# Clustering Assignment

Assignment 4

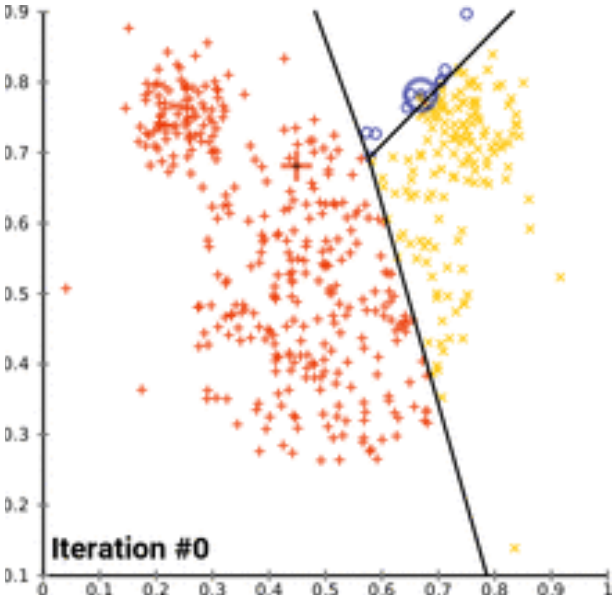
Anchit Gupta
MT19060

# K-Means Algorithm

*Clustering Algorithm*

## Introduction

K-Means algorithm is the method to partition the given dataset into k partitions. Each observation is present to nearest mean, that mean is also called centroid of the cluster. To calculate the distance to find the cluster for the beach observation the Euclidean Distance is used.




## Algorithm Used:

1. Select the **k** random observations from the dataset and name them as centroid.
2. Run this *n* time this step or until convergence is achieved till threshold
   2.1. Find the distance of every observation from the centroid
   2.2. Choose minimum distance and assign the cluster which has least distance.
   2.3. Calculate mean of cluster and assign that observation as the centroid of the cluster.

# Implementation

The k-means algorithm implemented using the as per the question, to cluster the instances in k clusters. So, I have assumed that no of clusters as 4(i.e. k=4). The n=20, i.e. number of iterations the Step 2 of the algorithm is performed.

The convergence is checked using the SSE i.e. the Sum of Squared Errors, for the

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} (p - m_i)^2$$

here $C_i$ is the i$^{th}$ Cluster, $m_i$ is the centroid or mean of the given Centroid $C_i$. The $p$ is the observation of the given cluster.

The convergence is shifting of the centroid of the cluster due to fact that the new point arriving in the cluster in every iteration till some time.

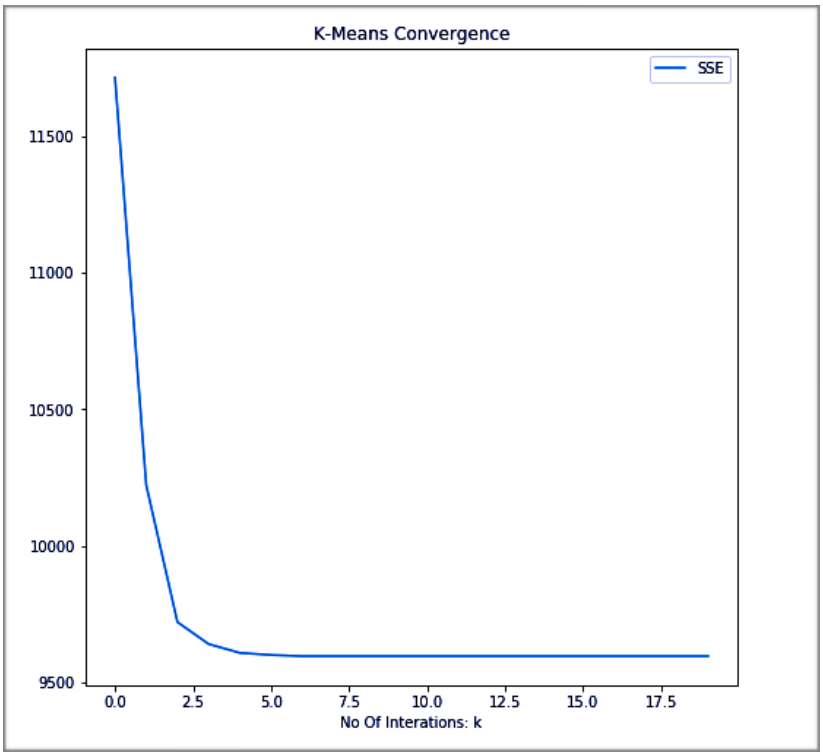



Fig:2 SSE of the k-Means Algorithm

# Precision, Recall and F1-Score

| | Predicted Class | | |
|---|---|---|---|
| Actual Class | | Class = Yes | Class = No |
| | Class = Yes | True Positive | False Negative |
| | Class = No | False Positive | True Negative |

Precision is the ration of the correctly predicted positive instances to the total positive instances in the given result.

$$Precision = TP/(TP + FP)$$

Recall is the ratio of the correctly predicted positive instances to all actual instances which belong to class *Yes*.

$$Recall = TP/(TP + FN)$$

F1-Score is the harmonic mean of the Precision and Recall of the given observation.

$$F1Score = 2/(Precision^{-1} + Recall^{-1})$$

Here all precision, recall and F1-score is higher the better.

