PRML LAB 6 REPORT

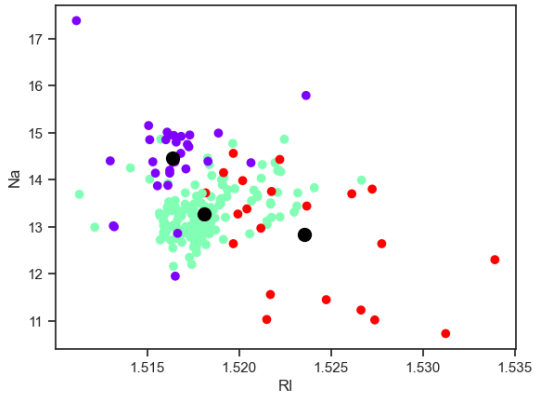
MANISH(B21CS044)

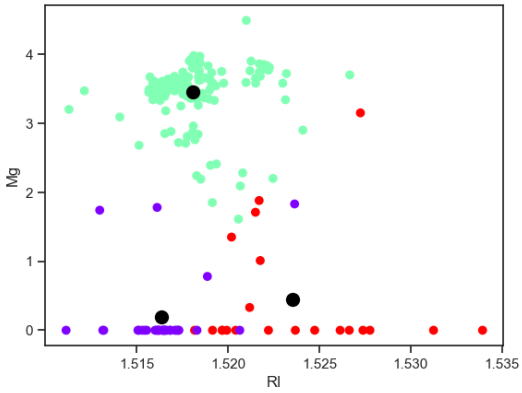
QUESTION 1

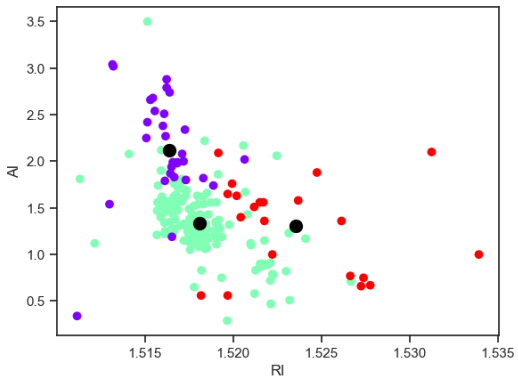
a) Build a k-means clustering algorithm(can use sklearn library) and implement using the value of k which you find suitable. Visualize this part by showing the clusters along with the centroids. **[10 Marks]**

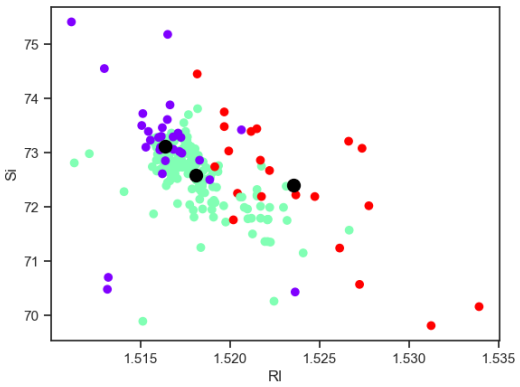
After,performing preprocessing and visualization of the dataset. We have selected the value of k = 3 and clustered the dataset using k-means clustering.
Here is the visualization of clusters along their centers.

The visualization is between every two features and black dots are centroids.



