CSE 575: Statistical Machine Learning
Project 2: Unsupervised Learning – K-Means

Raktim Mukhopadhyay
ASU ID – 1217167380

Overview

K-means clustering is a type of unsupervised learning, which is used when we have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

Objectives

1. Implement a strategy (STRATEGY – 1) in which the initial centroids are picked up randomly from the given dataset
2. Implement a strategy (STRATEGY – 2) in which the first centroid is picked up randomly, for the i^{th} centroid (i>1) a data sample is chosen among all possible data samples such that the mean distance of this chosen sample to all previous (i-1) centers is maximum
3. Implement STRATEGY -1 and STRATEGY -2 for K=2 to 10
4. Calculate the objective function $\sum_{i=1}^{k}\sum_{x\in \mathrm{D}}\left|\left|x-\mu\right|\right|^{2}$ for K= 2 to K=10.
5. Plot the values of objective function (WCSS) vs number of clusters (K)

Dataset Description

The dataset provided for this project is an unlabeled dataset having two columns and three hundred rows. Considering the first column as the x-coordinate and the second column as the y-coordinate we get a plot shown below. The plot helps in understanding the data visually and will help a lot during identifying clusters.

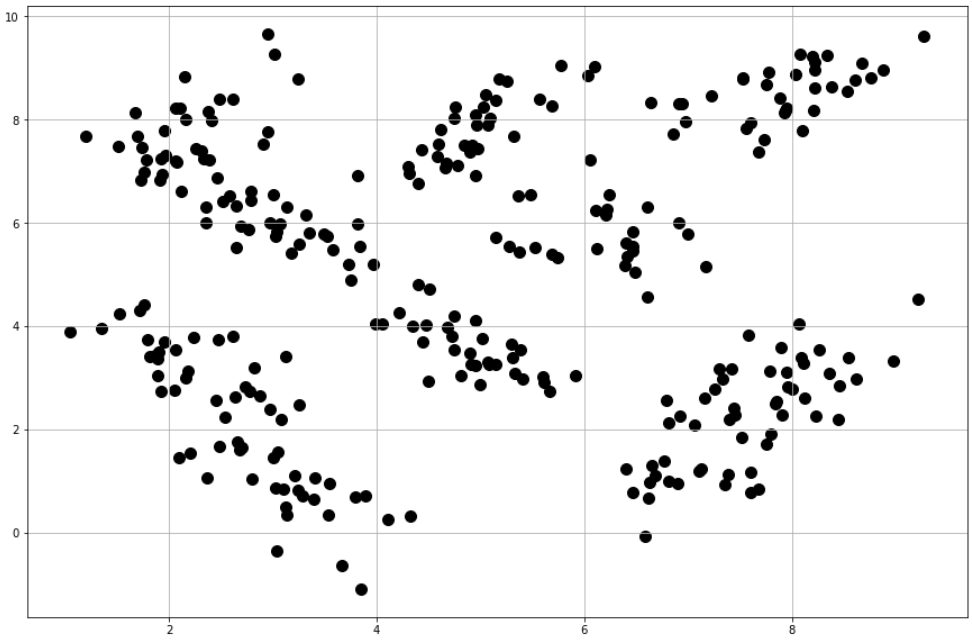


Fig: Plot of dataset

- **Strategy – 1**

We have been asked to plot the objective function vs K plot twice with different initializations.
In this strategy the centroids are randomly initialized for K=2 to K=10

First Run:

K	2	3	4	5	6	7	8	9	10
Val Obj. Fun	1921.0	1294.2	788.96	788.96	463.21	404.88	349.88	241.43	224.97

Plot of Objective Function vs Number of Clusters:

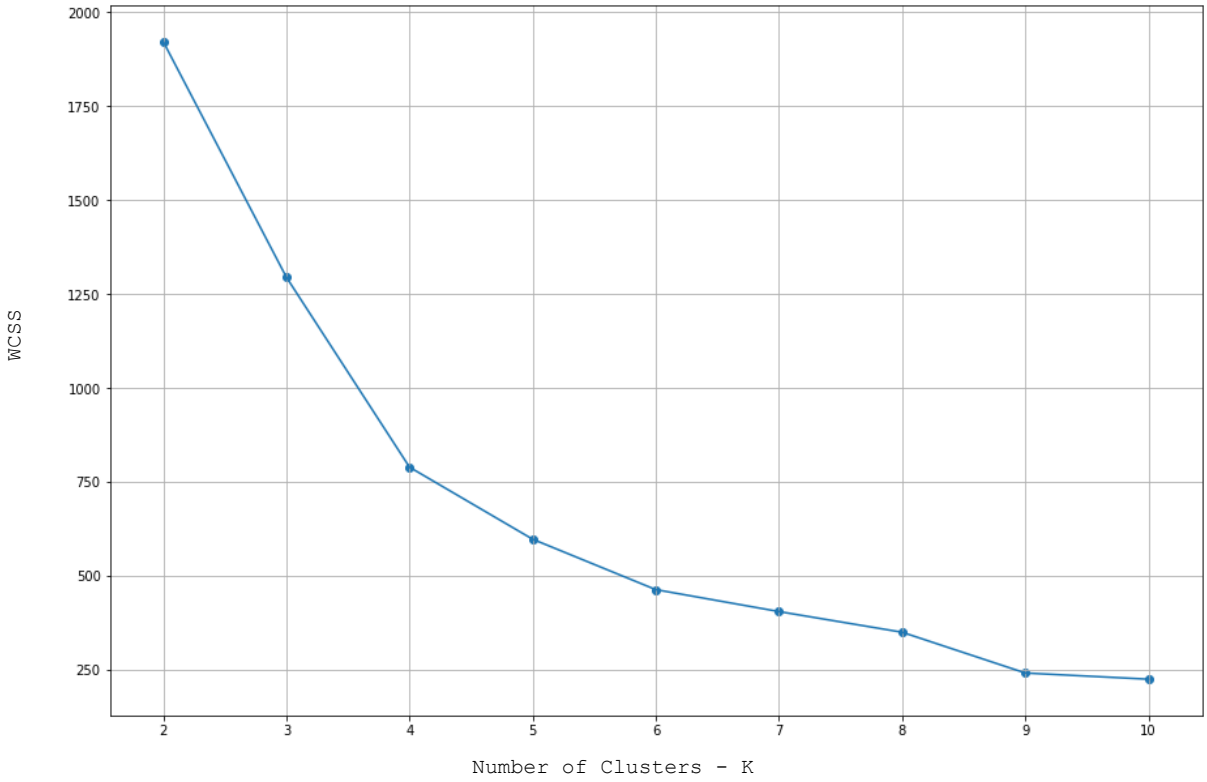
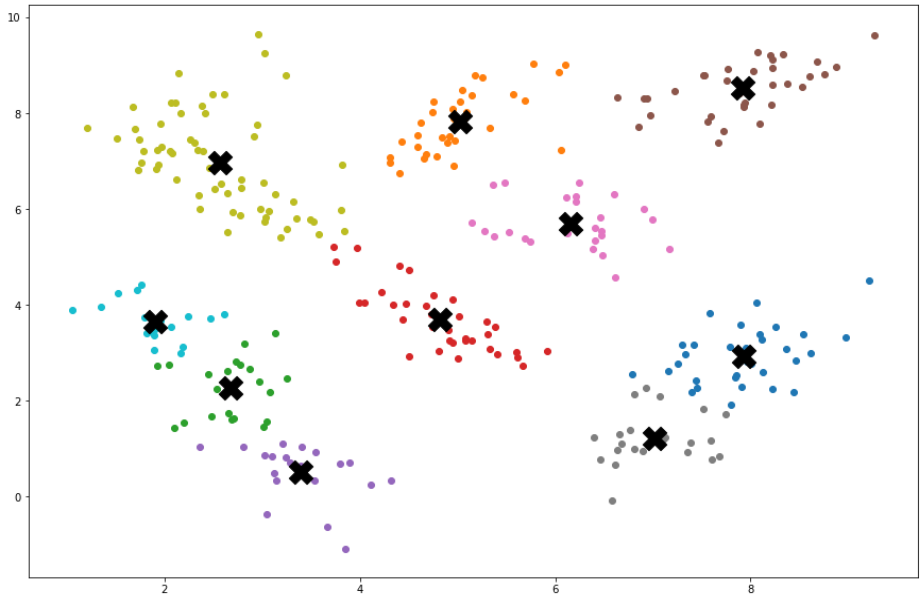


Fig: Objective function vs K

Discussion on the Plot:
We can see that value of WCSS decreases rapidly from K=2 to K=4. After that the value of WCSS decreases slowly from K=4 to K=6. From K=6 to K=9 the decrease in objective function further slows down. From K=9 to K=10 there is a slight decrease in the objective function.