Now, Come to the question which is to show the all three above in the implemented k-means algorithm.

# Question 2

In this the objective is to implement the algorithm k-means and find the precision, recall and F1-Score for normal configuration i.e. un-normalised data and using Euclidean Distance major.

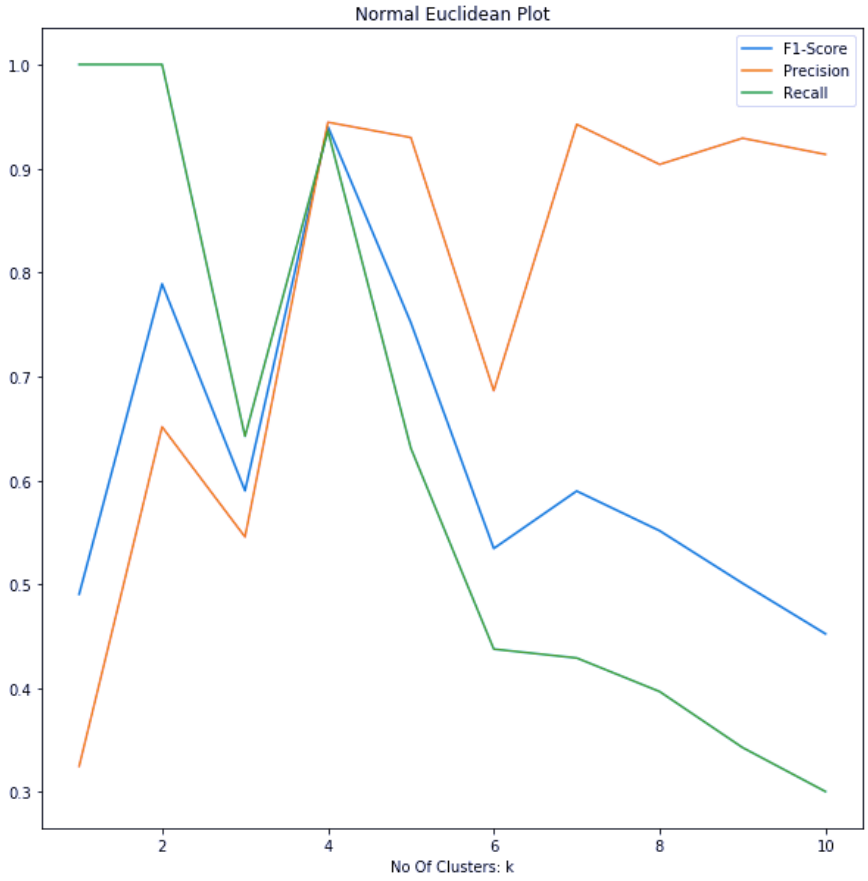



Fig:3 Plotting Precision, Recall and F1-score

# Question 3

In this the objective is to implement the algorithm k-means and find the precision, recall and F1-Score for normalised data having l2 normalisation for which I have used the Sklearn Preprocessing library and using Euclidean Distance major by varying the number of cluster in the algorithm from k=1 to k=10.

