Projeto em Fundamentos de Big Data

Nome dos(as) Alunos(as): Arthur Lins, Lucas Fernandes e Victor Aroucha
Título do Projeto/Demanda: Detecção de transações fraudulentas com cartão de crédito em tempo real

Introdução:

O conjunto de dados que a gente está usando traz informações de transações com cartões de crédito feitas por clientes europeus em setembro de 2013. A ideia principal é conseguir identificar automaticamente transações suspeitas de fraude. Esse dataset é muito usado em estudos sobre o tema porque ele é real, mas ao mesmo tempo mantém a privacidade dos clientes.

As variáveis originais foram transformadas usando uma técnica chamada Análise de Componentes Principais (PCA). Essa técnica mistura várias colunas que eram parecidas entre si e cria novas variáveis chamadas componentes principais. No nosso caso, essas variáveis são as colunas V1 até V28. Elas não representam coisas específicas (tipo país, tipo de loja ou categoria do produto), mas guardam as relações matemáticas que ajudam a diferenciar o que é uma transação normal e o que pode ser fraude.

Além dessas, o dataset tem três colunas que dá pra entender de forma direta. A coluna Time mostra o tempo (em segundos) desde a primeira transação registrada, o que ajuda a ver a sequência e possíveis padrões de tempo nas fraudes. A coluna Amount indica o valor da transação em euros, que serve pra ver se as fraudes tendem a acontecer mais em valores baixos ou altos. E por último, a coluna Class, que é o alvo do nosso projeto: ela vale 0 pra transações normais e 1 pra transações fraudulentas.

Um ponto importante é que esse conjunto é bem desbalanceado, tem mais de 284 mil transações normais e só 492 fraudes, o que dá menos de 0,2% do total. Esse é o principal desafio do problema e, mais pra frente, a gente vai precisar aplicar técnicas específicas pra lidar com isso na hora da modelagem e avaliação dos resultados.

Mesmo sendo só uma amostra de dois dias de transações, esse conjunto de dados representa bem um problema típico de Big Data. Em um sistema real de cartões de crédito, milhares de transações acontecem a cada segundo, no mundo todo, e precisam ser analisadas em tempo real pra evitar fraudes.

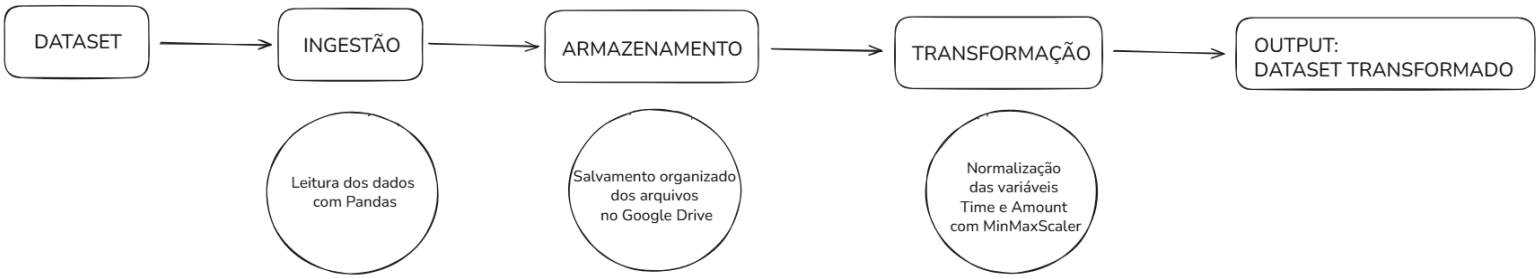
O desafio aqui não é só o volume de informações, mas também a velocidade com que os dados chegam e a complexidade das variáveis que precisam ser processadas e comparadas pra decidir se algo é suspeito ou não. Além disso, é um caso em que a quantidade de dados

legítimos é enorme comparada às poucas fraudes, o que exige técnicas de processamento, armazenamento e modelagem típicas de ambientes de Big Data.

Por isso, mesmo esse dataset sendo uma versão reduzida, ele permite simular as mesmas etapas e dificuldades que aparecem em projetos reais de Big Data e aprendizado de máquina aplicados à detecção de fraudes.

AV1:

Diagrama do Pipeline de Dados:



Tecnologias Utilizadas e Possíveis Refinamentos:

Ingestão: Python + Pandas
Armazenamento: Google Drive
Transformação: Scikit-learn (MinMaxScaler)
Visualização: Matplotlib / Seaborn

Tecnologias que poderiam ser usadas para refinamento (pagas ou avançadas):

Ingestão em tempo real: Apache Kafka (Permitem ingestão de dados em tempo real em grande escala)

Armazenamento distibuído: AWS ou Databricks (Armazenamento em nuvem de alta disponibilidade e integração com pipelines automatizados)
Processamento de grandes volumes: Apache Spark (Databricks) (Escalabilidade e processamento paralelo de grandes datasets)
Visualização: Power BI (Dashboards interativos e monitoramento em tempo real)

Arquitetura parcial implementada (ambiente simulado, processamento em batch):

Visão geral: Implementamos um ETL simples em batch no Google Colab, usando o Google Drive como armazenamento. O fluxo foi: ingestão (leitura completa do CSV), transformação (normalização), armazenamento do resultado.

Ingestão (batch)

- Leitura direta do arquivo creditcard.csv a partir do Drive: /content/drive/MyDrive/BigData/creditcard.csv
- Checagens rápidas: df.shape, df.info(), verificação de nulos (df.isnull().sum() = 0), distribuição da coluna alvo (df['Class'].value_counts()) e inspeção das 5 primeiras fraudes (df[df['Class']==1].head())

Armazenamento

- Organização dos artefatos no Drive sob /content/drive/MyDrive/BigData

Transformação

- As colunas **Time** e **Amount** foram **normalizadas** com **MinMaxScaler** (intervalo [0, 1]).
- As demais variáveis (V1…V28) já estão padronizadas (derivadas de PCA) e **não precisaram de ajuste**
- Visualizações: histogramas de Amount e Time **antes e depois** da normalização para justificar a escolha (diferença de escala e cauda longa em Amount)

Decisões de design

- **Batch** em vez de streaming, por ser a entrega da **AV1** (foco em ingestão, transformação e armazenamento)

- **MinMaxScaler** escolhido para alinhar a escala de Time e Amount às demais features (PCA), sem alterar a forma da distribuição

Equipe responsável e divisão de tarefas:

INTEGRANTE	RESPONSABILIDADES PRINCIPAIS
Arthur Lins	Ingestão (leitura completa do CSV), transformação (normalização), visualizações e documentação no Colab.
Lucas Fernandes	Organização do Github, estrutura do repositório e revisão do Documento de Arquitetura.
Victor Aroucha	Consolidação das justificativas técnicas (PCA, normalização), diagrama do pipeline e revisão final do material da AV1.