Cluster Plot of K=10:

Just plotting the cluster plot for K=10 to visualize the clustering done by **Strategy – 1 Run - 1**



Second Run:

K	2	3	4	5	6	7	8	9	10
Val Obj. Fun	1921.0	1293.7	805.1	592.9	476.1	391.2	290.9	301.0	219.9

Plot of Objective Function vs Number of Clusters:

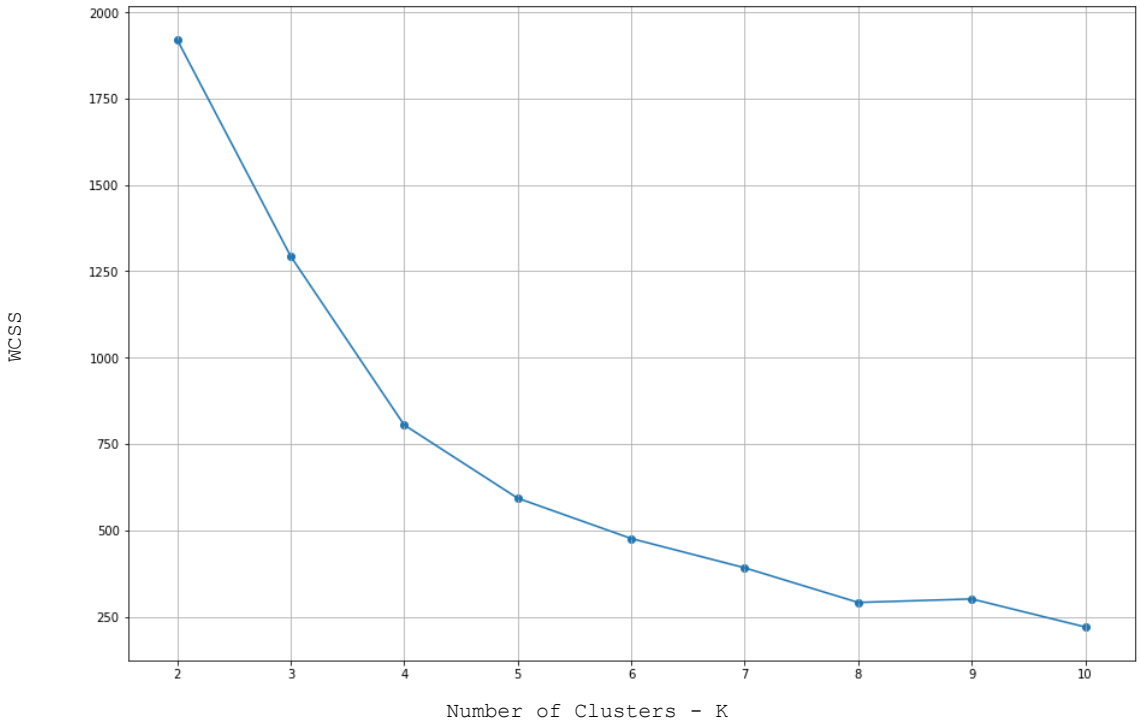
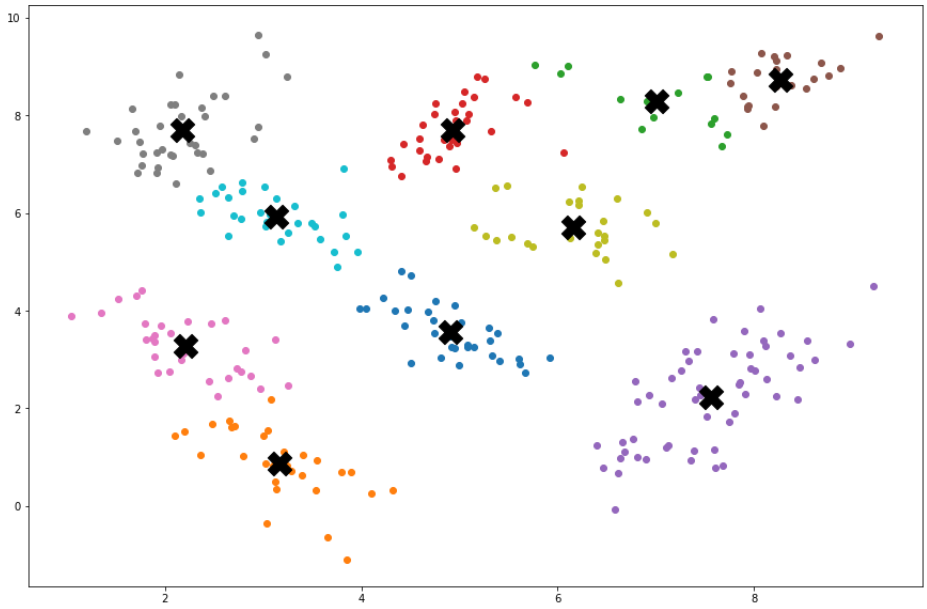


