

Segmentacao.Clientes2

September 25, 2025

Problema de Negócio:

Considerando dados históricos de clientes que realizaram compras em nossa empresa, faça o agrupamento (segmentação) dos clientes por similaridade de características em 3 grupos e envie o relatório para a equipe de Marketing.

```
[2]: # Versão da Linguagem Python
from platform import python_version
print('Versão da Linguagem Python Usada Neste Jupyter Notebook:',
      python_version())
```

Versão da Linguagem Python Usada Neste Jupyter Notebook: 3.10.12

```
[ ]: # Imports
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

0.1 Carregando os Dados

```
[ ]: # Carrega os dados
df_clientes = pd.read_csv('dados/dados_clientes.csv')
```

```
[ ]: type(df_clientes)
```

```
[ ]: # Visualiza as 10 primeiras linhas
df_clientes.head(10)
```

0.2 Análise Exploratória

```
[ ]: # Resumo estatístico
df_clientes[['idade', 'renda_anual', 'pontuacao_gastos']].describe()
```

0.3 Pré-Processamento dos Dados

```
[ ]: # Cria o padronizador dos dados
padronizador = StandardScaler()

[ ]: # Aplica o padronizador somente nas colunas de interesse
dados_padronizados = padronizador.fit_transform(df_clientes[['idade',
↪ 'renda_anual', 'pontuacao_gastos']])

[ ]: # Visualiza os dados
print(dados_padronizados)
```

0.4 Construção do Modelo de Machine Learning Para Segmentação de Clientes

```
[ ]: # Definimos o número de clusters (k)
k = 3

[ ]: # Criamos o modelo K-means
kmeans = KMeans(n_clusters = k)

[ ]: # Treinamento do modelo com os dados padronizados
kmeans.fit(dados_padronizados)

[ ]: # Atribuimos os rótulos dos clusters aos clientes
df_clientes['cluster'] = kmeans.labels_

[ ]: # Exibe o resultado (10 primeiras linhas)
df_clientes.head(10)

[ ]: # Salvamos o resultado em disco
df_clientes.to_csv('dados/segmentos.csv', index = False)
```


0.5 Gerando Relatório dos Segmentos de Clientes com Power BI no Jupyter Notebook

<https://pypi.org/project/powerbiclient/>

```
[ ]: # Instala o pacote
!pip install -q powerbiclient

[ ]: # Carrega as funções usadas para autenticar e gerar relatórios
from powerbiclient import QuickVisualize, get_dataset_config, Report
from powerbiclient.authentication import DeviceCodeLoginAuthentication

[ ]: # Define a autenticação no Power BI Service
device_auth = DeviceCodeLoginAuthentication()
```

```
[ ]: # Cria o relatório no Power BI
relatorio_PBI = QuickVisualize(get_dataset_config(df_clientes), auth = ,
↪device_auth)
```

```
[ ]: # Renderiza (visualiza) o relatório
relatorio_PBI
```

1 Fim