Inteligência Artificial e Big Data
Aula 15

Prof. Me Daniel Vieira

SENAI

Agenda

1 Exercício - algoritmos K-means
2 - Exercício algoritmos K Means
3 - Atividade EAD

Exercícios

Agrupamento de Ações da Bolsa de Valores com K-Means
Objetivo: Utilizar o algoritmo K-Means para agrupar ações da bolsa de valores com base em características como preço, volume de negociação e capitalização de mercado.

Exercícios

Símbolo	Preço da Ação	Volume de Negociação	Capitalização de Mercado
AAPL	150.12	3000000	2000000000000
GOOGL	2700.45	1000000	1800000000000
MSFT	300.89	2500000	2200000000000
AMZN	3200.10	1200000	1700000000000
TSLA	700.78	5000000	750000000000
FB	350.40	900000	900000000000

Exercícios

Importe os dados e visualize o conjunto de dados para entender a estrutura das informações.

Utilize a biblioteca Python e o algoritmo K-Means para agrupar as ações com base nas características de preço, volume de negociação e capitalização de mercado.

Execute o algoritmo K-Means com um número arbitrário de clusters, por exemplo, 4 clusters, e análise os resultados.

Visualize os grupos formados em um gráfico tridimensional, onde os eixos representam o preço, volume de negociação e capitalização de mercado.

Exercícios

```python
# Aula 15 IABD
import numpy as np # biblioteca para criar arrays numpy
import pandas as pd # biblioteca que permite a criação de um dataframe
from sklearn.cluster import KMeans # biblioteca que permite criar clusters com o algoritmo kmeans
import seaborn as sns # biblioteca para plotar gráficos com um visual mais atraente
import matplotlib.pyplot as plt # biblioteca matplotlib
from mpl_toolkits.mplot3d import Axes3D # biblioteca que permite criar gráficos 3d
```

✓ 4.7s Python

```python
# cria base de dados
dados = pd.DataFrame({
    'Símbolo':["AAPL","Google","MSFT","AMZN","TSLA","FB"],
    'Preço':[150.12,2700.45,300.89,3200.10,700.78,350.40],
    'Volume de negociação':[3000000,1000000, 2500000,  1200000,5000000,900000 ],
    'Capitalização de mercado':[2000000000000, 1800000000000,2200000000000 ,1700000000000,750000000000 ,
                                900000000000   ]

})
```

✓ 0.0s Python

Exercícios

```
dados.head() # visualização das primeiras linhas
```

✓ 0.0s Python

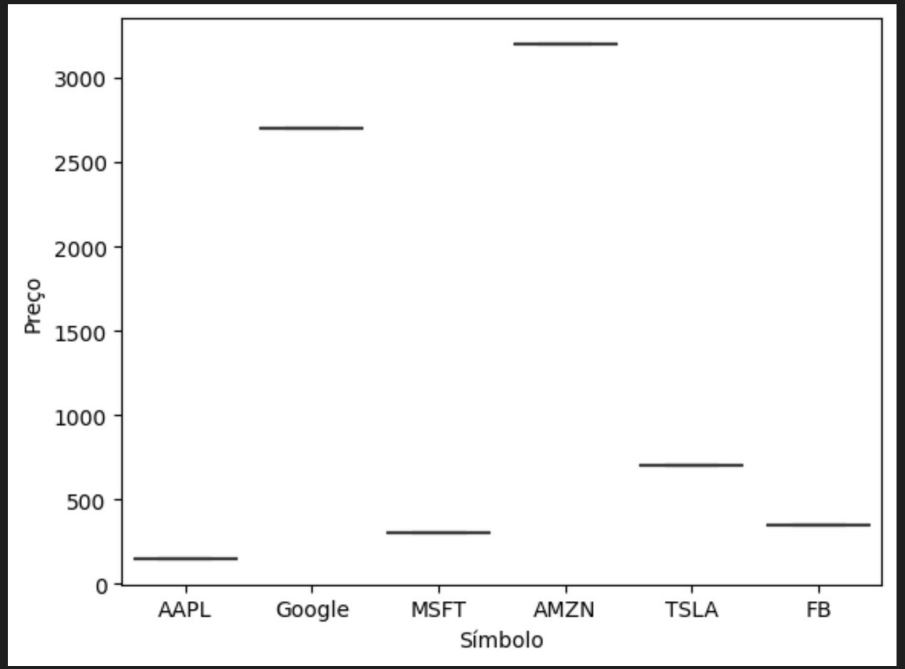
	Símbolo	Preço	Volume de negociação	Capitalização de mercado
0	AAPL	150.12	3000000	2000000000000
1	Google	2700.45	1000000	1800000000000
2	MSFT	300.89	2500000	2200000000000
3	AMZN	3200.10	1200000	1700000000000
4	TSLA	700.78	5000000	750000000000

