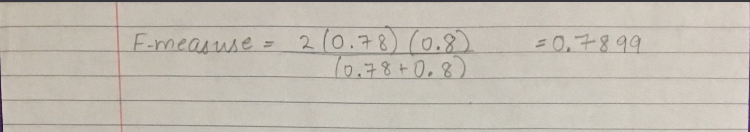
Kanisha Shah

UID: 504958165

**1. Clustering Evaluation**

A close up of text on a white background

Description automatically generatedA close up of a piece of paper

Description automatically generated



Purity = 0.9

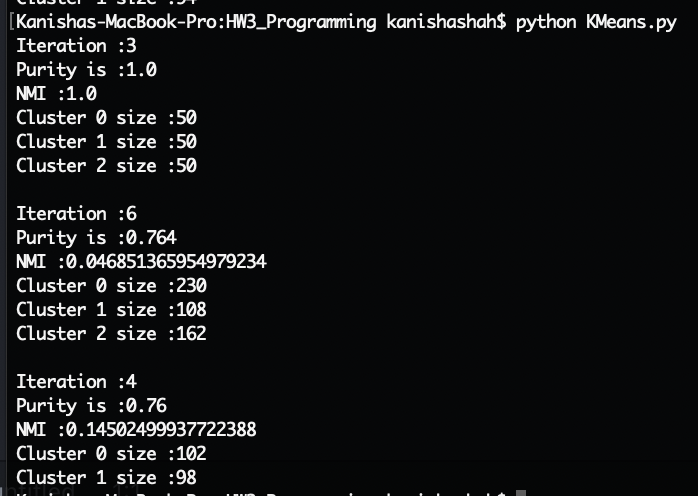
NMI = 0.81

Recall = 32/40 = 0.8

Precision = 32/41 = 0.78

F-Measure = 0.7899

**2. K-Means**



Dataset 1

A screenshot of a cell phone

Description automatically generated

We get a purity of 1. This means that all the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 1. Since NMI measures the quality of clustering, there is a perfect correlation. between both the clusters (ground truth and predicted). So, this clustering has a very good quality.

Dataset 2

A screenshot of a cell phone

Description automatically generated

We get a purity of 0.764. This means that 76.4% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.04685. Since NMI measures the quality of clustering, the correlation is very low between both the clusters (ground truth and predicted). So, this clustering has a bad quality.

Dataset 3

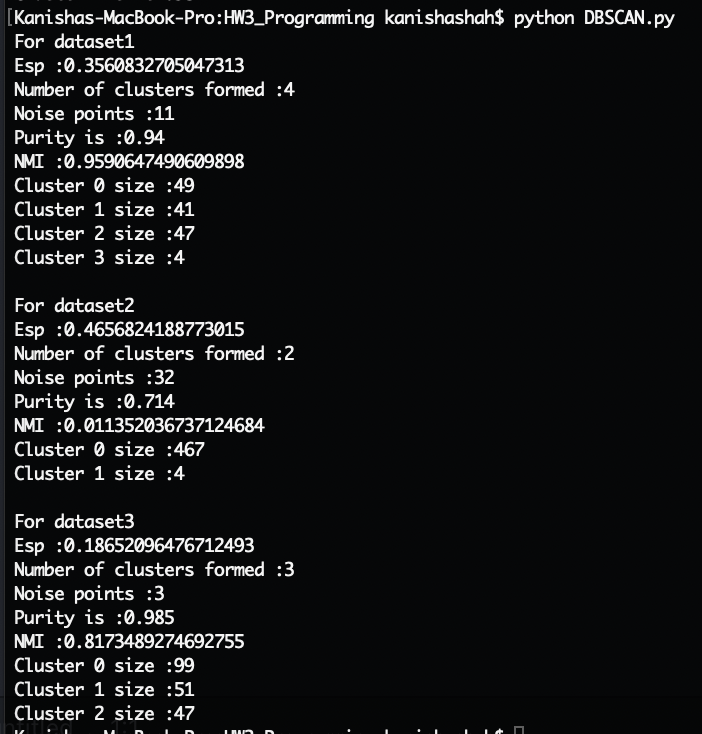
A screenshot of a cell phone

Description automatically generated

We get a purity of 0.76. This means that 76% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.145. Since NMI measures the quality of clustering, the correlation is low between both the clusters (ground truth and predicted). So, this clustering does not have a good quality.

