**U19EC046 | ML | LAB 9**

**AIM**

Write a program to implement the K-mean clustering for a sample training data set stored as a .CSV file.

**THEORY**

K- Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.

Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

**STEPS**

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

**ELBOW METHOD**

This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

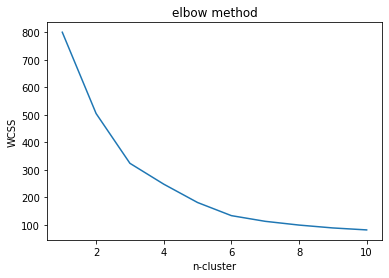
To find the optimal value of clusters, the elbow method follows the below steps:

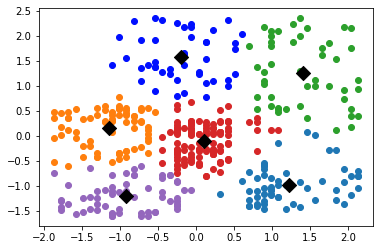
* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WCSS value.
* Plots a curve between calculated WCSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

**CODE**

|  |
| --- |
| import matplotlib.pyplot as plt  import pandas as pd  import numpy as np  from sklearn.cluster import KMeans  from sklearn import preprocessing  data = pd.read\_csv('/content/Social\_Network\_Ads.csv')  X = data.iloc[:, 0:-1]  scaler = preprocessing.StandardScaler().fit(X)  X = scaler.transform(X)  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('elbow method')  plt.xlabel('n-cluster')  plt.ylabel('WCSS')  plt.show()  kmeans = KMeans(n\_clusters=6, init = 'k-means++', random\_state=42)  kmeans.fit(X)  res = kmeans.predict(X)  centers = kmeans.cluster\_centers\_  colors = ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#000fff']  for i, e in enumerate(X):    plt.scatter(e[0], e[1], c=colors[res[i]])  for [x, y] in centers:    plt.scatter(x, y, c='black', marker = 'D', s=100)  plt.show() |

**OUTPUT**





**CONCLUSION**

In this Practical we have studied and implemented K-means clustering algorithm and also understood how to choose the cluster size using the elbow method.