Exercícios

```
sns.boxplot(x = dados['Símbolo'],y=dados['Preço']) #plota boxplot
```

✓ 0.4s

```
<Axes: xlabel='Símbolo', ylabel='Preço'>
```



Exercícios

```
dados_ativos = pd.get_dummies(dados,columns=['Símbolo']) # transforma variaveis categóricas em true ou false
```

✓ 0.0s Python

```
dados_ativos # exibe dados após transformação
```

✓ 0.0s Python

	Preço	Volume de negociação	Capitalização de mercado	Símbolo_AAPL	Símbolo_AMZN	Símbolo_FB	Símbolo_Google	Símbolo_MSFT	Símbolo_TSLA
0	150.12	3000000	2000000000000	True	False	False	False	False	False
1	2700.45	1000000	1800000000000	False	False	False	True	False	False
2	300.89	2500000	2200000000000	False	False	False	False	True	False
3	3200.10	1200000	1700000000000	False	True	False	False	False	False
4	700.78	5000000	750000000000	False	False	False	False	False	True
5	350.40	900000	900000000000	False	False	True	False	False	False

Exercícios

```python
kmeans = KMeans(n_clusters=4) # cria 4 clusters
kmeans.fit(dados_ativos)
```

✓ 0.4s Python

```
c:\Users\Eng. Daniel Vieira\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The defa
  super()._check_params_vs_input(X, default_n_init=10)
```

▾ KMeans

KMeans(n_clusters=4)

```python
sse = kmeans.inertia_ # métrica inertia
print("SSE",sse)
```