K-Means is simple to understand and implement. It is an efficient algorithm with the time complexity of O(tkn). Here, n is the number of objects, k is the number of clusters, and t is the number of iterations. Since, K-means has unsupervised learning, it makes it very expensive. K-Means cannot cluster non-convex shapes. It only works for spherical shapes. We need to specify the number of clusters before running the algorithm unlike some other algorithms like DBSCAN. It is sensitive to noisy data and outliers. It is only applicable to continuous n-dimensional space.

**3. DBSCAN**



Dataset 1

A screenshot of a cell phone

Description automatically generated

We get a purity of 0.94. This means that 94% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.959. Since NMI measures the quality of clustering, the correlation is high between both the clusters (ground truth and predicted). So, this clustering has a good quality.

Although the clustering is same as K-Means, we can see some noise points in the plot which were not seen in K Means.

Dataset 2

We get a purity of 0.714. This means that 71.4% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.011. Since NMI measures the quality of clustering, the correlation is low between both the clusters (ground truth and predicted). So, this clustering has a bad quality.

This dataset has varied density and size. We can see DBSCAN having trouble with it.

A screenshot of a cell phone

Description automatically generated

Dataset 3

We get a purity of 0.985. This means that 98.5% of the data points in the output clustering matches with the ground truth clustering.

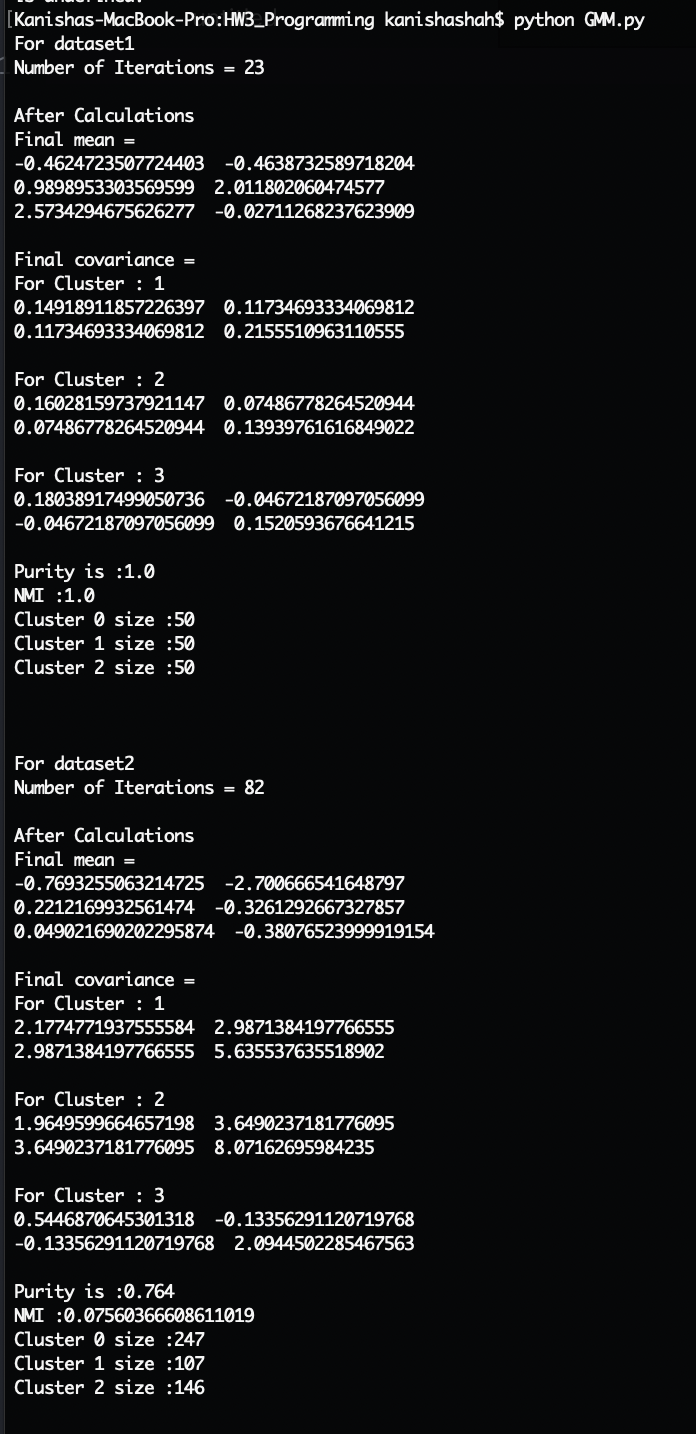
We get NMI (Normalized Mutual Information) of 0.817. Since NMI measures the quality of clustering, the correlation is high between both the clusters (ground truth and predicted). So, this clustering has a good quality.