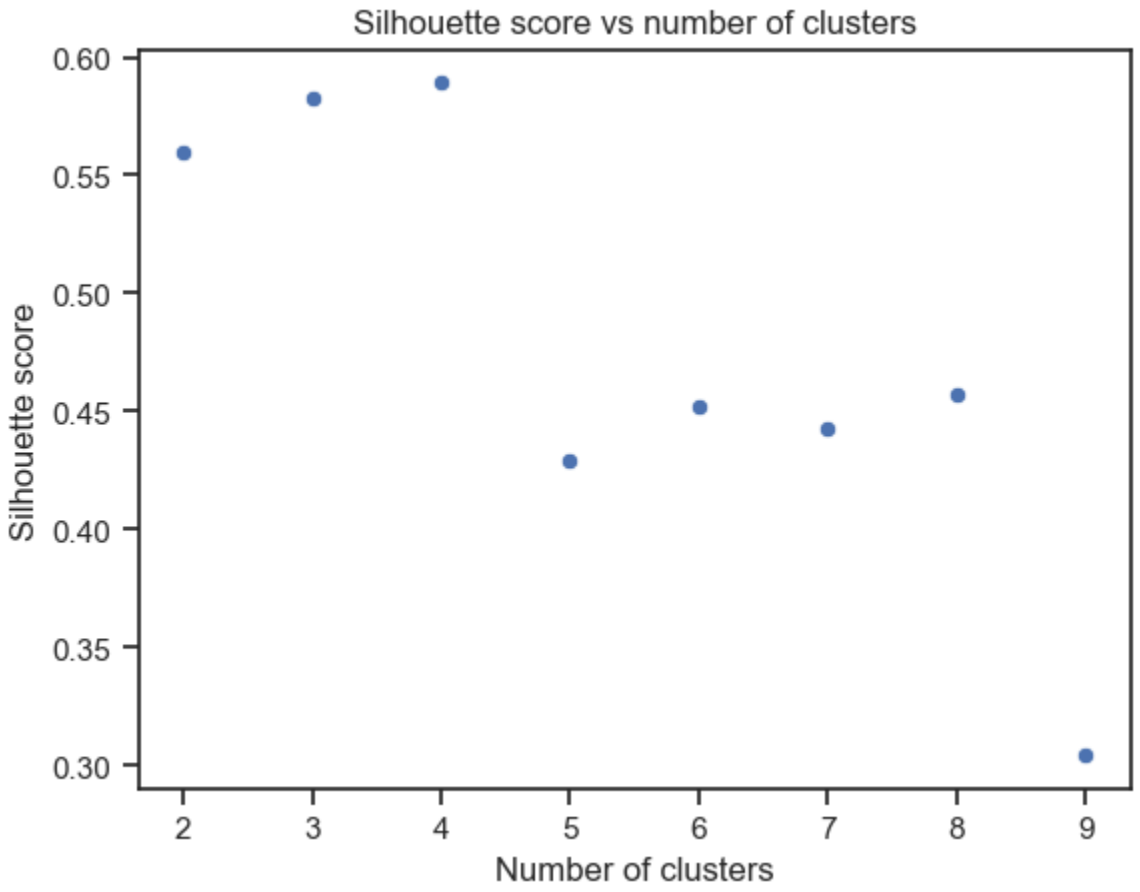






b) Use different values of k and find the Silhouette score and then tell which value of k will be optimal and why? **[8 Marks]**

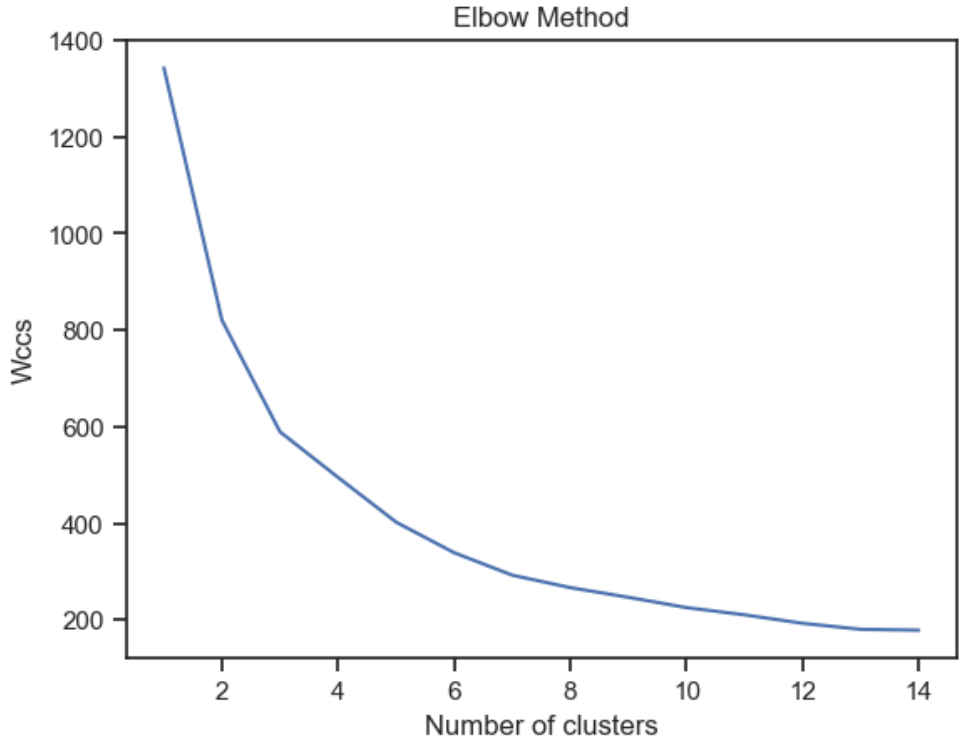
In this part using different values of K we have found out the best number of clusters using Silhouette Score. Here is the plot for the same:



Since we have checked for the silhouette score and the best score is 0.5890367504178873 is for the number of clusters = 4. So the best value of k = 4.

c) There are few methods to find the optimal k value for k-means algorithm like the Elbow Method . Use the above method to find the optimal value of k. **[5 Marks]**

In this part we have applied the elbow method to find the best value of K. Here is the plot for the same:



So from the elbow method we can see that after the value of k = 5 it doesn't decrease the value of the Wcss more fastly so we can say that the value of the K(no of clusters) = 5

d) Apply bagging with the KNN classifier as the base model. Show results with different values of K(=1,2,3). Comment on whether the accuracy changes or not after bagging with KNN along with the proper reason in terms of variance and bias.**[7 Marks]**

In this part we have implemented the Bagging Classifier for k = 1,2,3. Here are the accuracies for the same:

Accuracy of k=1: 0.8372093023255814

Accuracy of k=2: 0.7906976744186046

Accuracy of k=3: 0.7441860465116279

When k is very small, such as k=1, the KNN classifier tends to have high variance and low bias. This is because it relies heavily on the training data to make predictions and can overfit to the noise in the data. As a result, it may perform well on the training set but poorly on the test set, leading to a large difference between the training and testing accuracies.