✓ 0.0s Python

```
SSE 1.6250000008425e+22
```

Exercícios

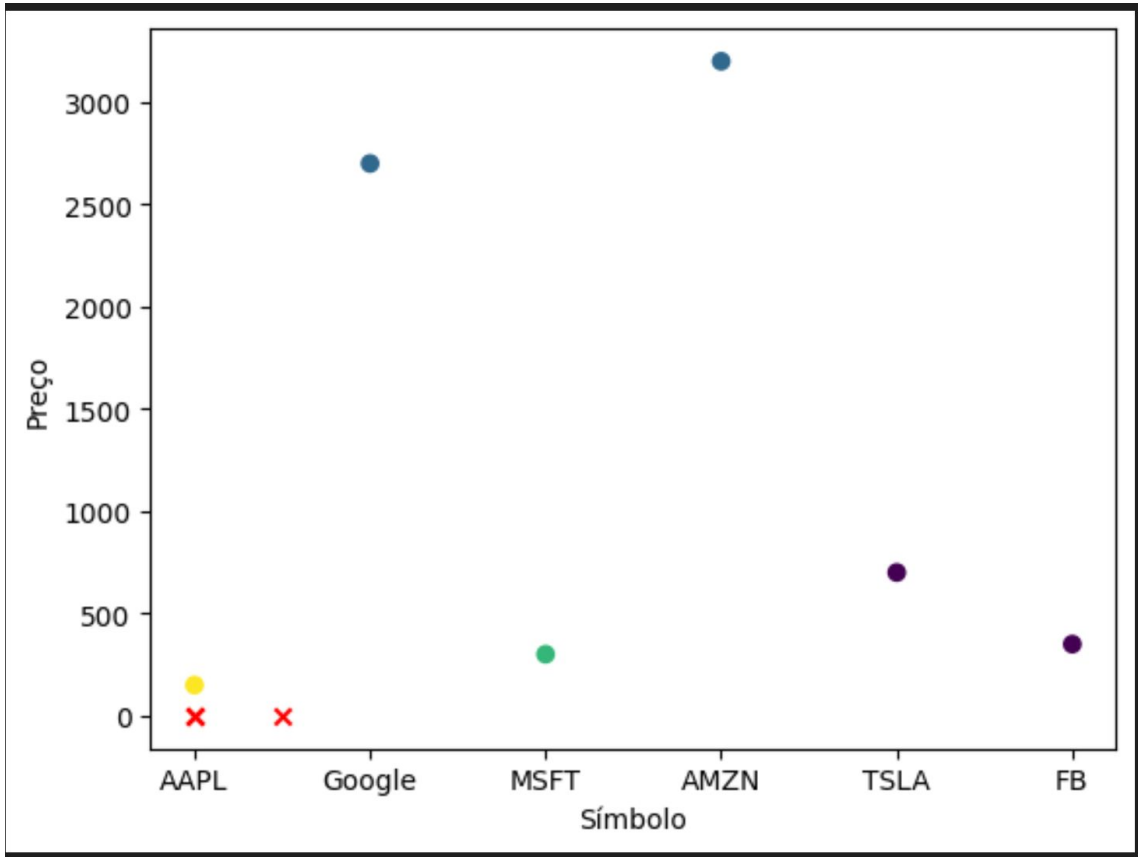
```python
# Visualizar os resultados do agrupamento
labels = kmeans.labels_ # Nomes dos itens do agrupamento
centroids = kmeans.cluster_centers_ # Valores do interior do agrupamento
```

✓ 0.0s Python

```python
# Plotando o gráfico das visualizações

plt.scatter(dados['Símbolo'],dados['Preço'], c= labels) # plota  o gráfico com a visualização dos clusters
plt.scatter(centroids[:, 4], centroids[:, 4], marker='x', color='red')
# : indica o começo  e 4 até qual linha vai

plt.xlabel("Símbolo")
plt.ylabel("Preço")
```

✓ 0.4s Python

Exercícios


3000
2500
2000
1500
1000
500
0
Preço
AAPL
Google
MSFT
AMZN
TSLA
FB
Símbolo

Exercícios

```python
# Passo 7: Analisar os grupos resultantes
dados_ativos['Grupo'] = labels
grouped_data = dados_ativos.groupby('Grupo').mean()
print(grouped_data)
```

