Code for the Customer Segmentation using K-means clustering(ML + Big data).

* import pandas as pd
* data = pd.read\_csv("Mall\_Customers.csv")
* data.head()
* data
* data.tail()
* print("Number of rows",data.shape[0])

print("Number of columns",data.shape[1])

* data.info()
* data.describe()
* data.columns
* X = data[['Annual Income (k$)','Spending Score (1-100)']] # selecting required columns
* X
* from sklearn.cluster import KMeans
* k\_means=KMeans()

k\_means.fit(X)

* k\_means=KMeans()

k\_means.fit\_predict(X)

* wcss=[]

for i in range(1,11):

k\_means =KMeans(n\_clusters=i)

k\_means.fit(X)

wcss.append(k\_means.inertia\_)

* wcss
* import matplotlib.pyplot as plt
* plt.plot(range(1,11),wcss)
* plt.plot(range(1,11),wcss)
* plt.title("Elbow method")
* plt.xlabel("No. of clusters")
* plt.ylabel("wcss")
* plt.show()
* X = data[['Annual Income (k$)','Spending Score (1-100)']]
* k\_means = KMeans(n\_clusters=5,random\_state=42)
* y\_means = k\_means.fit\_predict(X)
* y\_means #customer segmented in 5 clusters
* plt.scatter(X.iloc[y\_means==0,0],X.iloc[y\_means==0,1],s=100,c="skyblue",label="Cluster 1") #segmentation

plt.scatter(X.iloc[y\_means==1,0],X.iloc[y\_means==1,1],s=100,c="silver",label="Cluster 2")

plt.scatter(X.iloc[y\_means==2,0],X.iloc[y\_means==2,1],s=100,c="yellow",label="Cluster 3")

plt.scatter(X.iloc[y\_means==3,0],X.iloc[y\_means==3,1],s=100,c="orange",label="Cluster 4")

plt.scatter(X.iloc[y\_means==4,0],X.iloc[y\_means==4,1],s=100,c="gray",label="Cluster 5")

plt.legend

* plt.scatter(X.iloc[y\_means==0,0],X.iloc[y\_means==0,1],s=100,c="skyblue",label="Cluster 1") # naming

plt.scatter(X.iloc[y\_means==1,0],X.iloc[y\_means==1,1],s=100,c="silver",label="Cluster 2")

plt.scatter(X.iloc[y\_means==2,0],X.iloc[y\_means==2,1],s=100,c="yellow",label="Cluster 3")

plt.scatter(X.iloc[y\_means==3,0],X.iloc[y\_means==3,1],s=100,c="orange",label="Cluster 4")

plt.scatter(X.iloc[y\_means==4,0],X.iloc[y\_means==4,1],s=100,c="gray",label="Cluster 5")

plt.title("Customer Segmentation")

plt.xlabel("Annual income")

plt.ylabel("Spending score")

plt.show()

plt.legend

* plt.scatter(X.iloc[y\_means==0,0],X.iloc[y\_means==0,1],s=100,c="skyblue",label="Cluster 1") # naming

plt.scatter(X.iloc[y\_means==1,0],X.iloc[y\_means==1,1],s=100,c="silver",label="Cluster 2")

plt.scatter(X.iloc[y\_means==2,0],X.iloc[y\_means==2,1],s=100,c="yellow",label="Cluster 3")

plt.scatter(X.iloc[y\_means==3,0],X.iloc[y\_means==3,1],s=100,c="orange",label="Cluster 4")

plt.scatter(X.iloc[y\_means==4,0],X.iloc[y\_means==4,1],s=100,c="gray",label="Cluster 5")

plt.scatter(k\_means.cluster\_centers\_[:,0],k\_means.cluster\_centers\_[:,1],s=100,c="black")

plt.title("Customer Segmentation")

plt.xlabel("Annual income")

plt.ylabel("Spending score")

plt.show()

plt.legend

* k\_means.predict([[78,22]])
* import joblib
* joblib.dump(k\_means,"Customer\_segmentation")
* p = joblib.load("Customer\_segmentation")
* p.predict([[78,22]])
* predict([[24,34]])