Data Mining (CS524)
Programming Assignment 2

Problem - 1
I this problem, we had to implement both DB-Scan and K-Means algorithm on Iris and Spiral Dataset.

1. K-Means Algorithm - To implement this algorithm I used a list of lists in python for saving both the centroids and the label of the datapoint. To find the best K value for both the datasets I plotted the SSE vs K graph for both the dataset (Given in figure-1, and figure-2).

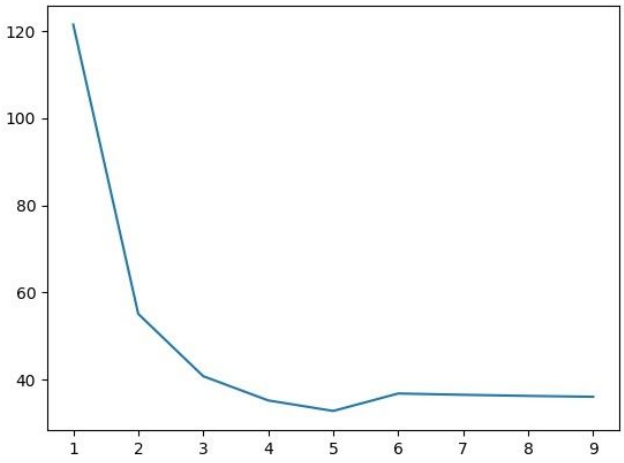

Figure - 1 Graph for SSE vs K for the Iris dataset.

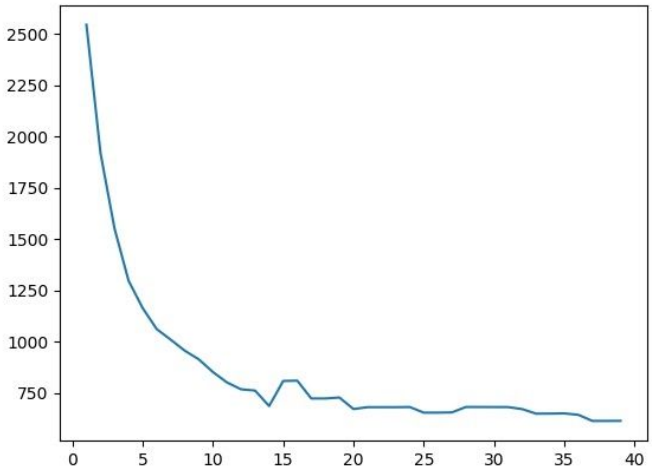

Figure - 2 Graphs for SSE vs K for the 2D Spiral dataset.

Iris Dataset - We can see that after K=3 the values for SSE loss are almost constant. Therefore we can make exactly 3 clusters out of the Iris Dataset. (So for Iris Dataset we choose the k value as 3 for our further experiments).

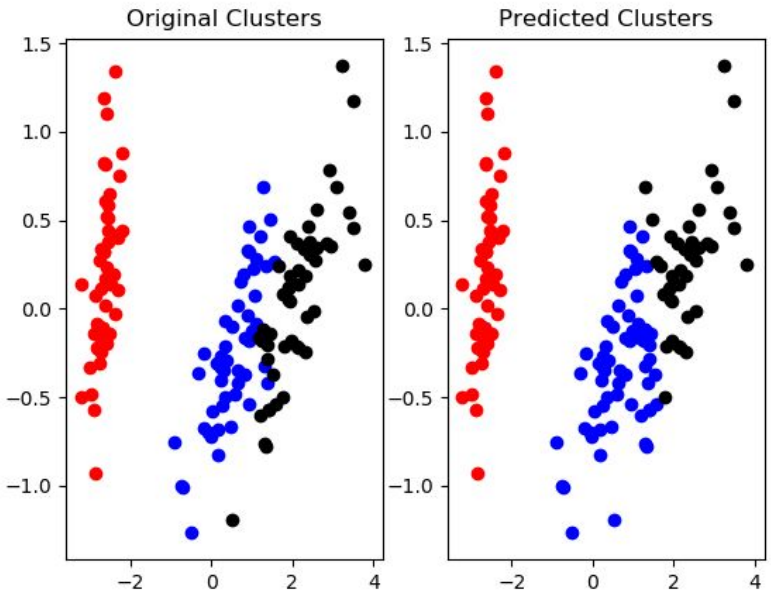

Figure-3 For Iris Dataset we transform the points into 2 dimensions using PCA. We use K-Means with K=3 for finding the cluster. The left image represents the original clusters and the right image represents the predicted clusters.

2D Spiral Dataset - Here the value continuously decreases for SSE up to K = 10. So we pick K = 10 as the number of clusters.

