Name: Madhusudan D. Pangarkar

Class: TEIT Roll No.: 41

Sub: Laboratory Practice-I (Machine Learning) (314448)

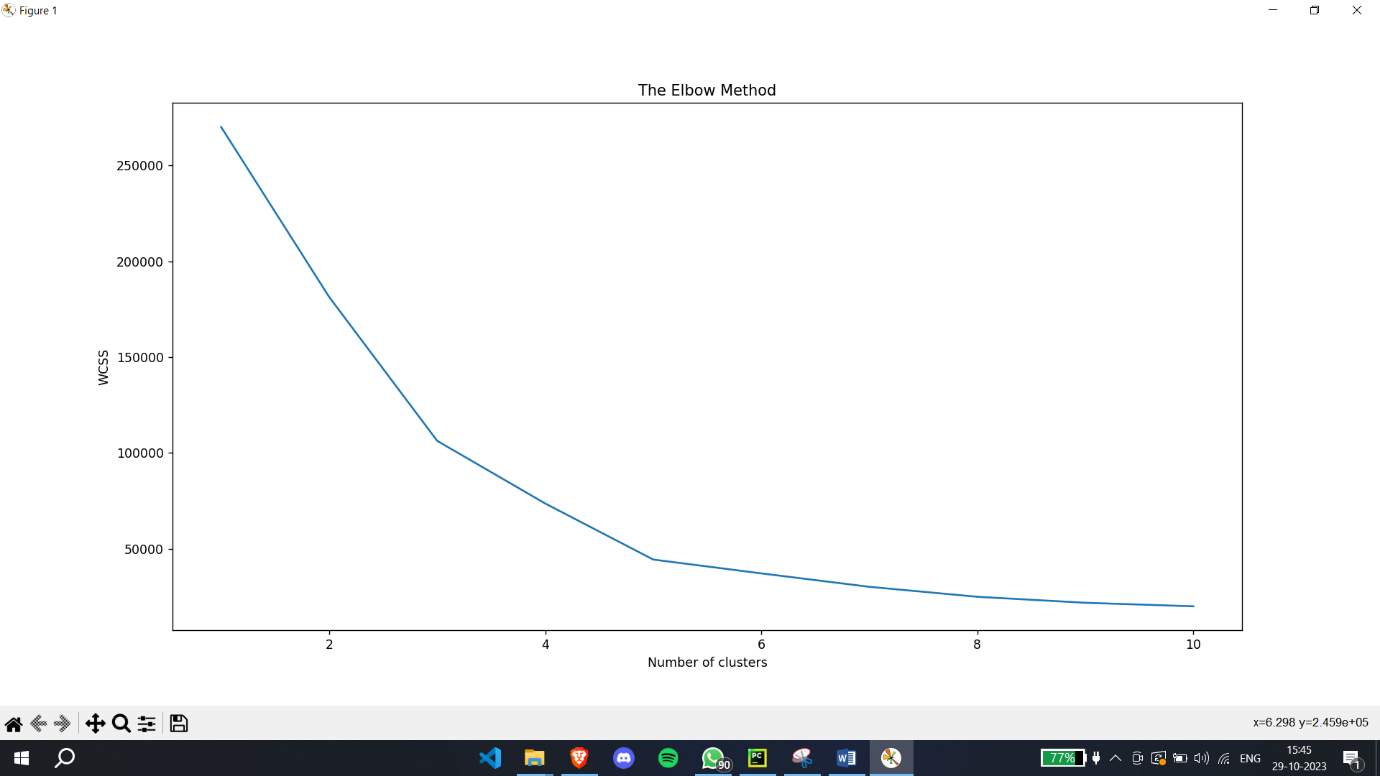
------------------------------------------------------------------------------------------------

**Assignment No-04) Assignment on Clustering Techniques**

**Code:**

import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
  
##Importing the dataset¶  
dataset = pd.read\_csv('E:\ML Practicals\Mall\_Customers.csv')  
X = dataset.iloc[:, [3, 4]].values  
  
##Using the elbow method to find the optimal number of clusters  
from sklearn.cluster import KMeans  
wcss = []  
for i in range(1, 11):  
 kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state = 42)  
 kmeans.fit(X)  
 wcss.append(kmeans.inertia\_)  
plt.plot(range(1, 11), wcss)  
plt.title('The Elbow Method')  
plt.xlabel('Number of clusters')  
plt.ylabel('WCSS')  
plt.show()  
  
##Training the K-Means model on the dataset¶  
kmeans = KMeans(n\_clusters = 5, init = 'k-means++', random\_state = 42)  
y\_kmeans = kmeans.fit\_predict(X)  
  
##Visualising the clusters  
plt.scatter(X[y\_kmeans == 0, 0], X[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')  
plt.scatter(X[y\_kmeans == 1, 0], X[y\_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')  
plt.scatter(X[y\_kmeans == 2, 0], X[y\_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')  
plt.scatter(X[y\_kmeans == 3, 0], X[y\_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')  
plt.scatter(X[y\_kmeans == 4, 0], X[y\_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')  
plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s = 300, c = 'yellow', label = 'Centroids')  
plt.title('Clusters of customers')  
plt.xlabel('Annual Income (k$)')  
plt.ylabel('Spending Score (1-100)')  
plt.legend()  
plt.show()  
  
##Hierarchical Clustering  
##Importing the libraries¶  
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
##Importing the dataset  
dataset = pd.read\_csv('Mall\_Customers.csv')  
X = dataset.iloc[:, [3, 4]].values  
##Using the dendrogram to find the optimal number of clusters¶  
import scipy.cluster.hierarchy as sch  
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))  
plt.title('Dendrogram')  
plt.xlabel('Customers')  
plt.ylabel('Euclidean distances')  
plt.show()  
##Training the Hierarchical Clustering model on the dataset  
from sklearn.cluster import AgglomerativeClustering  
hc = AgglomerativeClustering(n\_clusters = 5, affinity = 'euclidean', linkage = 'ward')  
y\_hc = hc.fit\_predict(X)  
##Visualising the clusters  
plt.scatter(X[y\_hc == 0, 0], X[y\_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')  
plt.scatter(X[y\_hc == 1, 0], X[y\_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')  
plt.scatter(X[y\_hc == 2, 0], X[y\_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')  
plt.scatter(X[y\_hc == 3, 0], X[y\_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')  
plt.scatter(X[y\_hc == 4, 0], X[y\_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')  
plt.title('Clusters of customers')  
plt.xlabel('Annual Income (k$)')  
plt.ylabel('Spending Score (1-100)')  
plt.legend()  
plt.show()

**Output:**

****