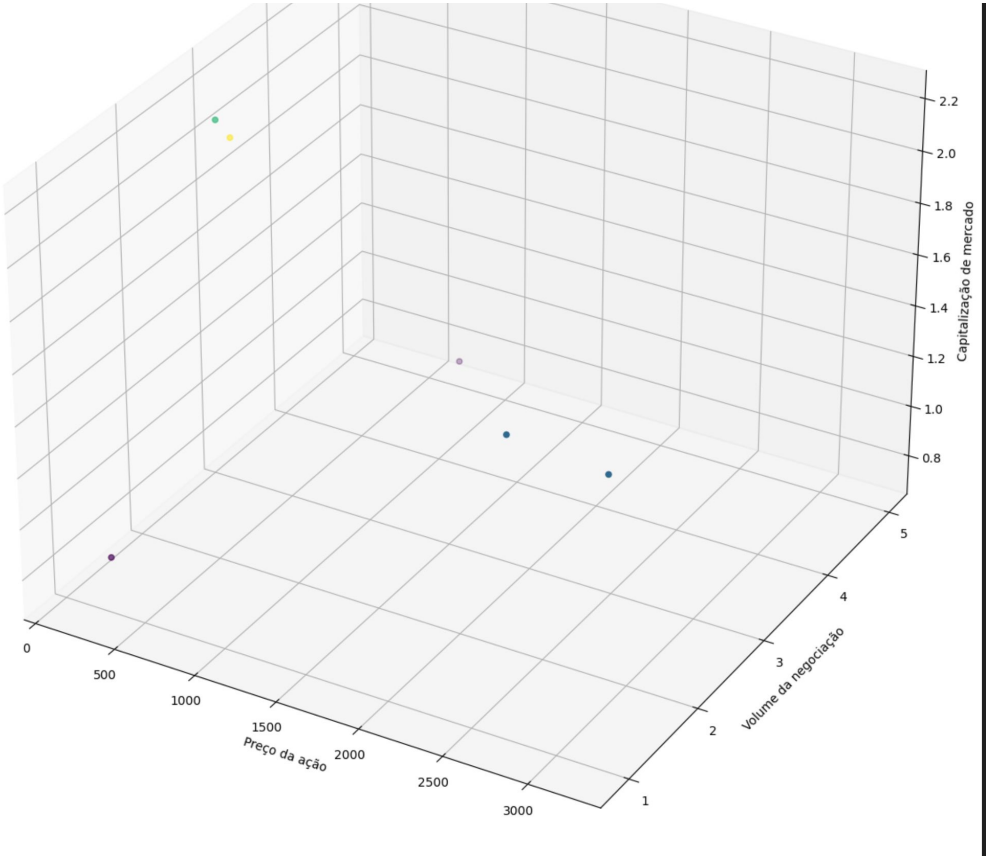
✓ 0.0s Python

```
          Preço  Volume de negociação  Capitalização de mercado  Símbolo_AAPL  \
Grupo
0       525.590             2950000.0              8.250000e+11           0.0
1      2950.275             1100000.0              1.750000e+12           0.0
2       300.890             2500000.0              2.200000e+12           0.0
3       150.120             3000000.0              2.000000e+12           1.0

       Símbolo_AMZN  Símbolo_FB  Símbolo_Google  Símbolo_MSFT  Símbolo_TSLA
Grupo
0               0.0         0.5             0.0           0.0           0.5
1               0.5         0.0             0.5           0.0           0.0
2               0.0         0.0             0.0           1.0           0.0
3               0.0         0.0             0.0           0.0           0.0
```

Exercícios

```
# Criando o gráfico em 3D
from mpl_toolkits.mplot3d import Axes3D
# Criando o gráfico em 3D
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(dados['Preço'], dados['Volume de negociação'], dados['Capitalização de mercado'], c=labels)


ax.set_title('Kmeans clusters encontrados' )
ax.set_xlabel('Preço da ação')
ax.set_ylabel('Volume da negociação')
ax.set_zlabel('Capitalização de mercado')

plt.show()
```

Exercícios


Capitalização de mercado
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
Volume da negociação
1
2
3
4
5
Preço da ação
0
500
1000
1500
2000
2500
3000