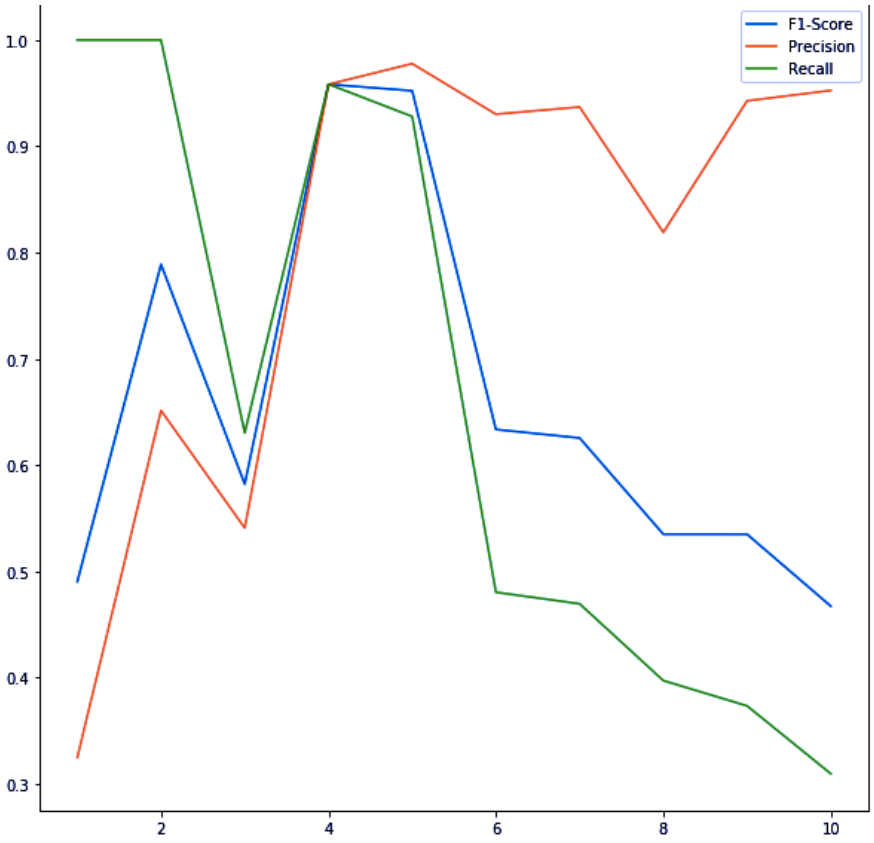



Fig 4. Plotting Precision, Recall and F1-Score of normalised data.

# Question 4

In this the objective is to implement the algorithm k-means and find the precision, recall and F1-Score for un-normalised data and using Manhattan Distance major by varying the number of cluster in the algorithm from k=1 to k=10.

$$ManhattanDistance = |x_2 - x_1| + |y_2 - y_1|$$

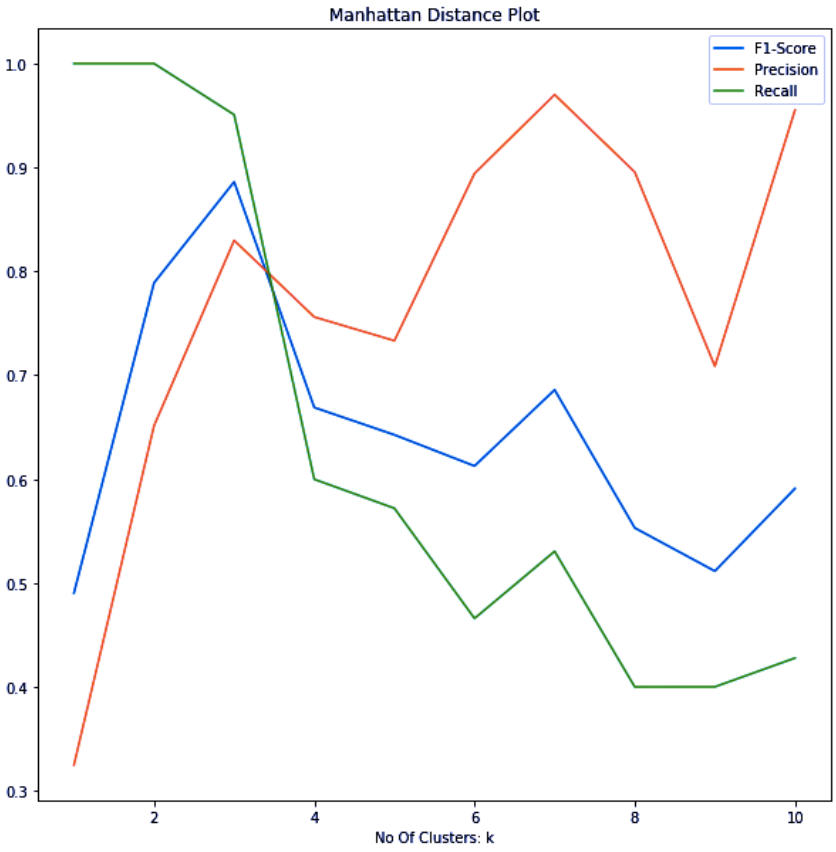



Fig 5. Plotting Precision, Recall and F1-Score using Manhattan Distance.

# Question 5

In this the objective is to implement the algorithm k-means and find the precision, recall and F1-Score for un-normalised data and using Cosine Similarity Distance major by varying the number of cluster in the algorithm from k=1 to k=10.

$$Cosine\ Distance = 1 - Cosine\ Similarity = 1 - < X . Y > /|X||Y|$$

where X and Y are observations.

