LAB ASSIGNMENT – VII
Clustering

Student's Name: Gajraj Singh Chouhan **Mobile No: +91-9351159849**

Roll Number: B19130 **Branch: DSE**

1 a.

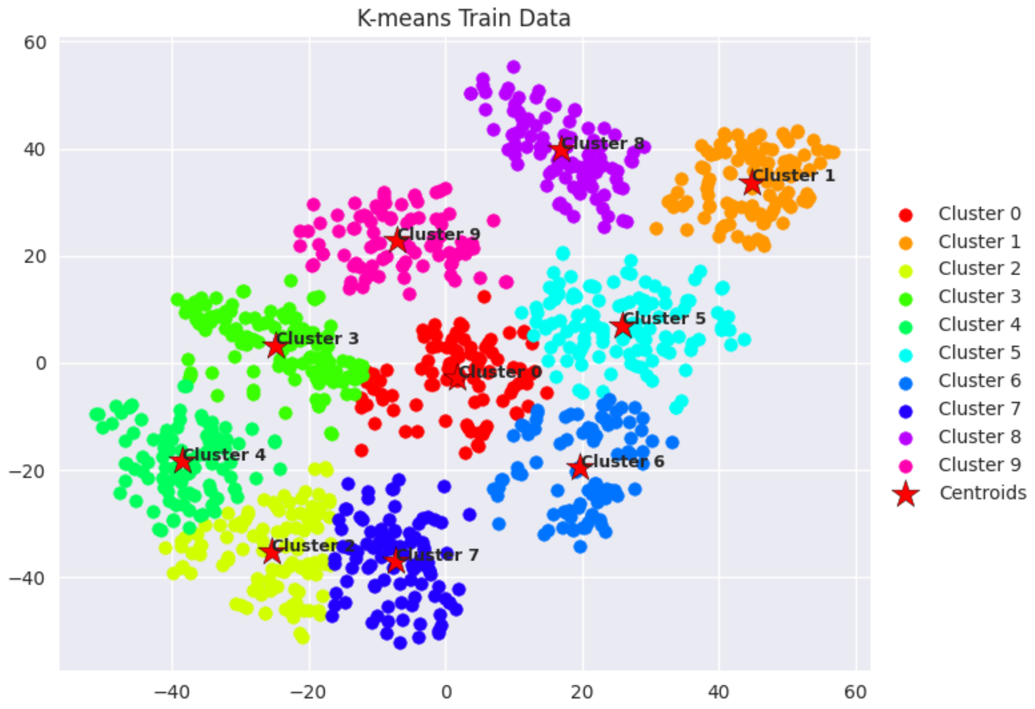


Figure 1 K-means (K=10) clustering on the training data

Inferences:

1. Inferring from the clusters, the K-Means algorithm must have high accuracy (60-70%) as it has identified the cluster correctly. The boundary between them looks distinct and clear although the cluster at the bottom and centre have less clear boundaries (Cluster 3, 4).
2. Some of the shapes are circular but not all, there is also a linear boundary between the clusters at the bottom left.

b. The purity score after training examples are assigned to the clusters is **0.701**

c.

