LISTA 9 - IA

Andriel Mark da Silva Pinto - 1528633.

Questão 1 – Etapas de pré-processamento

Link do repositório:
https://github.com/andrielmark/PUCMINAS/tree/main/QUARTO%20PERÍODO/IA/LISTA9

O objetivo desta questão é realizar todas as etapas de pré-processamento na base *Credit Card Fraud Detection*, disponível no Kaggle, para preparar os dados antes da aplicação dos algoritmos de agrupamento.

Inicialmente, a base foi carregada e analisada por meio dos comandos tradicionais **(head(), info(), describe()**). A partir disso, observei que o conjunto possui **31 atributos**, todos numéricos, sem valores ausentes, além de apresentar forte desbalanceamento na variável *Class*, que marca as transações fraudulentas (aproximadamente 0,17% dos registros).

Em seguida, verifiquei se existiam valores nulos utilizando **df.isnull().sum().** Como nenhum atributo continha dados ausentes, não foi necessária a aplicação de métodos de imputação. Também busquei dados redundantes utilizando **df.duplicated()**, sendo identificadas **1081 linhas duplicadas**, que foram removidas para evitar distorções estatísticas e facilitar o processo de clusterização.

Para análise de outliers, utilizei duas abordagens complementares: **Z-Score** e **IQR**. O Z-Score permitiu observar valores muito distantes da média de cada atributo, enquanto o método do IQR foi mais adequado para atributos com distribuição não uniforme, como *Amount* e *Time*. Um ponto só era removido quando aparecia como outlier nos dois métodos, garantindo que apenas casos realmente extremos fossem eliminados.

A padronização foi realizada usando **StandardScaler**, já que K-Means e DBSCAN dependem de medidas de distância e são fortemente afetados por diferenças de escala. Após o escalonamento, realizei também uma análise de correlação e multicolinearidade. Alguns componentes apresentaram indícios de redundância entre si, então decidi aplicar **PCA (10 componentes)** para reduzir dimensionalidade, facilitar o agrupamento e diminuir ruído.

A etapa de codificação não foi necessária, pois todos os atributos já eram numéricos. Da mesma forma, o balanceamento da classe não foi realizado, já que a variável *Class* não

deve ser utilizada como alvo em uma tarefa de clusterização; ela foi usada apenas posteriormente como referência para avaliação dos grupos formados.

Por fim, fiz a divisão **treino–teste estratificada**, garantindo diversidade nas amostras, mesmo que a classe não seja utilizada para treinar nenhum modelo. Essa separação permitiu avaliar melhor a estabilidade dos agrupamentos.

Questão 2 – Algoritmos de agrupamento

Nesta etapa, utilizei os algoritmos **K-Means**, **DBSCAN** e **Self-Organizing Maps (SOM)** para verificar se era possível identificar dois grupos naturais na base, lembrando que a coluna *Class* foi removida antes de aplicar os métodos.

Antes do pré-processamento, os resultados foram fracos para todos os modelos. O K-Means apresentou clusters muito misturados, o DBSCAN classificou a maior parte dos dados como ruído, e o SOM apresentou um mapa pouco informativo, sem regiões bem definidas.

Após todo o pré-processamento, os algoritmos apresentaram melhorias significativas:

- **K-Means** conseguiu formar grupos mais bem delimitados, embora ainda houvesse bastante mistura entre transações normais e fraudulentas. O *Silhouette Score* ficou próximo de **0.14**, um valor baixo, mas superior aos demais métodos e adequado considerando a complexidade da base.

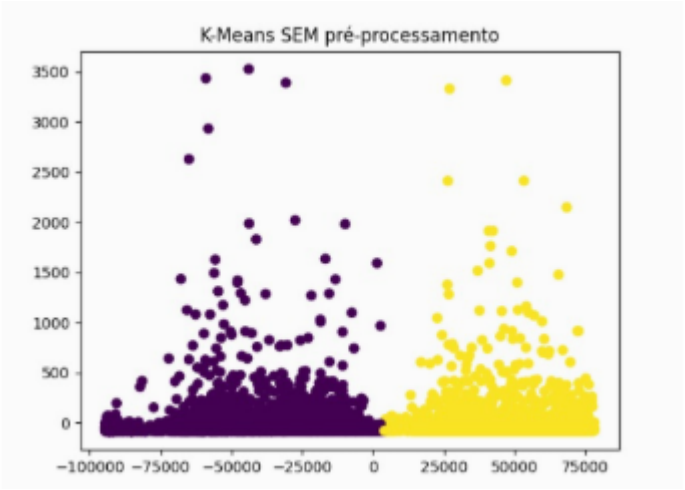
- **DBSCAN** formou pequenos agrupamentos válidos, mas continuou muito sensível aos parâmetros de densidade, resultando em um número elevado de pontos sendo marcados como ruído (**-1**). Isso dificulta a avaliação do modelo pelas métricas tradicionais.

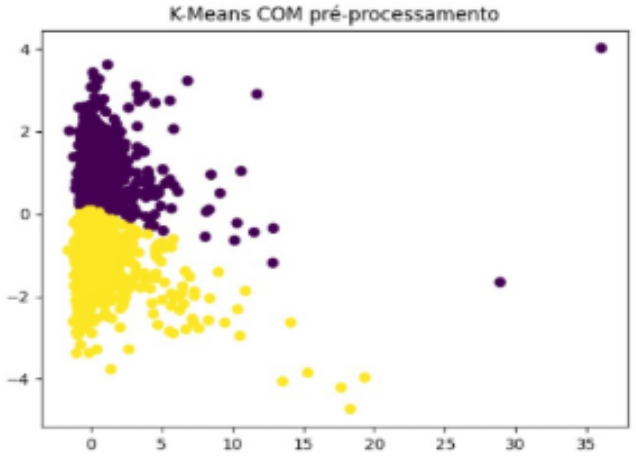
- **SOM** mostrou regiões específicas de maior concentração de padrões semelhantes, o que pode indicar áreas onde transações incomuns se acumulam. Apesar disso, o método não separou claramente dois grupos completos, mas apresentou comportamento interessante ao representar padrões não lineares da base.