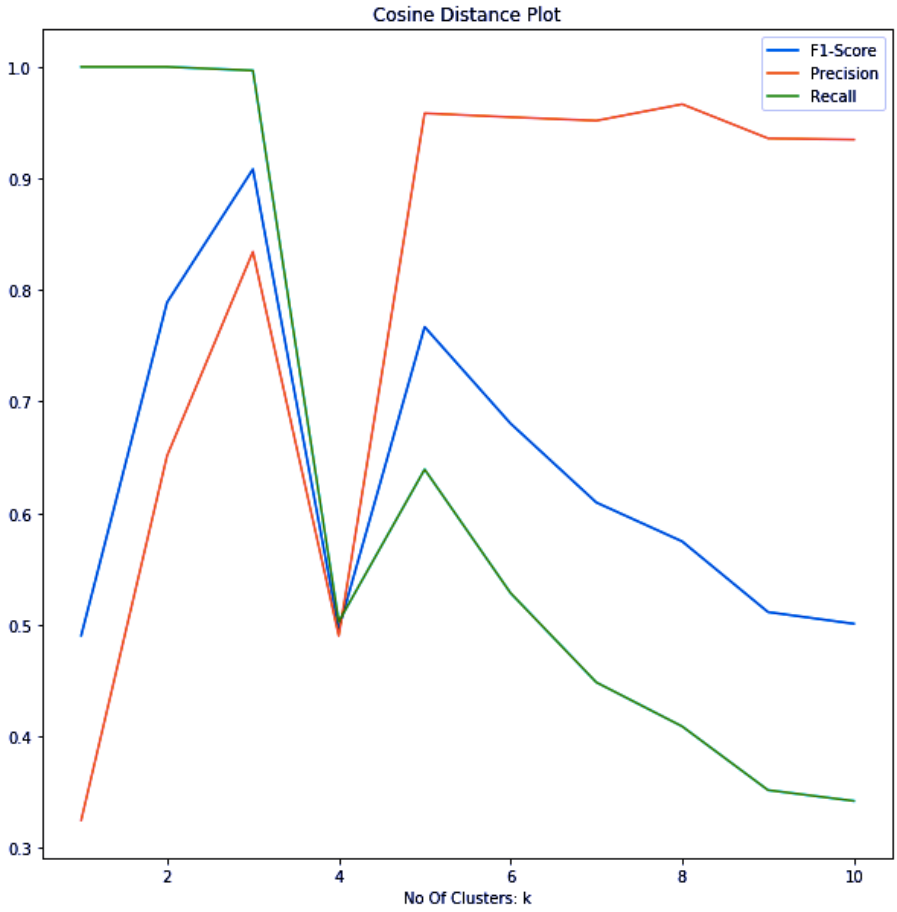



Fig 6. Plotting Precision, Recall and F1-Score using the Cosine Distance Major.

# Analysis

| | Best k | Precision achieved at k | Recall Achieved at k | F1-Score Achieved at k |
|---|---|---|---|---|
| **Original & Euclidean Distance** | 4 | 0.9443 | 0.9360 | 0.9402 |
| **Normalised & Euclidean Distance** | 4 | 0.9583 | 0.9584 | 0.9583 |
| **Original & Manhattan Distance** | 3 | 0.8297 | 0.9507 | 0.8861 |
| **Original & Cosine Similarity Distance** | 3 | 0.8342 | 0.9967 | 0.9083 |

As we already known the classes of the given observations but treating data for unsupervised learning by removing the classes from the observations. Given the dataset have 4 classes so it's obvious that the procedure should use such conditions which gave us highest score at k=4.

Here all the methods we encounter gave result near k but only 1st and 2nd method are best as they get quite accurately giving the best results are coming at the k which is same as of the original.

Also the fact that the precision is highest for the normalised data at k=4. But in case of 4th method our recall is abruptly showing the higher recall but it's precision is very low at that time. The 4th method is showing higher precision values on further larger cluster numbers which is quietly in correct as the recall is very low at that point which is near between 0.3 and 0.4 . The 3rd method is showing better recall but still showing the lower precision relatively with others. While manhattan is showing rise in precision at k=7 but still recall is very low lies between 0.6 and 0.7 which is relatively lower. So, here the final conclusion can be drawn is that euclidean distance is better than the other distance major for k-means considering giving better accuracy overall. But not able to comment on fact that whether data should be normalised or not as given data nature is uniform or not. Also, the results are quite near but normalised data is showing is showing slightly better results than the un-normalised data with euclidean distance.

# References:


- Stackoverflow: Precision and Recall
- Stanford Precision and Recall
- Wikipedia: K-means Fig: 1