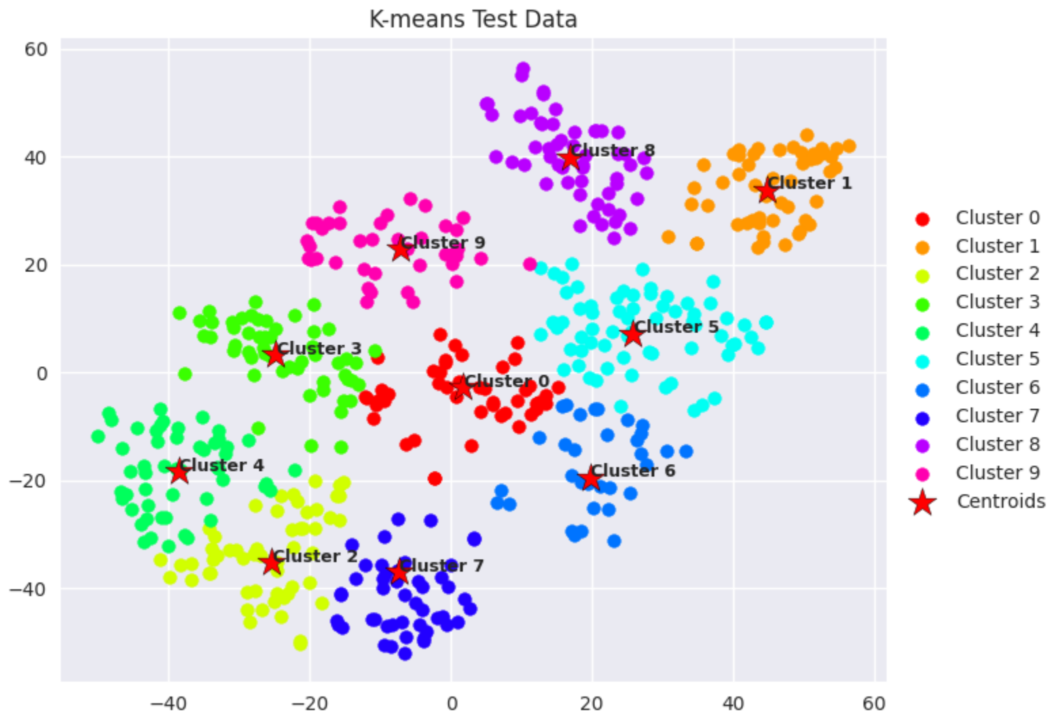


Figure 2 K-means (K=10) clustering on the test data

Inferences:

1. The train examples have fewer points hence lower density, the shapes of clusters have been the same. The gaps between the cluster can also be seen and the concentration of data at the cluster centre has decreased.

d. The purity score after test examples are assigned to the clusters is **0.686**

Inferences:

1. Purity score of training data is higher because we have fitted the training data on the model and it will be more similar when we are predicting the training data instead of testing data hence the higher purity score. However, the difference is very low.
2. The limitations of KMeans include clustering of outliers as they can change the position of centroids so we should consider removing outliers, we also have to reduce the data from higher dimensions. We also have to specify the “K” value for the algorithm to work. It also fails when we have multiple classes at the same spherical distance then it can’t distinguish those classes.

2 a.

