**A. The dataset that you are using, and the source for it. Dimensions of the dataset, and the specific dimensions that you plan to use for unsupervised learning.**

**Ans:** Dataset that is used is Flower dataset.csv. In this data set we have four data points or dimensions: x1 as sepal length, x2 as sepal width, x3 as petal length, x4 as petal width and label y as flower name. As the question demands two dimension we are considering x1 and x2 our data points and ignoring other data points. We also ignore label as we are required to work on unsupervised learning. Data points x1 and x2 are continuous as length and width are continuous variable (There is always another possible value for length and between any two values.).

**B. Parameters for K-means, and results in visual form.**

**Ans:** We are using our dimension x1 and x2 for k-mean working. The appropriate value of k was found by elbow method and by this value of k we plot and visualize clusters.

**C. Name of the density based algorithm that you've chosen. And parameters for density based clustering, and results in visual form**

**Ans:** I have chosen dbscan (density-based spatial clustering of applications with noise) algorithm. Dbscan take two parameters. Epsilon(**E**) and Minimum Points(**M**). E determines a chosen radius such that if it includes enough points within it. M determines the minimum number of data points required in a neighborhood to be defined as a cluster. The main idea behind DBSCAN is that a point belongs to a cluster if it is close to many points from that cluster.

**D. Comparison of b and c (Above steps, max write 5 lines) and cover why two algorithms performed in a different way. Which one you'll prefer for your specific dataset and why.**

**Ans:** K-means clustering is sensitive to the number of clusters specified. Number of clusters need not be specified in dbscan. K-means Clustering is more efficient for large datasets. Dbscan Clustering cannot efficiently handle high dimensional datasets. K-means Clustering does not work well with outliers and noisy datasets. DBScan clustering efficiently handles outliers and noisy dataset. According to the data set I chose, I will prefer dbscan as in k-mean centroids can get dragged by the outliers resulting in clusters but dbscan can detect outliar and only make cluster of points that are close to each other. In our dataset outlier is not needed to be a part of cluster.