérios de Seleção**: Dado o severo desbalanceamento de classes e a importância crítica de identificar corretamente as transações fraudulentas (minimizando Falsos Negativos), o foco principal na avaliação preliminar foi em métricas como Recall (Sensibilidade) e F1-Score para a classe minoritária (fraude). A acurácia geral, embora calculada, foi considerada com cautela, pois pode ser enganosamente alta em datasets desbalanceados devido à correta classificação da classe majoritária.
- **Justificativa para a Escolha de Random Forest e CatBoost**:
  - **Random Forest**: Nos testes preliminares, utilizando o parâmetro

class_weight='balanced' para mitigar o efeito do desbalanceamento, o Random Forest demonstrou um desempenho notável. Ele alcançou um Recall de 0.81 (81%) e um F1-score de 0.87 para a classe de fraude (Classe 1). A matriz de confusão mostrou 79 Verdadeiros Positivos (fraudes corretamente detectadas) e apenas 19 Falsos Negativos (fraudes não detectadas).

- **Categorical Boosting (CatBoost)**: Este modelo, mesmo em uma fase de otimização preliminar com RandomizedSearch, também se destacou. A análise de sua matriz de confusão (79 Verdadeiros Positivos, 19 Falsos Negativos, 1 Falso Positivo) indicou um Recall de aproximadamente 0.81 (79/(79+19)) e um F1-score de aproximadamente 0.89 para a classe de fraude.
- **Desempenho de Outros Modelos**: Em contraste, modelos como KNN apresentaram resultados insatisfatórios, com um Recall de apenas 15% e F1-score de 27% para fraudes. O SVM com kernel linear, apesar de uma alta acurácia geral, teve um Recall de apenas 30% para fraudes. O XGBoost, sem otimização específica para o desbalanceamento ou ajuste fino de hiperparâmetros naquela fase, também mostrou um Recall muito baixo para fraudes (3.12%). O AdaBoost teve um desempenho intermediário, com F1-score de 0.75 e Recall de 0.70 para fraudes.

Otimização e Ajuste Fino dos Modelos Selecionados

Com base nos resultados promissores da fase de seleção, os modelos Random Forest e CatBoost foram submetidos a um processo de otimização e ajuste fino de hiperparâmetros, visando maximizar seu desempenho preditivo e robustez.

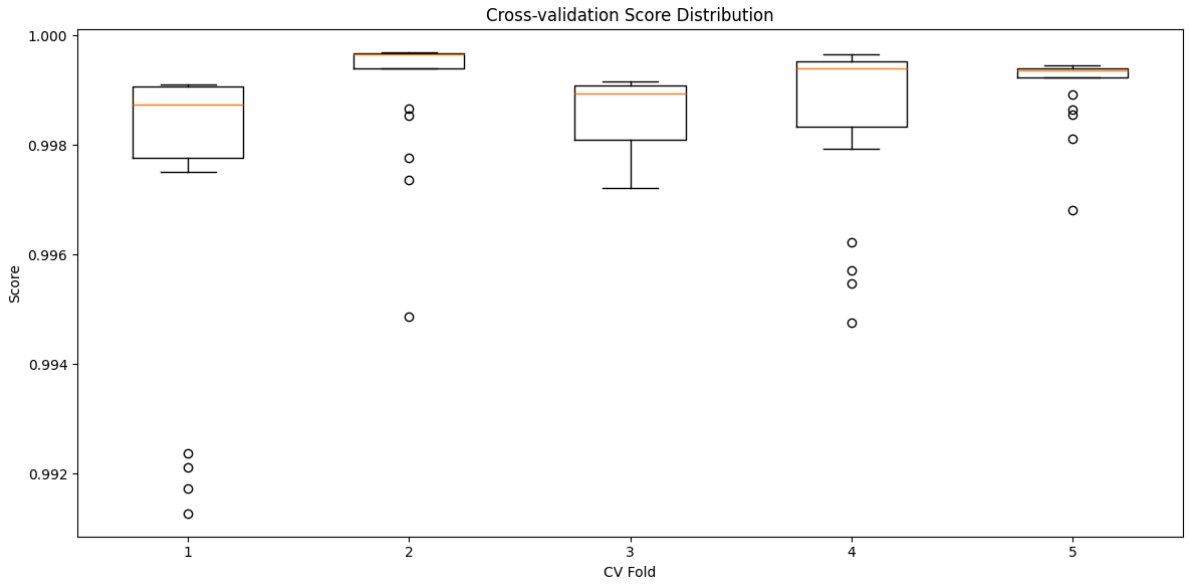
- **Técnica de Otimização**: A principal ferramenta utilizada para a exploração do espaço de hiperparâmetros foi o RandomizedSearchCV da biblioteca Scikit-learn. Esta técnica realiza uma busca por combinações de hiperparâmetros amostradas aleatoriamente de distribuições especificadas, o que pode ser mais eficiente do que uma busca exaustiva em grade (GridSearchCV) quando o espaço de busca é grande ou os recursos computacionais são limitados. A literatura de pesquisa em ML para detecção de fraude também aponta o RandomizedSearchCV como uma das técnicas de otimização de hiperparâmetros utilizados, embora o GridSearch seja mais comum. A utilização do RandomizedSearchCV demonstra uma escolha pragmática na busca por configurações de hiperparâmetros eficazes, respeitando as limitações de recursos computacionais, especialmente a escassez de hardware de alta performance.
- **Validação Cruzada**: Para avaliar o desempenho de cada combinação de hiperparâmetros e evitar o sobreajuste (overfitting) ao conjunto de treinamento, foi empregada a validação cruzada StratifiedKFold. Esta técnica divide o conjunto de treinamento em k subconjuntos (folds), garantindo que cada fold

