# Project Part 2 Report

**Abstract:**

**In this part of the project, we have implemented K means clustering using 2 different strategies and analyzed the objective function for both the strategies for k values varying from 2-10. We have used a dataset containing 2-D data points.**

**Strategy1:**

**In this method we have started with k=2 and initialized k random number of centroids from our data set, then we have calculated the distance of each datapoint in the dataset to each of the centroids. After calculating the distance, we consider the centroid with minimum distance and assign this datapoint to that cluster. In this manner we calculate the distance and assign data points to each cluster. We calculate the mean of each cluster and now reassign the centroid value to this mean and follow the same steps. We calibrate the centroids until we reach a point where the centroids remain the same in consecutive iterations. In the same way we calculate the centroids and divide the points into clusters for k=2 to 10. We follow the same steps twice and calculate the objective function for each k value.**

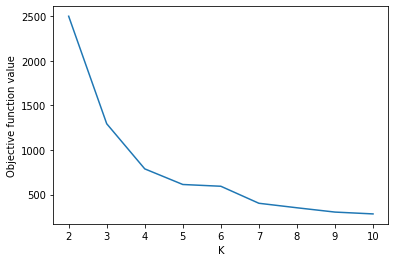
**Objective function values & plot:**

We have plotted the k value on X-axis and the corresponding Objective function values on Y-axis in a graph.

**Strategy1 1st iteration:**

k = [(2, 2498.1135603167704), (3, 1293.7774523911341), (4, 788.2693490065556), (5, 613.2824392056037), (6, 593.5140687553135), (7, 402.91835862050755), (8, 352.4503318125102), (9, 304.2433444803494), (10, 283.6598409869368)]

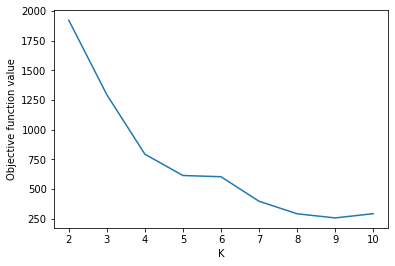
**Plot:**

****

**Strategy1 2nd iteration:**

k = [(2, 1921.0334858562057), (3, 1293.7774523911348), (4, 792.7110095863355), (5, 613.282439205604), (6, 602.912202768955), (7, 396.45651401453733), (8, 290.8573092438312), (9, 255.96788691784874), (10, 291.6051740736677)]

**Plot:**

****

**Strategy2:**

**In this strategy we choose the initial centroid and compute the other centroids. We calculate the distance of each point from all the centroids and sum up all the distances. We select the point with maximum distance and update the centroid. Now we calculate the Euclidean distance from this centroid and update the clusters. We follow the same strategy until we have the same centroids in consecutive iterations. We follow the same steps for k in range of 2 to 10. We apply this strategy twice. We calculate the objective function for each k value. We follow the same method to calculate the objective function for both strategy1 and strategy2.**

**Objective function:**

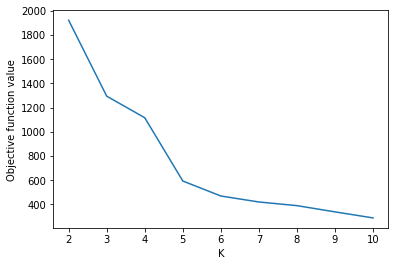
**Je= 2**

**mi= 1/ni**

**Objective function values & Plots :**

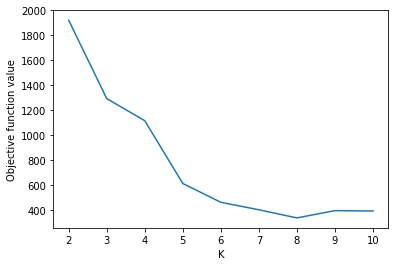
**1st iteration:**

k = [(2, 1921.0334858562055), (3, 1293.7774523911357), (4, 1115.5344812362407), (5, 592.8779292654727), (6, 468.9538755370984), (7, 419.57835150204346), (8, 389.34746617094305), (9, 337.8479763907384), (10, 287.9208137368957)]

****

**2nd iteration :**

k = [(2, 1921.0334858562055), (3, 1293.7774523911346), (4, 1116.3315012668595), (5, 613.2824392056036), (6, 462.9263558248376), (7, 402.9183586205073), (8, 337.80376816338804), (9, 396.22169495705356), (10, 393.1005164982468)]

****

**Conclusion:**

In this part of the project, we have applied both strategies twice and we can clearly observe that in strategy 1 we see a huge drop in the value of loss function from k=2 to k=4 and from there on the loss function reduced significantly for increasing k value. In strategy 2 we can observe that there is major difference in the values of loss function from k=2 to k=3 from there on we can see a significant change for each k value. In both the strategies after k=8 there isn’t much difference in the objective function value.