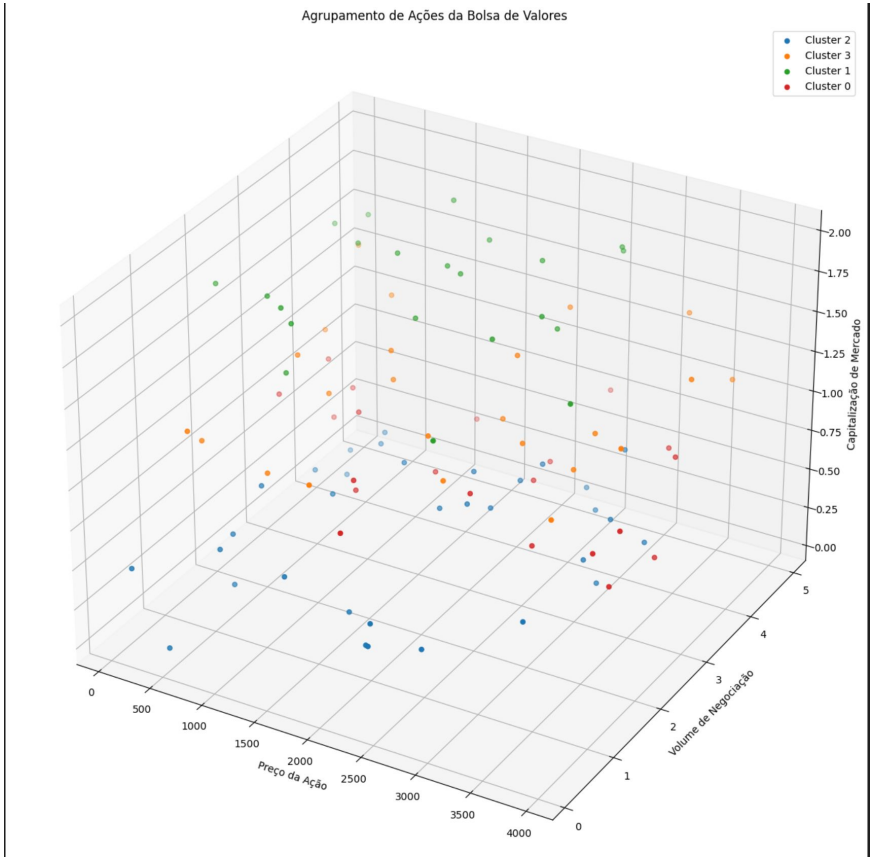
Exercícios - Gerando dados aleatórios

```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
# Gerar dados fictícios de ações
np.random.seed(0)
n_samples = 100
symbols = ['AAPL', 'GOOGL', 'MSFT', 'AMZN', 'TSLA', 'FB']
prices = np.random.uniform(50, 4000, n_samples)
volume = np.random.randint(100000, 5000000, n_samples)
market_cap = np.random.uniform(1e10, 2e12, n_samples)

data = pd.DataFrame({'Símbolo': np.random.choice(symbols, n_samples),
                     'Preço da Ação': prices,
                     'Volume de Negociação': volume,
                     'Capitalização de Mercado': market_cap})

# Realizar o agrupamento com K-Means
data = pd.get_dummies(data, columns=['Símbolo'], drop_first=True)
features = data[['Preço da Ação', 'Volume de Negociação', 'Capitalização de Mercado']]
kmeans = KMeans(n_clusters=4, random_state=0).fit(features)
```

Exercícios - Gerando dados aleatórios

```
# Visualizar os grupos em um gráfico tridimensional
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(111, projection='3d')

for cluster in data['Cluster'].unique():
    cluster_data = data[data['Cluster'] == cluster]
    ax.scatter(cluster_data['Preço da Ação'], cluster_data['Volume de Negociação'], cluster_data['Capitalização de Mercado'], label=f'Cluster {cluster}')

ax.set_xlabel('Preço da Ação')
ax.set_ylabel('Volume de Negociação')
ax.set_zlabel('Capitalização de Mercado')
ax.set_title(f'Agrupamento de Ações da Bolsa de Valores ')

plt.legend()
plt.show()

# Analisar os clusters
cluster_means = data.groupby('Cluster').mean()
print(cluster_means)
```

Exercícios - Gerando dados aleatórios



Exercícios - Gerando dados aleatórios

Cluster	Preço da Ação	Volume de Negociação	Capitalização de Mercado
0	2292.157282	2.600392e+06	8.548502e+11
1	1721.803918	2.772412e+06	1.812930e+12
2	1669.222561	2.546484e+06	3.508185e+11
3	2110.593040	2.425316e+06	1.312753e+12

Cluster	Símbolo_AMZN	Símbolo_FB	Símbolo_GOOGL	Símbolo_MSFT	Símbolo_TSLA
0	0.238095	0.142857	0.190476	0.047619	0.190476
1	0.227273	0.136364	0.181818	0.090909	0.181818
2	0.212121	0.181818	0.181818	0.181818	0.151515
3	0.250000	0.083333	0.208333	0.000000	0.250000

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95