ASSIGNMENT NO. 5

AIM: Assignment on K-means.

PREREQUISITE: Python programming

THEORY:

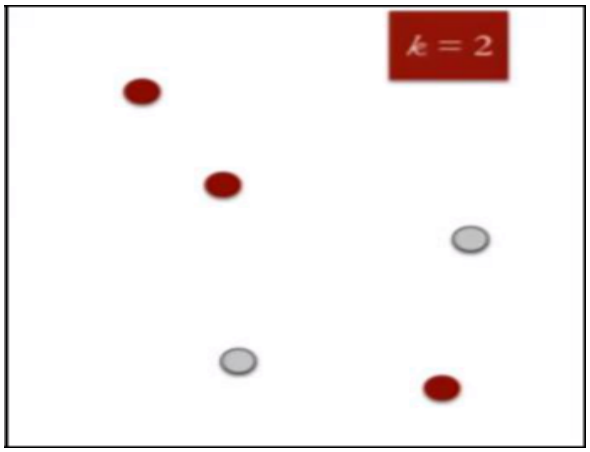
One of the simplest and most widely used unsupervised learning algorithm. It involves a simple way to classify the data set into fixed no. of ***K*** clusters. The idea is to define ***K*** centroids, one for each cluster.

The final clusters depend on the initial configuration of centroids. So, they should be initialized as far from each other as possible.

K-Means is *iterative* in nature and *easy* to implement.

Algorithm Explained

Let there be ***N*** data points. At first, ***K*** centroids are initialised in our data set representing *K* different clusters.

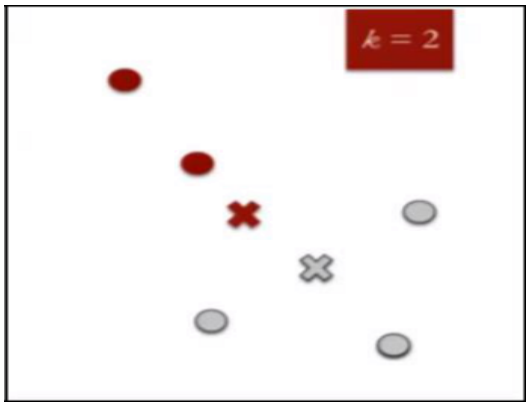


Step 1: N = 5, K = 2

Now, each of the ***N*** data points are assigned to closest centroid in the data set and merged with that centroid as a single cluster. In this way, every data point is assigned to one of the centroids.

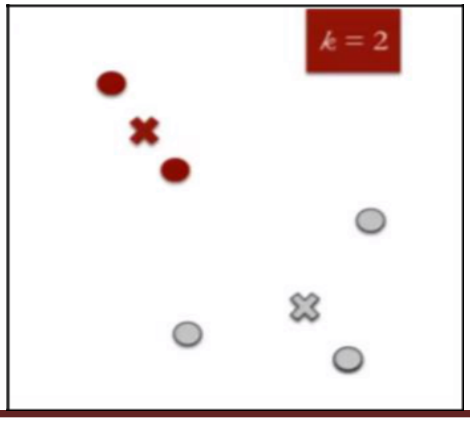
Step 2: Calculating the centroid of the 2 clusters

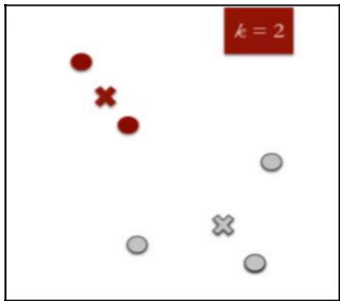
Then, ***K*** cluster centroids are recalculated and again, each of the ***N*** data points are assigned to the nearest centroid.



Step 3: Assigning all the data points to the nearest cluster centroid

Step 3 is repeated until no further improvement can be made.





Step 4: Recalculating the cluster centroid. After this step, no more improvement can be made.