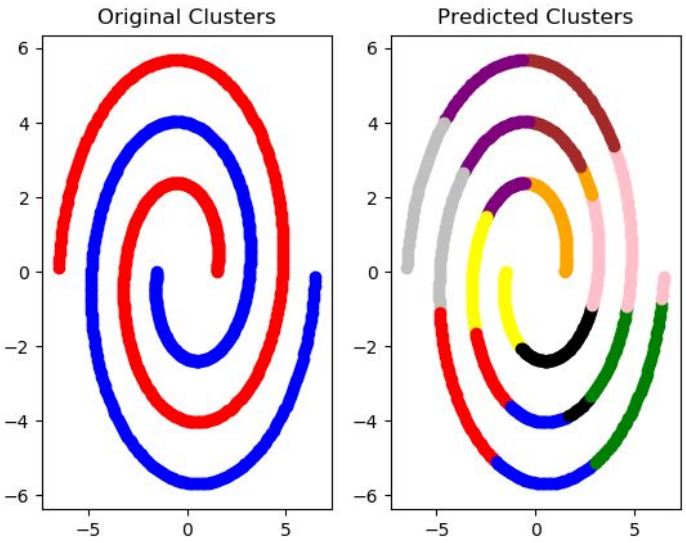

Figure-4 For the 2D spiral dataset we find the clusters using k=10 with the k-means algorithm. It clearly shows clustering done here is not very good.

In Figure-3 and Figure-4 we represent the clustering using the K-means algorithm for the Iris dataset and 2D spiral dataset respectively.

Observations -

a) Clustering was done for the iris dataset is very good and only a small portion of points are misclassified. The SSE loss for the iris dataset is also very low (approx. 40).
b) Clustering for the 2D spiral dataset is very bad. The algorithm is not able to understand the data in this dimension. If we can represent the data in a higher dimension with some redundancies maybe it would be able to cluster the dataset. The SSE loss for this clustering is very high despite the high number of clusters (approx. 680).
c) As K-means clustering uses distance metric for making clusters it is not able to cluster the local portions of the dataset.

2. DB-Scan Algorithm - In this algorithm, we use a dictionary (hashing data structure) to optimize the algorithm. In a dictionary, we can find any element in 0(1) time. So for each core point, we insert all the neighboring points in the dictionary, and to update the label of the data points we used this dictionary.

Iris Dataset - For the iris dataset, we found that at h=0.2 and min_points=5 gets the best accuracy but using this combination it does not produces 3 separate clusters. Instead, it classifies the elements of the 3rd clusters as noise points.

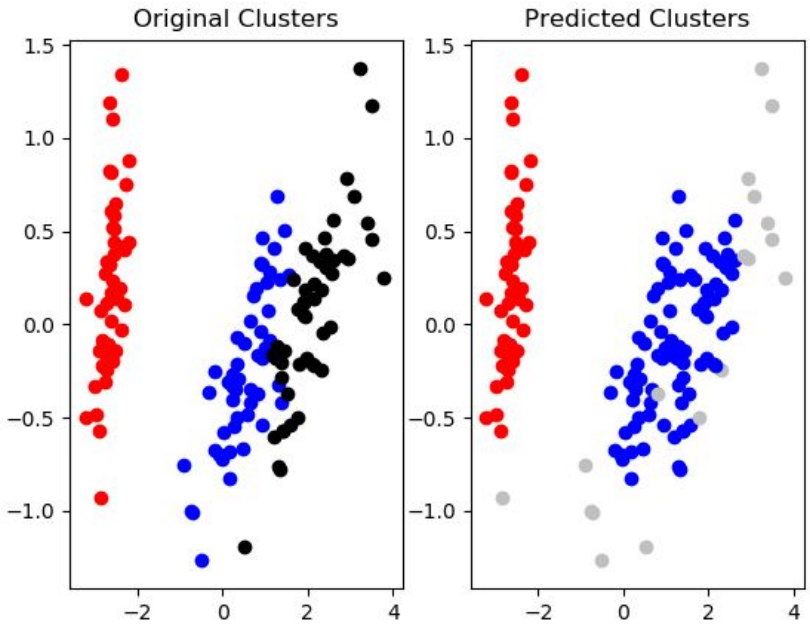

Figure-5, For the DB-Scan algorithm, we find the clusters for the iris dataset. The left side represents the original clusters and the right side represents the predicted clusters.

2D Spiral Dataset - This dataset is correctly identified by the DB-Scan algorithm. In this case, as the DB-Scan algorithm is finding the local clusters it is able to easily find any shape of the cluster in the dataset.