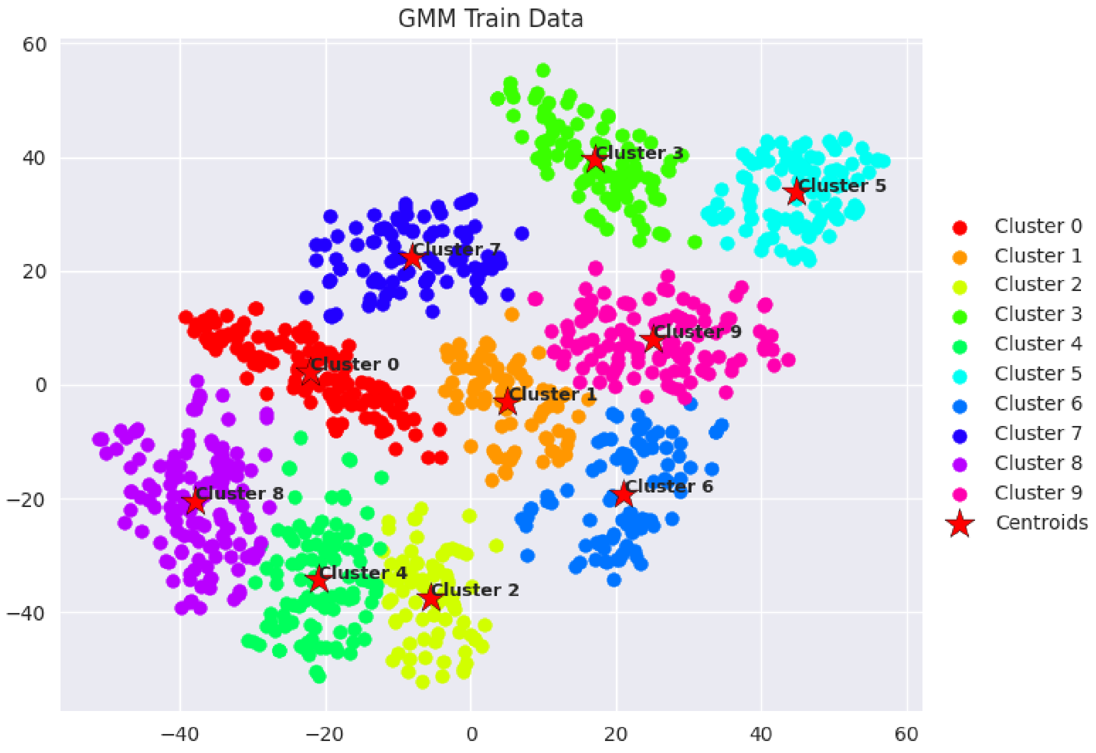


Figure 3 GMM clustering on the training data

Inferences:

1. Inferring from the clusters, the algorithm is accurate (60-70%) as it has identified them pretty much correctly. GMM assumes that data is based on the Gaussian distribution and elliptical shape of GMM may not fit well for all of them.
2. Yes, the boundary of clusters is elliptical but it isn’t perfectly elliptical in certain cases.
3. GMM has more non-linear boundary instead of linear like KMeans, and shape is more elliptical. This can be seen in the lower half of the graph where the non-linear boundary fits well than the linear one in case of KMeans.

b. The purity score after training examples are assigned to the clusters is **0.719**

c.