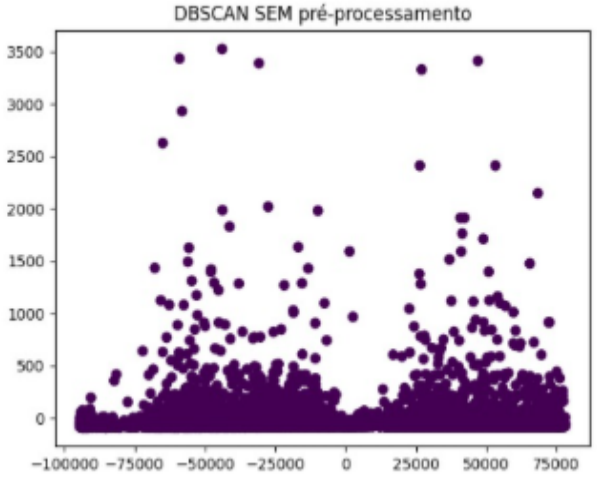
No geral, mesmo com o pré-processamento extenso, nenhum dos três algoritmos conseguiu detectar dois grupos naturais de forma clara. Os resultados reforçam que este problema é muito mais adequado a **abordagens supervisionadas**, já que a separação entre transações legítimas e fraudulentas não ocorre de maneira naturalmente agrupável nos espaços de atributos.

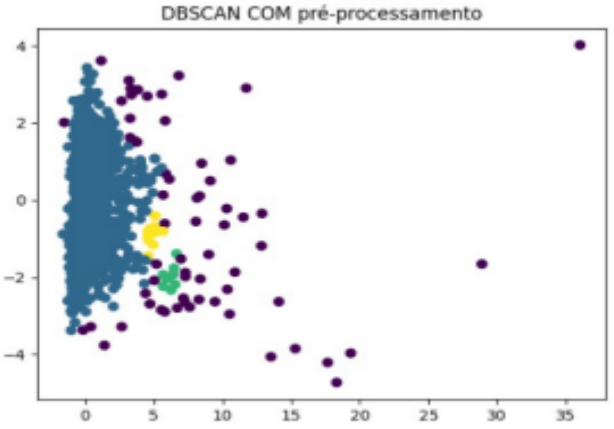
O algoritmo com melhor desempenho foi o **K-Means**, por conseguir representar uma separação mínima entre os comportamentos, embora ainda limitada. DBSCAN e SOM captaram aspectos particulares da base, mas não realizaram uma divisão consistente em dois grupos.

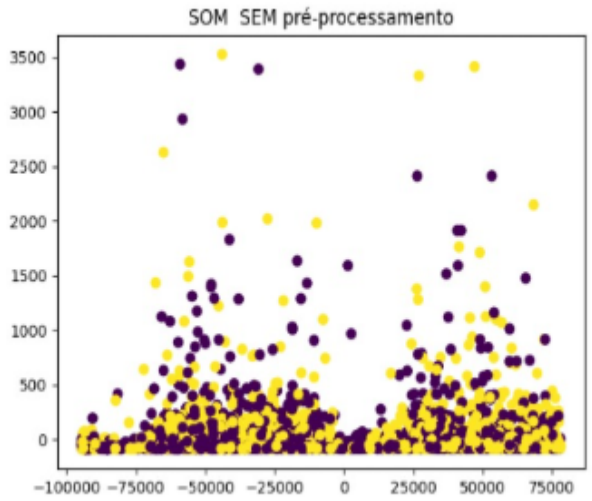
RESULTADOS:

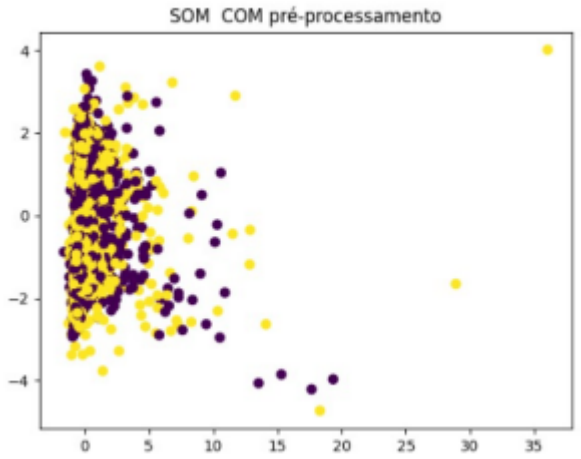

K-Means SEM pré-processamento
3500
3000
2500
2000
1500
1000
500
0
−100000 −75000 −50000 −25000 0 25000 50000 75000


K-Means COM pré-processamento
4
2
0
−2
−4
0 5 10 15 20 25 30 35


DBSCAN SEM pré-processamento
3500
3000
2500
2000
1500
1000
500
0
−100000 −75000 −50000 −25000 0 25000 50000 75000


DBSCAN COM pré-processamento
4
2
0
−2
−4
0 5 10 15 20 25 30 35


SOM SEM pré-processamento
3500
3000
2500
2000
1500
1000
500
0
−100000 −75000 −50000 −25000 0 25000 50000 75000


SOM COM pré-processamento
4
2
0
−2
−4
0 5 10 15 20 25 30 35