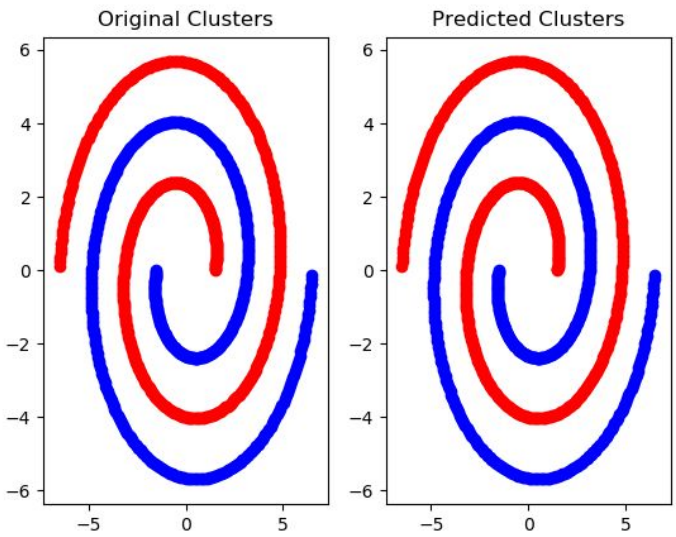

Figure-6 For the DB-Scan algorithm, we find the clusters for the 2D Spiral dataset. The left side represents the original clusters and the right side represents the predicted clusters.

In Figure-5 and Figure-6 we represent the cluster formed by the DB-Scan algorithm for the Iris dataset and the 2D spiral dataset respectively.

Observations -

1. It is able to correctly represent the 2D spiral dataset, with an accuracy of 100%. This is because the points are very near to each other and hence it is able to easily all such points as the core points and then make a label for them.
2. This algorithm works well for the Iris dataset but it is not able to find the third cluster because the points in the 3rd cluster are far away from each other and this is not the case for the 1st and 2nd cluster. Therefore it is not able to find good accuracy for this dataset.
3. If we have a very dense dataset then the DB-Scan algorithm can work really well for this dataset as it can make different core points for different clusters.

Comparison -

1. Time Complexity -
   K-means - a) Iris Dataset - 0.07827
   b) 2D spiral Dataset - 1.8735
   DB-Scan - a) Iris Dataset - 0.1627
   b) 2D spiral Dataset - 6.9411

   Therefore the time complexity for the k-means algorithm is much less than the DB-Scan algorithm. For both the datasets, the time taken by the k-means algorithm is less than the DB-Scan algorithm
2. Accuracy -
   K-means - a) Iris Dataset - 88.97%
   b) 2D spiral dataset - 50% (With k=2)
   DB-Scan - a) Iris Dataset - 72%
   b) 2D spiral Dataset - 100%

   Accuracy for different datasets differs for both the datasets. For the iris dataset, it is better to use k-means whereas k-means perform really bad on the 2D spiral dataset. For the 2D spiral dataset, we can prefer the DB-Scan algorithm as it is able to correctly cluster all the points. For the iris dataset, the DB-Scan algorithm is not able to perform better because the density for the 3rd cluster is very less and therefore it is not able to recognize the 3rd cluster.

Finally, we should use a different algorithm for different datasets. If the density of the dataset is high we can prefer the DB-Scan algorithm. Whereas if we have a dataset that can be separated by a distance metric then we can use the k-means algorithm.

Problem-2
In this problem, we implemented both EM and Denclue algorithms for finding the PDF of the datasets.

EM Algorithm - We found the means, sigma, and prior probabilities using k = 3 for the iris dataset and we use k=2 for the 2D spiral dataset. Further, we find the density estimate using these means, sigma, and prior probabilities.

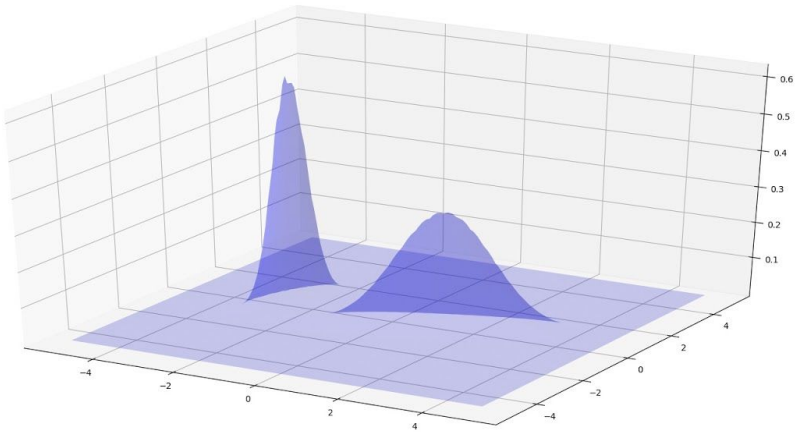

Figure-7 Density estimate for Iris Dataset using Expected maximization algorithm.