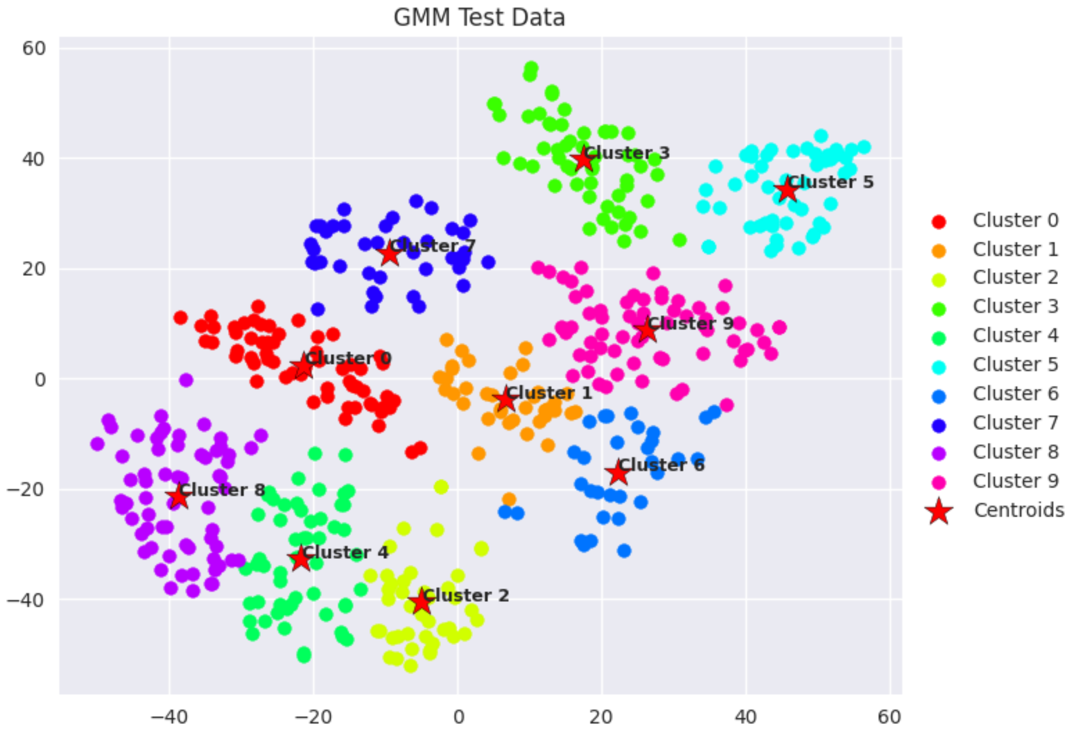


Figure 4 GMM clustering on the test data

Inferences:

1. The test data points are more away from each other so they have less density. They also appear more distinct. The shape of the clusters was resembling elliptical shape to a greater extent for the training data.

d. The purity score after test examples are assigned to the clusters is **0.704**

Inferences:

1. The training and the test purity scores are almost the same (training is slightly higher) and there is a negligible difference between them. This means that the model has fitted both of training and testing well.
2. GMM has certain limitations. We have to specify the number of clusters before applying the model. It makes assumptions about the Gaussian distribution of the data. It is computationally intensive when we are dealing with large dimension data.

3 a.

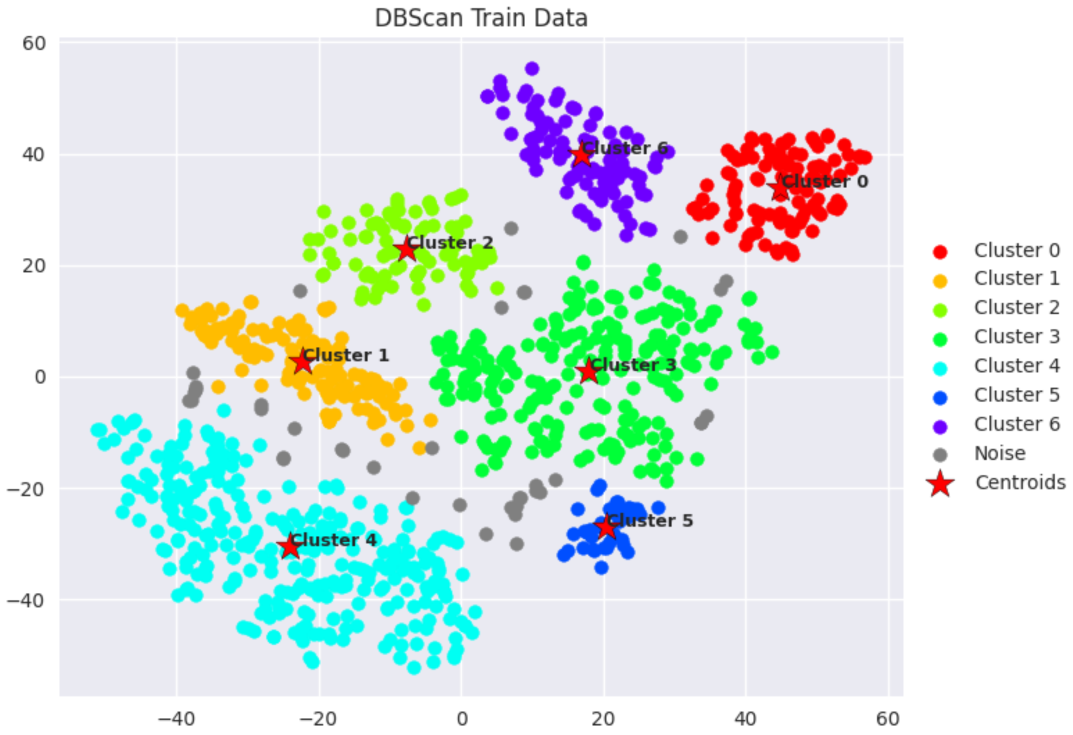


Figure 5 DBSCAN clustering on the training data

Inferences:

1. The number of clusters according to DBSCAN is 7 whereas our data consisted of 10 total clusters. This is because some clusters were located in a dense area with no low-density area separating them. When the clusters are quite

dense and no low-density area exists in the vicinity, DBSCAN becomes inefficient. DBSCAN also identifies the outliers in the data. So it is less accurate than previous models for our data.