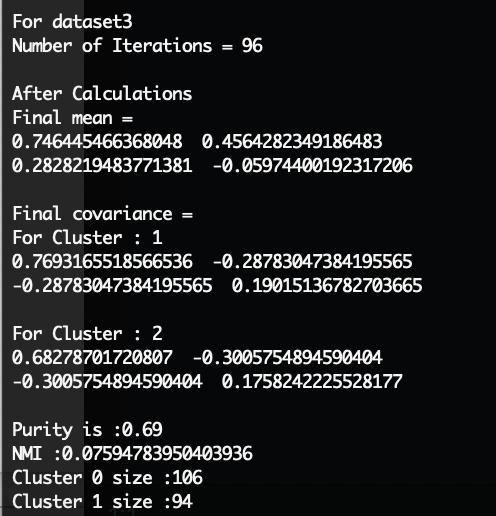
DBSCAN makes 2 clusters whereas K-Means made 2 clusters because K-Means is not applicable with non-convex shapes.

A screenshot of a cell phone

Description automatically generated

DBSCAN does not need to be given the number of clusters before running the algorithm like K-Means. DBSCAN can cluster any size and shape effectively. It can handle noise data. Since DBSCAN relies on the density of the dataset, it does not work well with the datasets hang varied densities. The constants: Eps distance (ε) and MinPts needs to set beforehand as they determine the number and size of clusters. It is challenging to set them just right. DBSCAN has a time complexity of O(n2).

**4. GMM**





We get a purity of 1. This means that all the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 1. Since NMI measures the quality of clustering, there is a perfect correlation. between both the clusters (ground truth and predicted). So, this clustering has a very good quality.

GMM does not perform as good as DBSCAN. It cannot classify outliers.

Dataset 1

A screenshot of a cell phone

Description automatically generated

Dataset 2

A screenshot of a cell phone

Description automatically generated

We get a purity of 0.764. This means that 76.4% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.0756. Since NMI measures the quality of clustering, there is a low correlation. between both the clusters (ground truth and predicted). So, this clustering has a bad quality.

We can see that some of the points assigned blue are in purple’s area.

Dataset 3

We get a purity of 0.69. This means that 69% of the data points in the output clustering matches with the ground truth clustering.

We get NMI (Normalized Mutual Information) of 0.0759. Since NMI measures the quality of clustering, there is a low correlation. between both the clusters (ground truth and predicted). So, this clustering has a bad quality.

Clearly, DBSCAN clusters the dataset better than GMM and K-Means.

A screenshot of a cell phone

Description automatically generated

GMM (Gaussian Mixture Models) is good at clustering spherical shapes only. It cannot deal with non-convex shapes. The computation increases with the dimensionality of the data and is expensive for high-dimensional data. It gives us more flexibility using cluster covariance compared to K-means. So, GMM can cluster datasets with different densities, sizes having spherical shapes.