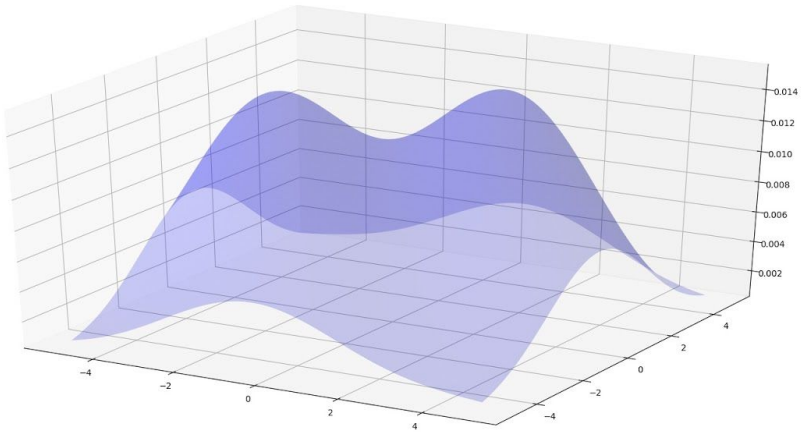

Figure-8 Density estimate for 2D Spiral Dataset using Expected maximization algorithm.

Denclue - We use the Gaussian kernel for determining the density estimate of the dataset. We also implement the denclue algorithm for finding the clusters but we do not use it for determining the density estimate.

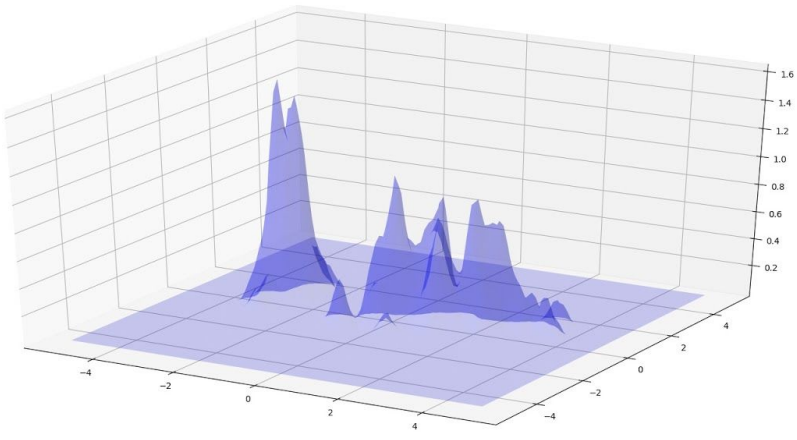

Figure-9 Density estimate for Iris Dataset using the Denclue algorithm.

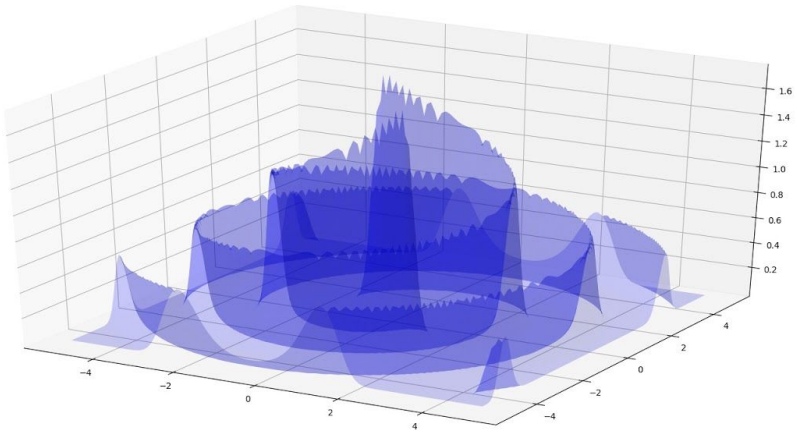

Figure-10 Density estimate for 2D spiral Dataset using the Denclue algorithm.

In Figure-7, Figure-8, Figure-9, and Figure-10 we represent the 3D plots density estimates of different algorithms and different datasets.

We observe that the EM algorithm produces a much smoother density estimation than the Denclue algorithm. So if the dataset can be classified accurately using the EM algorithm then we should use the EM algorithm because it will give a much smoother density estimate. For the 2D spiral dataset, the EM algorithm produces a very poor result and does not accurately represent the density function whereas the Denclue algorithm produces much better results.

So, for the iris dataset, we should use the EM algorithm, and for the 2D spiral dataset, we should use the denclue algorithm.