Fig: Objective function vs K

Discussion on the Plot:
We can see that value of WCSS decreases rapidly from K=2 to K=4. After that the value of WCSS decreases slowly from K=4 to K=6. From K=6 to K=8 the rate of decrease in objective function further slows down. From K=8 to K=9 the value of WCSS doesn't change much and hence that segment is nearly a straight line. From K=9 to K=10, there is a small decrease.

Cluster Plot of K=10:
Just plotting the cluster plot for K=10 to visualize the clustering done by **Strategy – 1 Run - 2**



We can see that as compared to Strategy-1 Run 1 plot the clusters in this plot are different but that does not influence the WCSS in a very aggressive manner.
Also, the different clustering can be attributed to the fact that different initializations are used in two different runs.

- **Strategy – 2**

In this strategy the first centroid is picked up randomly, for the ith centroid (i>1) a data sample is chosen among all possible data samples such that the mean distance of this chosen sample to all previous (i-1) centers is maximum

First Run –

K	2	3	4	5	6	7	8	9	10
Val Obj. Fun	1921.0	1294.2	805.1	613.2	463.2	367.6	383.2	240.3	223.0

Plot of Objective Function vs Number of Clusters:

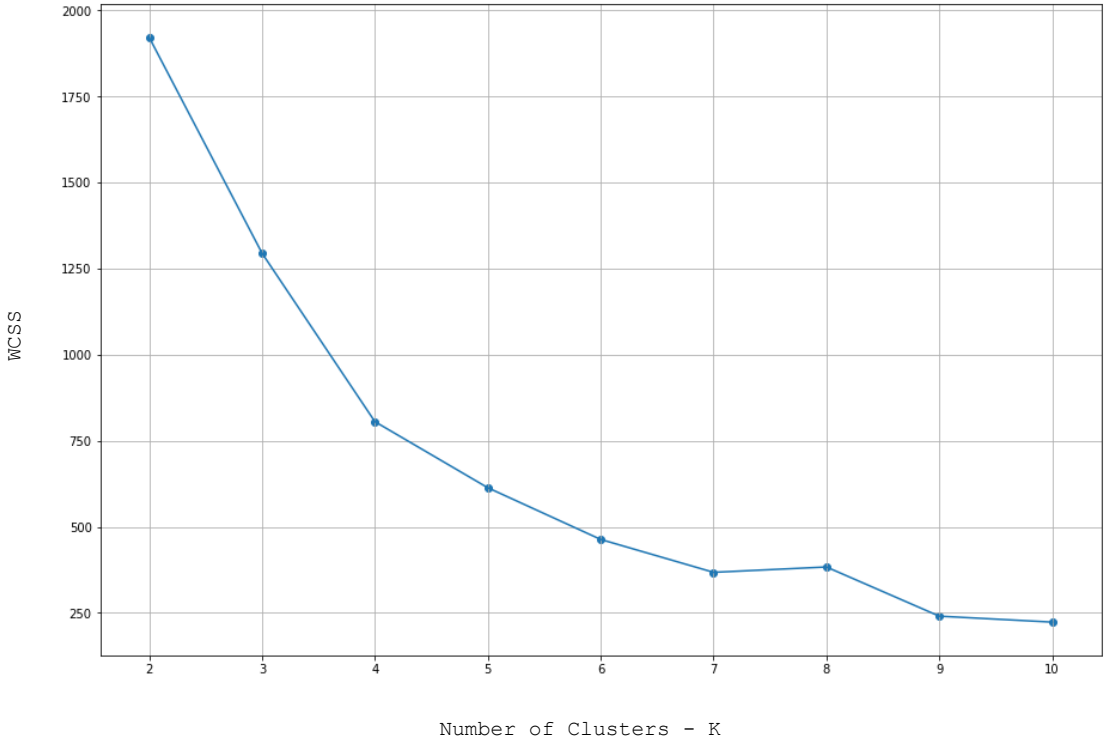
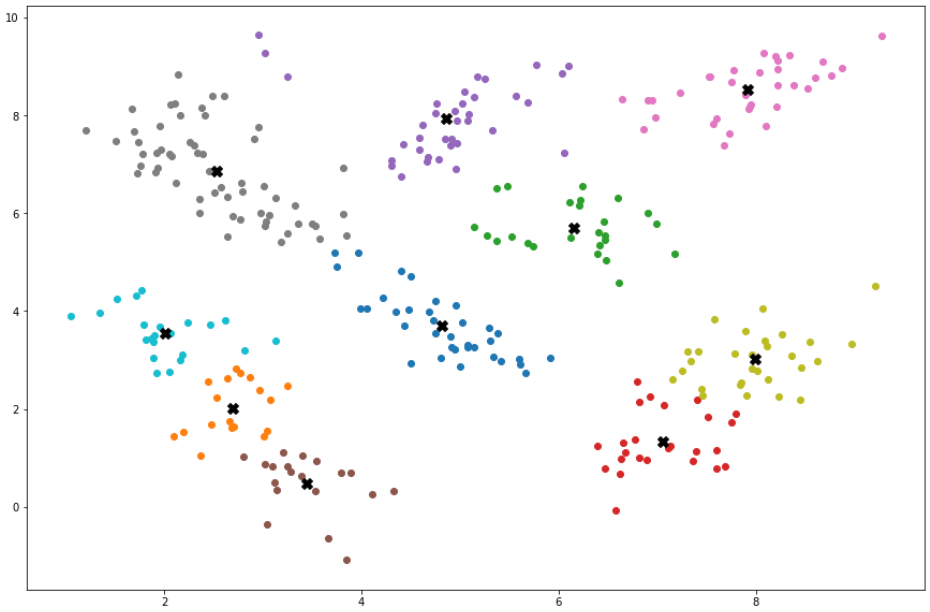


