**Name: Harshitha Komatineni**

**UTA Student ID: 1001968082**

**PROJECT-3 REPORT**

**Problem:**

KMeans is a well-known unsupervised machine learning algorithm. But here we want to use KMeans algorithm as a supervised technique to solve a classification problem.

**Data:**

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository.

It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

The columns in this dataset are:

SepalLengthCm

SepalWidthCm

PetalLengthCm

PetalWidthCm

Species

**Method – KMeans Algorithm:**

Steps followed to build the KMeans algorithm:

Step – 1: Choose the number of clusters (k) – The first step is to pick the number of clusters k (in our case k=3, as there are 3 classes in the dataset). It can be determined by a dendrogram plot or the number of classes in a classification problem.

Step – 2: Select k random points from the data as centroids – After fixing the value of k, we randomly select k points from the data. We will use these points as the initializers for the centroids.

Step – 3: Assign all the points in the data to the nearest centroids – After initializing the centroids we assign each point in the training data to the closest cluster centroid based on Euclidean distances.

Step – 4: Recompute the centroids – After assigning each point to the initial centroids, we update the centroids by taking the averages of the data points in each cluster. Those averages then become the new centroids for the clusters.

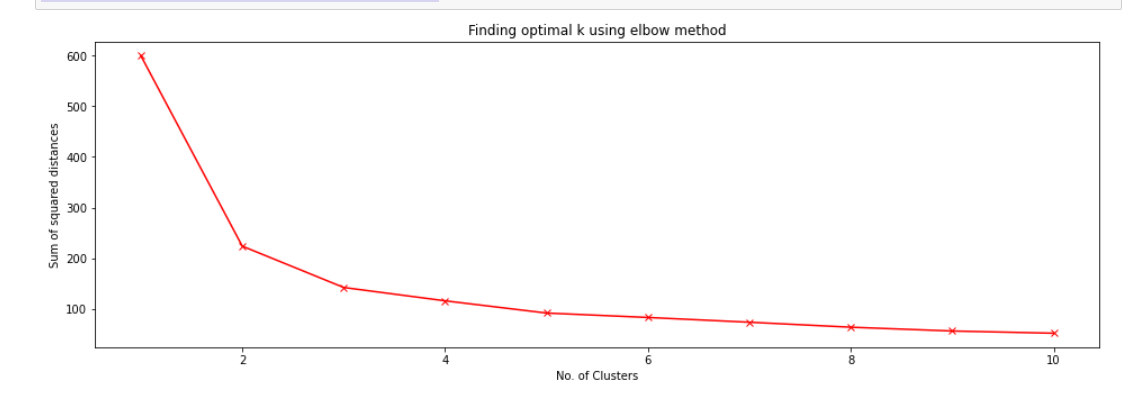
Step – 5: Repeat steps 3 & 4 – We repeat the steps 3 and 4 and update the cluster centroids accordingly. We will stop this process if, 1. Centroids of newly formed clusters do not change, 2. Points remains in the same cluster, 3. Maximum number (predefined) of iterations are reached.

For getting the predictions for new datapoints using a trained KMeans algorithm, we calculate the Euclidean distances of the data points from the cluster centroids and assign the datapoints to the cluster from which the distance is minimum.

**Results:**

Our algorithm can predict the classes with an accuracy of **85.33%**. So, this KMeans algorithm can be used as a classification algorithm in our regular practice.

**Note**: The N (number of clusters) can be decided on the basis of the elbow formed by plotting a graph between no. of clusters vs. the sum of square distances between the centroids.



**REFERENCES:**

1. [**https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670**](https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670)
2. [**https://medium.com/nerd-for-tech/k-means-python-implementation-from-scratch-8400f30b8e5c**](https://medium.com/nerd-for-tech/k-means-python-implementation-from-scratch-8400f30b8e5c)
3. [**https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/**](https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/)