2. The plot from DBSCAN has fewer clusters than the plots from k-means and GMM. DBSCAN has also identified the outliers and has not assigned them any cluster which was not the case with the other models. There are clear regions of low density between nearly all pair of clusters which was not quite significant for the models used earlier. Some of the clusters have significantly more examples than the other clusters in DBSCAN.

b. The purity score after training examples are assigned to the clusters is **0.591**

c.

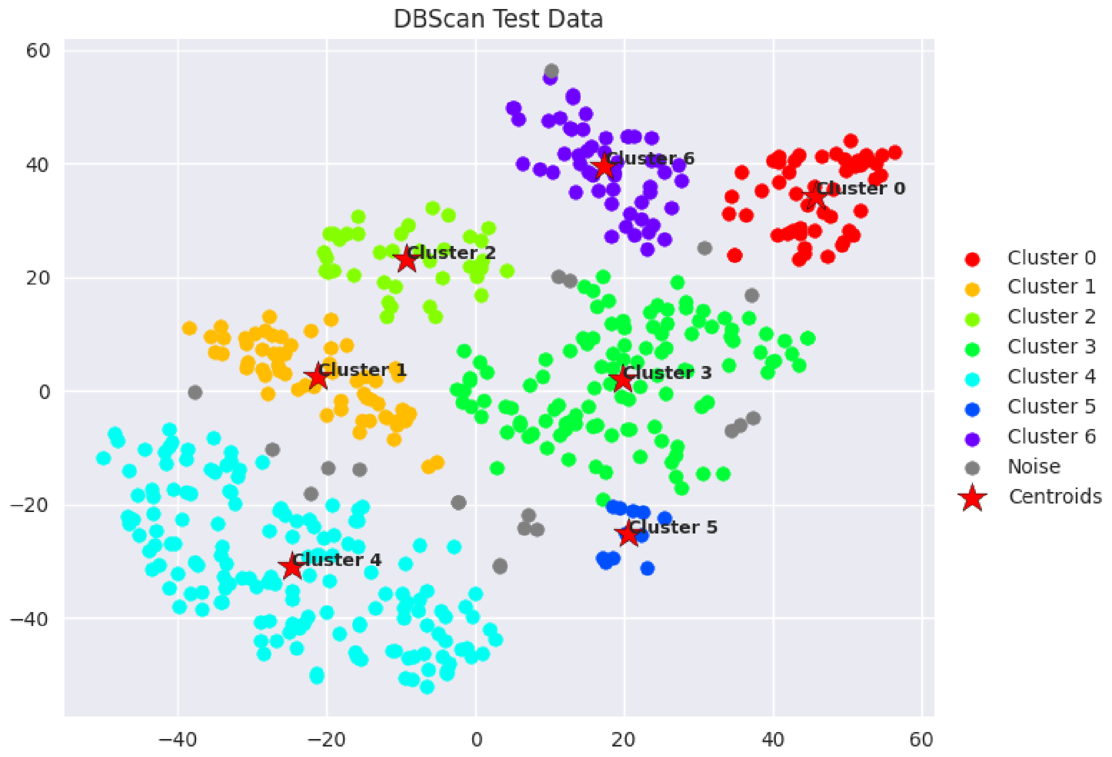


Figure 6 DBSCAN clustering on the test data

Inferences:

1. The clusters in the test data have a lesser number of examples than the clusters in the training data. The clusters were much more concentrated for the training data whereas, for the test data, they are more spread out and not that concentrated and dense.

