

CURSO SISTEMAS DE INFORMAÇÃO

MINERAÇÃO DE DADOS

DETECÇÃO DE ANOMALIAS OU REGRAS DE ASSOCIAÇÃO COM PYCARET

Esta atividade tem por objetivo aplicar os conhecimentos de Machine Learning não supervisionado utilizando a biblioteca PyCaret e com a biblioteca MLxtend para resolver uma das seguintes tarefas, a depender do dataset designado a cada aluno:

- Detecção de Anomalias (*Anomaly Detection*)
- Regras de Associação (*Association Rules*)

Cada aluno deverá identificar, explorar e modelar seu dataset, explicando claramente os atributos utilizados e justificando todas as decisões tomadas ao longo do processo.

Passos gerais (para todos os alunos):

Independente da tarefa, todos devem:

- Importar o dataset e realizar a análise exploratória.
- Explicar todos os atributos do conjunto de dados.
- Descrever a origem do dataset.
- Comentar cada etapa do código.
- Apresentar uma interpretação dos resultados obtidos e justificando a viabilidade de aplicação em um cenário real do modelo treinado

CASO A SUA TAREFA SEJA DETECÇÃO DE ANOMALIAS (PYCARET)

Siga os passos abaixo:

- **Importe e explore o dataset.**

Descreva estatísticas, tipos de variáveis e possíveis outliers.

- **Configure o PyCaret para Anomaly Detection com setup().**

Obs.: Não existe variável alvo; o objetivo é identificar pontos com comportamento anormal.

- **Crie e compare diferentes modelos de detecção de anomalias, como:**
 - IForest
 - LOF
 - PCA
 - KNN

- Opcional: Realize ajuste de hiperparâmetros com tune_model().
- **Avalie e visualize o modelo final com plot_model().**

Use gráficos como:

- anomaly score plot
- tsne
- umap
- feature importance

- **Interprete o resultado:**

Explique quais observações foram detectadas como anômalas e o motivo.

- **Discuta o uso em produção:**

Comente sobre:

- qualidade do modelo
- viabilidade de aplicação em ambiente de produção
- necessidade de atualização



- custo de manutenção

CASO A SUA TAREFA SEJA REGRAS DE ASSOCIAÇÃO (MLXTEND)

Siga os passos abaixo:

- Importe e explore o dataset.

Explique o formato das transações e o significado de cada variável.

- Prepare o dataset para Association Rules.

Caso esteja em formato tabular, pode ser necessário convertê-lo para formato transacional.

- Gere itemsets frequentes usando **MLxtend**
- Gere regras de associação com métodos como Apriori ou FP-Growth usando **MLxtend** (instalação → !pip install mlxtend)

```
from mlxtend.frequent_patterns import apriori, fpgrowth
```

- Analise as regras obtidas, observando:
 - support
 - confidence
 - lift
- Visualize os resultados.

A MLxtend não possui gráficos prontos, então você deverá produzi-los com matplotlib ou seaborn.scatter plot.

Sugestões de visualizações:

- Scatter plot (support × confidence, cor = lift)
- Matrix plot mostrando produtos mais frequentes
- Graph network (utilizando networkx ou pyvis) para ver relações entre itens
- Gráfico 3D (support × confidence × lift)



- Interprete as regras:

Explique o significado das associações descobertas e possíveis aplicações práticas.

- O que cada regra significa no contexto do dataset.
- Quais insights podem ser aplicados em um cenário real.
- Exemplifique possíveis usos:
 - Sugestão de produtos (cross-selling)
 - Organização de gôndolas
 - Combos de ofertas
 - Sistemas de recomendação baseados em coocorrência de itens

- Discuta a aplicabilidade em produção:

Inclua aspectos como:

- qualidade do modelo
- viabilidade de aplicação em ambiente de produção
- necessidade de atualização
- custo de manutenção

ENTREGA

Cada aluno deverá entregar um arquivo **.ipynb** com:

- todo o código comentado
- descrição da origem dos dados
- explicação dos atributos
- interpretação dos resultados
- justificativa sobre uso em produção

Na próxima aula, cada aluno terá 5 minutos para explicar seu trabalho. Os alunos, na apresentação, devem focar em explicar as métricas e interpretar os resultados alcançados.

A relação de qual dataset corresponde a cada aluno está disponível em arquivo separado.