mantenha a mesma proporção de amostras de cada classe que o conjunto de dados original. No projeto, utilizou-se k=5 folds. A cada iteração da validação cruzada, k-1 folds são usados para treinar o modelo e o fold restante é usado para avaliação. Este processo é repetido k vezes, e as métricas de desempenho são então agregadas (e.g., calculando a média).

Otimização do Random Forest:

Após o uso do RandomizedSearchCV e do StratifiedKFold no Random Forest, conseguimos fazer o ajuste fino dos hiperparâmetros e melhorar o desempenho do modelo.

- **Técnica de Otimização:** Após as execuções aleatórias, a melhor combinação de hiperparâmetros encontrada dentro dos que selecionamos foi: **class_weight = 'balanced'**, **criterion = 'entropy'**, **max_depth = 17**, **min_samples_leaf = 4**, **min_samples_split = 19**, **n_estimators = 153.**
- **Resultados Observados:** observamos um F1-score médio de aproximadamente 0.99. Segue os resultados da validação cruzada:



Fonte: André Silva, 2025.

Otimização do CatBoost:

Após uma análise dos resultados obtidos em uma primeira iteração decidimos que o Catboost seria um bom caminho para seguirmos com a otimização aonde foi realizada por meio de uma abordagem estruturada utilizando RandomizedSearchCV para ajuste fino dos hiperparâmetros e validação cruzada estratificada (StratifiedKFold) para garantir a robustez dos resultados.

- **Técnica de Otimização**: Foram realizadas execuções aleatórias conduzidas

pelo RandomizedSearch, explorando diferentes regiões do espaço de hiperparâmetros, sendo realizadas o teste com diversas combinações dos hiperparâmetros, ao qual foram escolhidos os seguintes valores : **Bagging Temperature (0.2786**), **Border Count (225**), **Depth (4)**, **Iterations (412)**, **L2 Regularization (2)**, **Learning Rate (0.1693)** e **Random Strength (0.4478)**

- **Resultados observados:** observamos um excelente desempenho com um **F1-score macro de 0.94**, sendo **1.0** para a classe não fraudulenta e **0.88** para a classe fraudulenta. Em comparação com resultados anteriores observamos uma redução leve no número de falsos negativos (FN), evidenciando que os ajustes contribuíram para uma detecção mais eficiente, mantendo apenas 1 falso positivo.Dessa forma o CatBoost, se mostrou extremamente eficaz e equilibrado no trade-off entre precisão e recall.

Métricas de Avaliação

A escolha das métricas de avaliação é um aspecto crítico em qualquer projeto de aprendizado de máquina, especialmente em cenários com classes desbalanceadas, como a detecção de fraude. Métricas inadequadas podem levar a conclusões enganosas sobre o desempenho do modelo.