In this process, a loop is generated.

K centroids change their location step by step until no more change is possible. This algorithm aims at minimising the **objective function**:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

It represent the sum of **euclidean distance** of all the data points from the cluster centroid which is minimised.

How to initialize K centroids?

1. **Forgy:** Randomly assigning K centroid points in our data set.

2. **Random Partition:** Assigning each data point to a cluster randomly, and then proceeding to evaluation of centroid positions of each cluster.

3. **KMeans++:** Used for ***small*** data sets.

4. **Canopy Clustering:** Unsupervised pre-clustering algorithm used as preprocessing step for K-Means or any Hierarchical Clustering. It helps in speeding up clustering operations on ***large data sets***.

How to calculate centroid of a cluster?

Simply the mean of all the data points within that cluster.

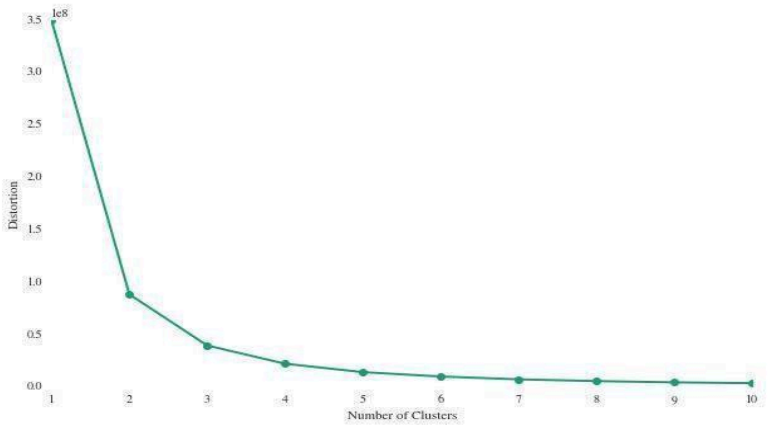
How to find value of K for the dataset?

In K-Means Clustering, value of ***K*** has to be specified beforehand. It can be determine by any of the following methods:

Elbow Method: Clustering is done on a dataset for varying values of and **SSE (Sum of squared errors)** is calculated for each value of *K*.

Then, a graph between *K* and SSE is plotted. Plot formed assumes the shape of an arm. There is a point on the graph where SSE does not decreases significantly with increasing *K*.

This is represented by elbow of the arm and is chosen as the value of *K*. (OPTIMUM)



K-Means v/s Hierarchical

1. For **big data**, ***K-Means*** is better!
   Time complexity of K-Means is linear, while that of hierarchical clustering is quadratic.

2. Results are reproducible in ***Hierarchical***, and not in K-Means, as they depend on intialization of centroids.
3. K-Means requires prior and proper knowledge about the data set for specifying ***K***. In ***Hierarchical***, we can choose no. of clusters by interpreting dendogram.

Some things to take note of clustering as follows

k-means clustering is very sensitive to scale due to its reliance on Euclidean distance so be sure to normalize data if there are likely to be scaling problems.

If there are some symmetries in your data, some of the labels may be mis-labelled
It is recommended to do the same k-means with different initial centroids and take the most common label.

REFERENCE:

[1] http://benalexkeen.com/k-means-clustering-in-python/
[2] https://www.analyticsvidhya.com

CONCLUSION:

K-Means is a simple yet powerful unsupervised learning algorithm widely used for clustering tasks. Through this assignment, we explored how the algorithm works by iteratively assigning data points to the nearest centroids and updating those centroids to minimize intra-cluster variance. Although it requires the number of clusters (K) to be predefined, techniques like the Elbow Method help estimate an optimal K. Despite its sensitivity to scale and initialization, K-Means remains efficient for large datasets and easy to implement. Understanding and applying K-Means provides valuable insights into data distribution and grouping patterns, making it a crucial tool in data science and machine learning.