Fig: Objective function vs K

Discussion on the Plot:
We can see that value of WCSS decreases rapidly from K=2 to K=4. After that the value of WCSS decreases slowly from K=4 to K=7. From K=7 to K=8 the value of WCSS doesn't change much and hence that segment is nearly a straight line. From K=8 to K=9, there is a small decrease. From K=9 to K=10, the value of the objective function remains nearly unchanged.

Cluster Plot of K=10:

Just plotting the cluster plot for K=10 to visualize the clustering done by **Strategy – 2 Run - 1**



Second Run –

K	2	3	4	5	6	7	8	9	10
Val Obj. Fun	1921.0	1293.7	805.1	592.0	476.1	368.9	289.8	258.5	228.2

Plot of Objective Function vs Number of Clusters:

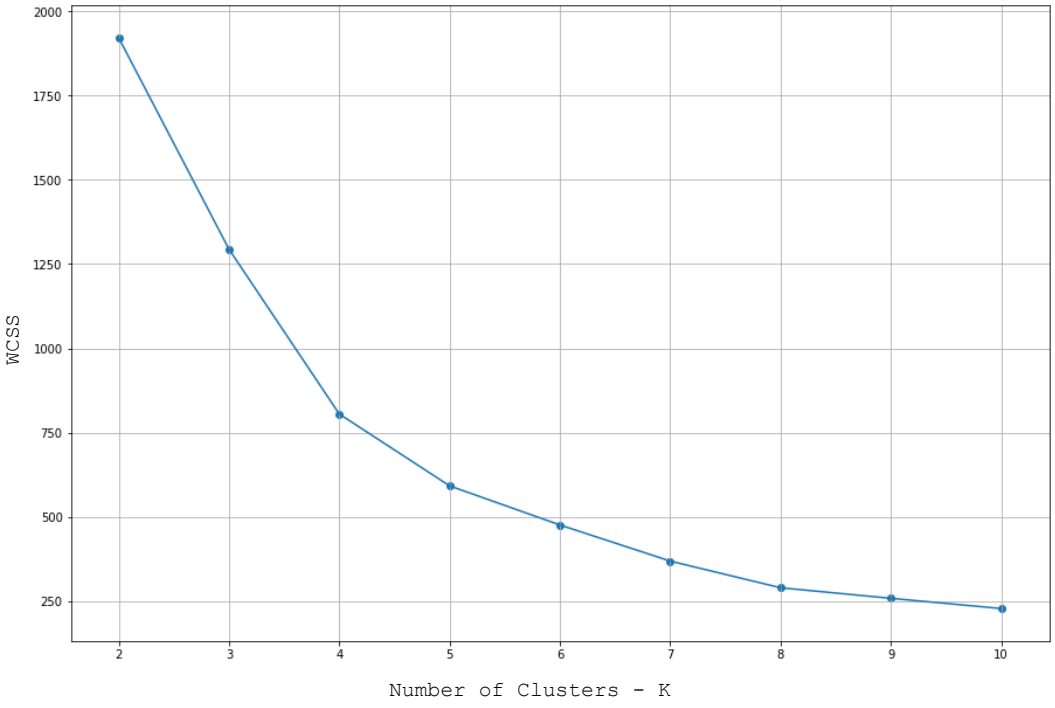
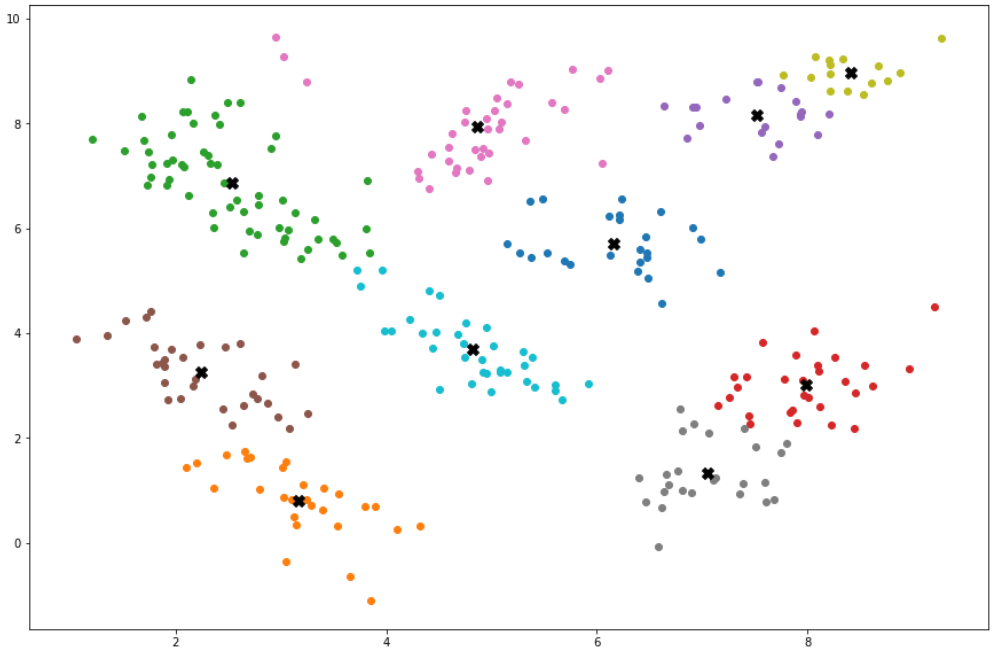


Fig: Objective function vs K

Discussion on the Plot:
We can see that value of WCSS decreases rapidly from K=2 to K=4. After that the value of WCSS decreases slowly from K=4 to K=7. From K=7 to K=8 the change in value of WCSS further slows down. From K=8 to K=9, there is a small decrease. From K=9 to K=10, the value of the objective function remains nearly unchanged.

Cluster Plot of K=10:
Just plotting the cluster plot for K=10 to visualize the clustering done by **Strategy – 2 Run - 2**



We can see that as compared to Strategy-2 Run 1 plot the clusters in this plot are different. Also as mentioned previously, the different clustering can be attributed to the fact that different initializations are used in two different runs.

Summary

Both the strategies were run many times intentionally to obtain the two types of clustering for K=10 and a somewhat different Elbow Plot shown in this project.

The optimal number of clusters appear to be 4 from the elbow plots for both Strategy-1 and Strategy-2.

The Strategy-1 is the random initialization approach and the Strategy-2 is the Farthest Point Approach. Running both the strategies a number of times give slightly different Elbow Plot each time since the centroid initialization is random in both the cases (Only First centroid is random in Strategy -2, but since the next centroid is dependent on the first centroid hence it can be called that in strategy-2 some kind of randomness is present). A robust and generalizable K-Means algorithm will need to choose the initial centroids such that WCSS is minimized. This is when K++ Algorithm is needed.