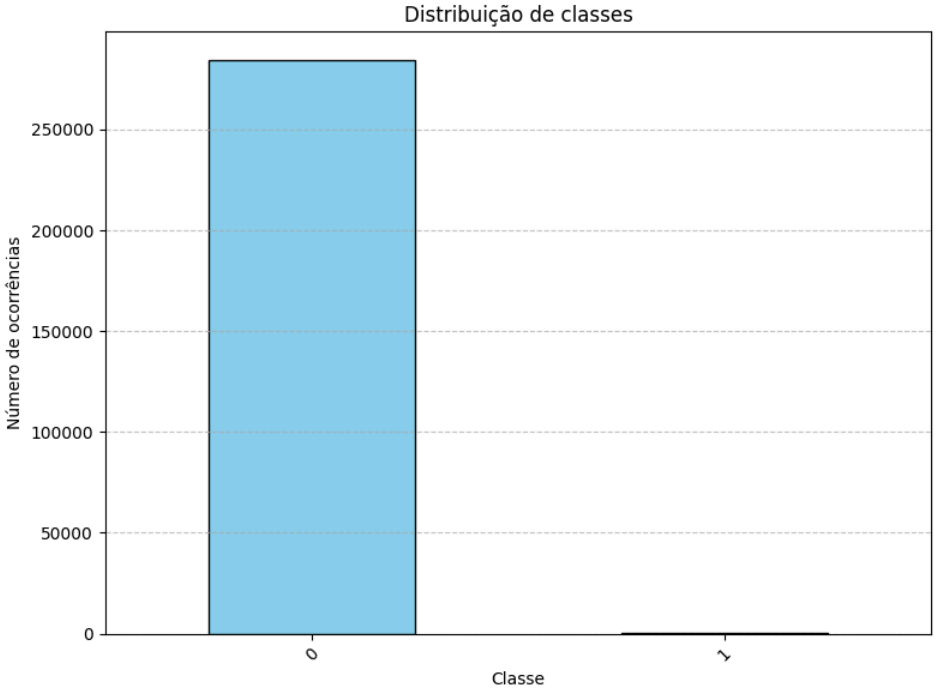
- **Métricas Primárias para a Classe Minoritária (Fraude)**: Dada a importância de detectar o maior número possível de fraudes reais (minimizar Falsos Negativos - FN) e, ao mesmo tempo, garantir que as transações sinalizadas como fraudulentas sejam de fato fraudes (minimizar Falsos Positivos - FP), as seguintes métricas foram priorizadas para a classe de fraude:
  - **Recall (Sensibilidade ou Taxa de Verdadeiros Positivos)**: Mede a proporção de transações fraudulentas reais que foram corretamente identificadas pelo modelo (Recall=VP/(VP+FN)). Um alto recall é crucial, pois indica que poucas fraudes estão passando despercebidas.
  - **Precisão (Valor Preditivo Positivo)**: Mede a proporção de transações classificadas como fraudulentas que são realmente fraudes (Precisão=VP/(VP+FP)). Uma alta precisão é importante para evitar o bloqueio desnecessário de transações legítimas e a consequente inconveniência para os clientes.
  - **F1-Score**: É a média harmônica da Precisão e do Recall (F1=2*(Precisão*Recall)/(Precisão+Recall)). Fornece um balanço entre essas duas métricas, sendo particularmente útil quando há um desequilíbrio entre as classes ou quando os custos de FP e FN são diferentes.
- **Outras Métricas Consideradas**:
  - **Acurácia Geral**: Proporção de todas as transações (fraudulentas e não fraudulentas) que foram classificadas corretamente (Acurácia=(VP+VN)/(VP+VN+FP+FN)). Embora comumente usada, a acurácia pode ser enganosa em datasets altamente desbalanceados. Um modelo que classifica todas as transações como não fraudulentas ainda

alcançaria uma alta acurácia neste cenário, mas seria inútil para detectar fraudes.

- **Matriz de Confusão**: Uma tabela que resume o desempenho da classificação, mostrando o número de Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN).[1] É fundamental para uma análise detalhada dos tipos de erros cometidos pelo modelo.

A forte ênfase em Recall e F1-score para a classe de fraude demonstra uma compreensão crítica da natureza do problema de detecção de fraude. No mundo real, o custo associado a um Falso Negativo (uma fraude não detectada que resulta em perda financeira direta e potencial dano à reputação) é tipicamente muito maior do que o custo de um Falso Positivo (uma transação legítima erroneamente bloqueada, causando inconveniência ao cliente, mas geralmente com menor impacto financeiro direto para a instituição). Portanto, as métricas que refletem a capacidade do modelo de identificar corretamente as fraudes (Recall) e de fazê-lo com um bom equilíbrio em relação aos alarmes falsos (F1-score) são mais indicativas da utilidade prática de um modelo de detecção de fraude do que a acurácia geral.

Figura 1 - Distribuição de classes no dataset



Fonte: André Silva, 2025.

Os testes simples consistem em uma escolha aleatória de hiperparâmetros, divisão do dataset em aproximadamente 25% de linhas para testes, treino do modelo com o restante do dataset e geração de relatório. Foram testados vários modelos classificatórios, excluindo alguns julgados impróprios por sua simplicidade, como o modelo de regressão logarítmica. Os resultados foram agregados e comunicados pelo nosso canal de comunicação no *WhatsApp*, e com base na análise dos gráficos e relatórios fizemos a escolha final do modelo.

Após testes com diferentes modelos foi possível escolher os modelos com maior potencial, o *Random Forest* e *CatBoost*, dentre os modelos treinados estes foram os que tiveram os melhores resultados preliminares.

Conclusão

A equipe do projeto se dedicou para desenvolver uma solução de aprendizado de máquina para a detecção de fraude em transações de cartão de crédito, um problema de grande relevância. A nossa metodologia envolveu a escolha de um dataset, análise exploratória de dados, pré-processamento, seleção de múltiplos algoritmos e otimização fina de hiperparâmetros. Após as avaliações, os modelos Random Forest e Categorical Boosting (CatBoost) foram considerados os mais adequados para atingir o nosso objetivo.

Durante o desenvolvimento do projeto, encontramos como fatores limitantes o Alto Desbalanceamento de Classes, uma Interpretação Limitada devido à anonimização de dados e uma limitação de Recursos Computacionais.

Por fim, os trabalhos futuros envolvem o estudo da melhor forma de implementação e a preparação de uma forma de monitoramento para acompanhar o desempenho do modelo após lançado.