

***INTELIGÊNCIA ARTIFICIAL***

**LABORATÓRIO # 11**

**ASSUNTO:** AGRUPAMENTO K-MEANS USANDO PYTHON

**Materiais de Apoio**

**Site oficial do Scikit Learn:**

**scikit-learn.org/**

**Clustering em Python (português):**

[**https://minerandodados.com.br/analise-e-agrupamento-de-clientes-com-machine-learning-k-means/**](https://minerandodados.com.br/analise-e-agrupamento-de-clientes-com-machine-learning-k-means/)

**Material bem interessante para Clustering em Python (português):**

[**https://www.youtube.com/watch?v=piDlrtfz58s**](https://www.youtube.com/watch?v=piDlrtfz58s)

**Exemplo - Importação de Bibliotecas, Preparação de Dados e Teste**

***Dataset usado:*** *bmw.csv*

*# Importação de Bibliotecas*

*import matplotlib.pyplot as plt*

*import pandas as pd*

*from sklearn.cluster import KMeans*

*# Leitura de Dados no formato .CSV*

*bmw = pd.read\_csv("bmw.csv", sep=';')*

*bmw.head() # primeiros 5 registros*

*# Informações quantitativas do dataset: 100 linhas e 9 colunas*

*bmw.shape*

*# Para Agrupamento (clustering), feature "id" não é bem-vinda!*

*# Vamos excluir o CustomerID..Vamos lá !*

*# Agora ficará: 0.Dealership, 1.Showroom, 2.ComputerSearch, 3.M5, 4.3Series, 5.Z4, 6.Financing, 7.Purchase*

*Xbmw = bmw.iloc[:, 1:9].values*

*# Criação do modelo de Agrupamento (Clustering) usando K-Means*

*# Parametrizando K-Means com 3 Clusters e Metodo de Inicialização Randômica*

*kmeans = KMeans(n\_clusters = 3, init = 'random')*

*kmeans*

*# Treino do modelo com os Dados*

*kmeans.fit(Xbmw)*

*# Exibindo os Centroids - São 3 grupos, portanto, 3 linhas com 8 colunas = 8 features*

*kmeans.cluster\_centers\_*

*# Treinando novamente o modelo com os Dados - Observe que novos Centroids são calculados (inicialização randômica!)*

*kmeans.fit(Xbmw)*

*# Exibindo os Centroids - São 3 grupos, portanto, 3 linhas com 8 colunas = 8 features*

*kmeans.cluster\_centers\_*

*# Executa K-Means para agrupar os dados e retorna Tabela de Distancias (fit\_transform)*

*# Calcula a distância entre os dados de cada linha (instância) e cada um dos 3 clusters*

*distance = kmeans.fit\_transform(Xbmw)*

*distance*

*# Para cada uma das 100 linhas do dataset, a qual dos 3 clusters ela pertence..0, 1 ou 2 ?*

*# São 100 elementos no array, um por linha*

*labels = kmeans.labels\_*

*labels*

*# Exibindo novamente os Centroids (atualizados) - São 3 grupos, portanto, 3 linhas com 8 colunas = 8 features*

*kmeans.cluster\_centers\_*

*# Visualizando os Clusters - Exemplo 1*

*# Dados: 0.Dealership, 1.Showroom, 2.ComputerSearch, 3.M5, 4.3Series, 5.Z4, 6.Financing, 7.Purchase*

*plt.scatter(Xbmw[:, 6], Xbmw[:,7], s = 100, c = kmeans.labels\_)*

*plt.scatter(kmeans.cluster\_centers\_[:, 6], kmeans.cluster\_centers\_[:, 7], s = 20, c = 'red')*

*plt.title('Customer Clusters and Centroids')*

*plt.xlabel('Financing')*

*plt.ylabel('Purchase')*

*plt.show()*

*# Visualizando os Clusters - Exemplo 2*

*# Dados: 0.Dealership, 1.Showroom, 2.ComputerSearch, 3.M5, 4.3Series, 5.Z4, 6.Financing, 7.Purchase*

*plt.scatter(Xbmw[:, 3], Xbmw[:, 7], s = 100, c = kmeans.labels\_)*

*plt.scatter(kmeans.cluster\_centers\_[:, 3], kmeans.cluster\_centers\_[:, 7], s = 20, c = 'red')*

*plt.title('Customer Clusters and Centroids')*

*plt.xlabel('M5')*

*plt.ylabel('Purchase')*

*plt.show()*

*# Cálculo do WCSSE - Within Cluster Sum of Squared Errors*

*wcsse = []*

*maxclusters = 11*

*for i in range(1, maxclusters):*

*kmeans = KMeans(n\_clusters = i, init = 'random')*

*kmeans.fit(Xbmw)*

*print (i,kmeans.inertia\_)*

*wcsse.append(kmeans.inertia\_)*

*# Plotagem do Gráfico para visualização do "cotovelo" - Método de Elbow*

*# Ao final do Estudo do Metodo de Elbow a seguir, foram gerados 11 clusters*

*plt.plot(range(1, maxclusters), wcsse)*

*plt.title('O Metodo Elbow')*

*plt.xlabel('Qtde, de Clusters')*

*plt.ylabel('WCSSE') # Within Cluster Sum of Squares*

*plt.show()*

*# Para o cliente 1 => Dealership=1, Showroom=1, ComputerSearch=1, M5=1, 3Series=1,*

*# Z4=1, Financing=1, Purchase=1, qual grupo ele pertence?*

*# O comando predict nos traz o cluster=4*

*newcustomer1= [ [1, 1, 1, 1, 1, 1, 1, 1] ]*

*kmeans.predict(newcustomer1)*

*# Para o cliente 2 => Dealership=1, Showroom=0, ComputerSearch=0, M5=0, 3Series=1,*

*# Z4=1, Financing=1, Purchase=1, qual grupo ele pertence?*

*# O comando predict nos traz o cluster=2*

*newcustomer2 = [ [1, 0, 0, 0, 1, 1, 1, 1] ]*

*kmeans.predict(newcustomer2)*

**Bom Trabalho!!**