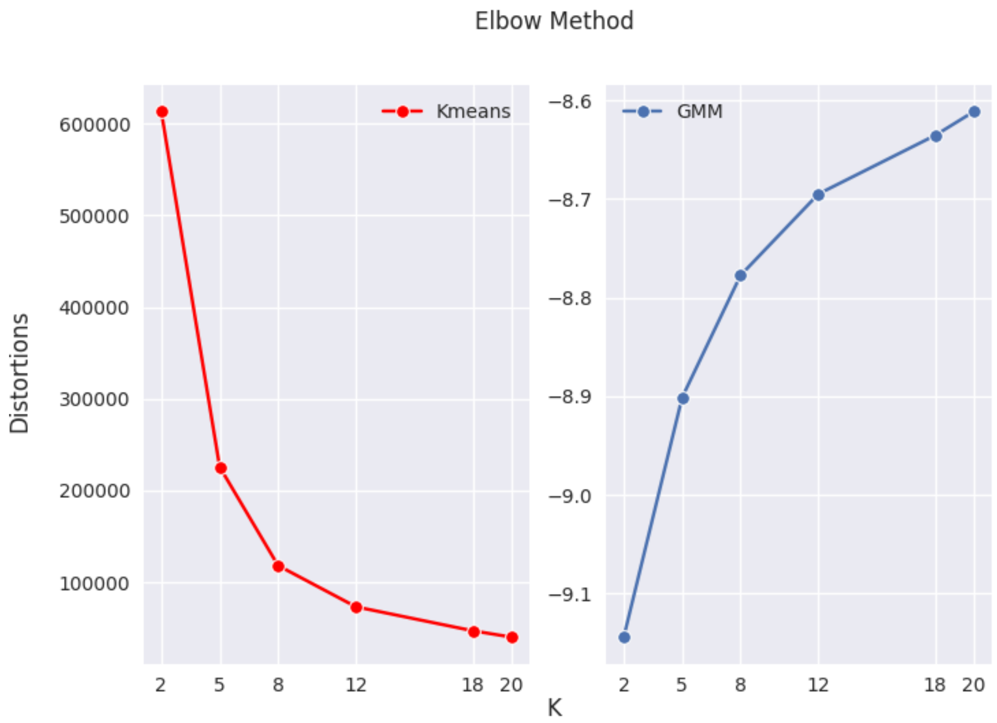
d. The purity score after test examples are assigned to the clusters is **0.584**

Inferences:

1. The training purity score is negligibly higher than the test purity score. This is because the model was originally fit on the training data so it shows a slightly better generalization ability on the training data. However, the difference between the purity scores of the training data and the test data is almost negligible and the model has almost the same clustering prowess for both the training and test data.
2. DBSCAN is not suitable when the data is completely dense and there is no low dense area to separate. The parameters epsilon and min points have to be chosen by the user.

Bonus Questions

A.



Elbow Method for GMM and Kmeans

- Elbow method is used to check how many clusters (K) do we want to have in our data set. It involves plotting the distortions obtained for each value of K, we pick the elbow of the curve as the number of clusters we wanna have in our data set. Elbow of a curve is the point where the slope visibly changes from high to low (close to flat).
- For KMeans we can see the rapid decrease till K=8 where the distortion decreases to 100,000 from 600,000, hence we can choose the number of clusters to be 8.
- For GMM also the rapid "increase" till K=8 make us infer we can choose K=8 here.

B.

	min_samples			
eps	**1**	**10**	**30**	**50**
1	0.984	0.100	0.100	0.100
5	0.209	0.591	0.158	0.100
10	0.100	0.100	0.100	0.503

Purity Score by varying eps and min_sample

- As we vary eps in problem 3 for *min_samples=10* we get maximum purity score for *eps=5* while rest is very low at 0.1.
- As we vary min_samples in problem 3 for *eps=5* we observe *min_samples=10* is giving us maximum purity score.
- Now we may choose *eps=1* and *min_samples=1* but it would not be suitable as it will be giving us too many clusters. In this dataset *eps=5* and *min_samples=10* would be better choice.