**GROUP – B**

**Assignment No: 5**

====================================================================

**Title: -** Implement K-Means clustering/ hierarchical clustering on sales\_data\_sample.csv dataset.

Determine the number of clusters using the elbow method.

=====================================================================**Objective:-**

-To learn about K-Mean Algorithm

- To understand the concept of clustering as well as elbow method

=====================================================================

**Theory:-**

* **Clustering**

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

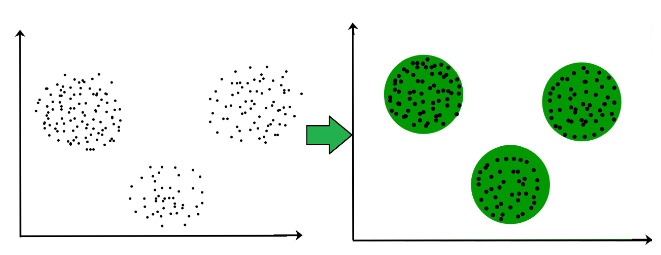


Figure.1. Clustering

For example: The data points in the graph above figure.1, clustered together can be classified into one single group. It can distinguish the clusters, and identify that there are 3 clusters in Figure.1.

* **K-Mean Clustering Algorithm**
* K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning. K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. The K defines the number of pre-defined clusters that need to be created in the process.
* It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
* 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:

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

* Hence each cluster has datapoints with some commonalities, and it is away from other clusters.
* The below diagram explains the working of the K-means Clustering Algorithm:

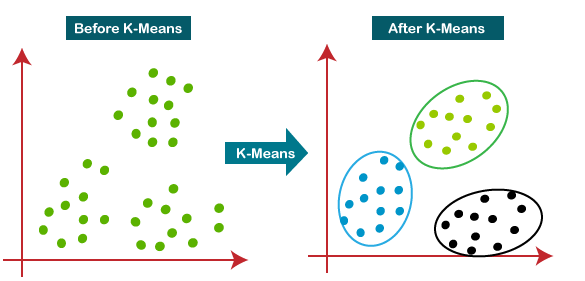


Figure. 2. K-Mean Clustering

* The working of the K-Means algorithm is explained in the below 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 mean 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.

* The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K.
* **Elbow Method**
* The Elbow method is one of the most popular ways to find the optimal number of clusters. 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:



* In the above formula of WCSS, ∑Pi in Cluster1 distance(Pi C1)2: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms. To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.
* To find the optimal value of clusters, the elbow method follows the below steps:

1. It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
2. For each value of K, calculates the WCSS value.
3. Plots a curve between calculated WCSS values and the number of clusters K.
4. 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.

* **Dataset Description**

Sample Sales Data, Order Info, Sales, Customer, Shipping, etc., Used for Segmentation, Customer Analytics, Clustering and More. Inspired for retail analytics. This was originally used for Pentaho DI Kettle. In particular, The sales\_data\_sample.csv file contains 25 features with 2823 data. The features or variables are the following:

1. ORDERNUMBER
2. QUANTITYORDERED
3. PRICEEACH
4. ORDERLINENUMBER
5. SALES
6. ORDERDATE
7. STATUS
8. QTR\_ID
9. MONTH\_ID
10. YEAR\_ID
11. PRODUCTLINE
12. MSRP
13. PRODUCTCODE
14. CUSTOMERNAME
15. PHONE
16. ADDRESSLINE1
17. ADDRESSLINE2
18. CITY
19. STATE
20. POSTALCODE
21. COUNTRY
22. TERRITORY
23. CONTACTLASTNAME
24. CONTACTFIRSTNAME
25. DEALSIZE

* Code Explanation:

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as plt*

*from sklearn.cluster import KMeans*

*df.dtypes*

*X = df.iloc[:, [3,4]].values*

*df.info()*

Import all libraries required for read values from dataset. The import seaborn portion of the code tells Python to bring the Seaborn library into your current environment. Iloc[] function, Purely integer-location based indexing for selection by position. .iloc[] is primarily integer position based (from 0 to length-1 of the axis), but may also be used with a boolean array.

*wcss = [] #within cluster sum of square*

*for i in range(1,11):*

*#init argument is the method for initializing the centroid*

*kmeans = KMeans(n\_clusters=i, init="k-means++", random\_state=42)*

*kmeans.fit(X)*

*#we calculate wcss value for each k value*

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

*ks = [1,2,3,4,5,6,7,8,9,10]*

*plt.plot(ks, wcss, 'bx-')*

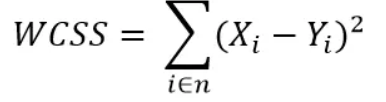
*plt.title("Elbow method")*

*plt.xlabel("K value")*

*plt.ylabel("WCSS")*

*df*

WCSS (Within Cluster Sum of Squares), WCSS is the sum of the squared distance between each point and the centroid in a cluster. Elbow method is one of the robust one used to find out the optimal number of clusters. In this method, the sum of distances of observations from their cluster centroids, called Within-Cluster-Sum-of-Squares (WCSS). This is computed as:



Yi is centroid for observation Xi. The below code performs this method.

*# mean is far from std this indicates high variance*

*from sklearn.preprocessing import StandardScaler*

*ss = StandardScaler()*

*scaled = ss.fit\_transform(X)*

*wcss =[]*

*#this loop will fit the k-means algorithm to our data and*

*#second we will compute the within cluster sum of squares and #appended to our wcss list.*

*for i in range(1,11):*

*clustering =KMeans(n\_clusters=i, init="k-means++", random\_state=42)*

*#i above is between 1-11 numbers. init parameter is the random #initialization method*

*#we select kmeans++ method. max\_iter parameter the maximum number of iterations there can be to*

*#find the final clusters when the K-meands algorithm is running.*

*#the next parameter is n\_init which is the number of times the #K\_means algorithm will be run with*

*#different initial centroid.*

*clustering.fit(scaled)*

*#kmeans algorithm fits to the X dataset*

*wcss.append(clustering.inertia\_)*

*#kmeans inertia\_ attribute is: Sum of squared distances of samples #to their closest cluster center.*

*#4.Plot the elbow graph*

*ks = [1,2,3,4,5,6,7,8,9,10]*

*plt.plot(ks, wcss, 'bx-')*

*plt.title("Elbow method")*

*plt.xlabel("K value")*

*plt.ylabel("WCSS")*

The kNN model, along with our data and a number of splits to make. In the code below, we use five splits which means the model with split the data into five (i.e. knn =5, which mentioned in knn classifier) equal-sized groups. It will loop through each group and give an accuracy score, which we average to find the best model.

=====================================================================

**Conclusion:-**Thus we have studied how to Implement K-Mean Clustering algorithm on sale dataset and determine number of clusters.

=====================================================================