On the other hand, when k is very large, such as k=3, the KNN classifier tends to have high bias and low variance. This is because it averages the predictions of more neighbors, which can lead to a smoother decision boundary and reduce the effect of noise in the data. However, this can also lead to underfitting and poor performance on both the training and testing sets.

QUESTION 2

a) Implement a k-means clustering algorithm from scratch. **[8 Marks]**
b) Make sure that it should:
  i) Be a class which will be able to store the cluster centers. **[1 Marks]**
  ii) Take a value of k from users to give k clusters. **[2 Marks]**
  iii) Be able to take initial cluster center points from the user as its initialization. **[1 Marks]**
  iv) Stop iterating when it converges (cluster centers are not changing anymore) or, a maximum iteration (given as max_iter by user) is reached. **[2 Marks]**

In this I have implemented the K-means Clustering algorithm from scratch. The class K-means contains fit , predict , final_centroids and SSE as methods.

Fit performs the clustering of the dataset.
Predict gives the cluster number of a particular dataset.
Final_centroids gives the final centroids of the dataset.
SSE gives the Sum of Squared Error Score for the model.

Max_iter , k and initial_centroids are the variables to pass in the class for intialization.

c) Train the k-means model on Olivetti data with k = 40 and 10 random 4096 dimensional points (in input range) as initializations. Report the number of points in each cluster. **[8 Marks]**

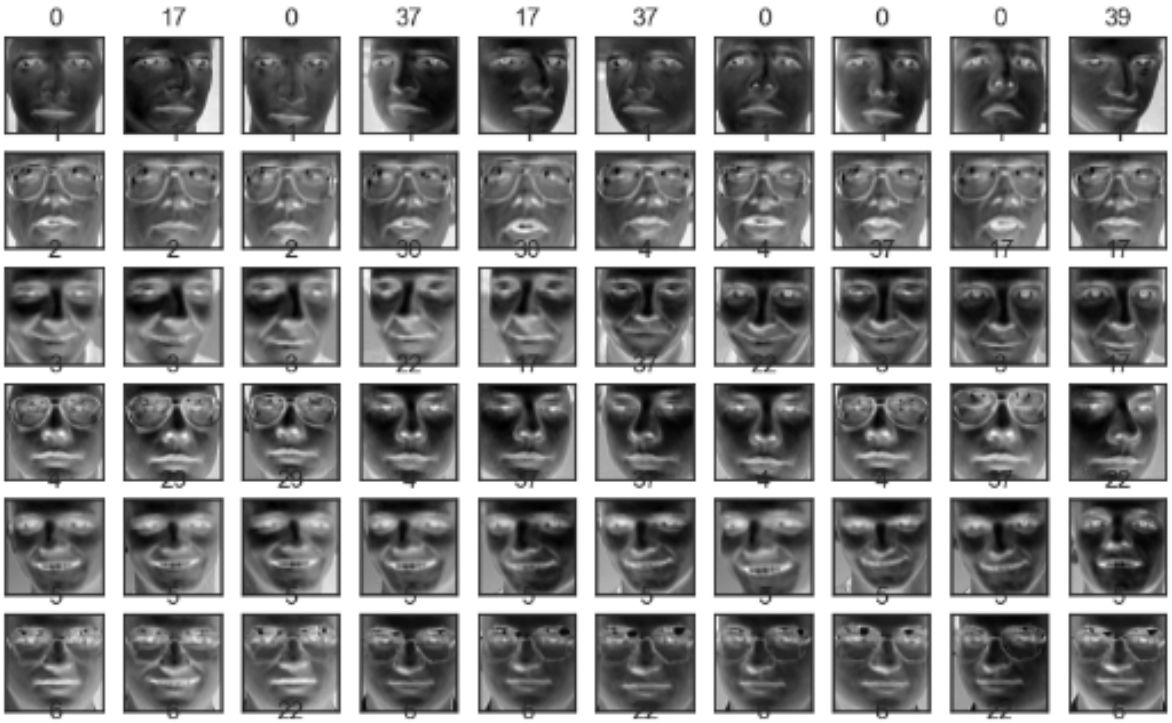
In this part we have clustered the Dataset for k = 40 and given 0+10*i as input points of the clusters.
Here are the number of points for each clusters:

Counter({4: 23, 37: 21, 22: 21, 29: 19, 23: 17, 17: 16, 10: 15, 0: 14, 30: 14, 2: 13, 8: 13, 3: 11, 12: 11, 1: 10, 5: 10, 13: 10, 21: 10, 24: 10, 26: 10, 27: 10, 31: 10, 32: 10, 33: 10, 38: 10, 39: 9, 28: 9, 6: 7, 11: 6, 34: 6, 7: 5, 14: 5, 18: 5, 20: 5, 36: 5, 25: 5, 16: 4, 19: 4, 9: 3, 15: 2, 35: 2})

d) Visualize the cluster centers of each cluster as 2-d images of all clusters. **[4 Marks]**

Here is the visualization of the each cluster centers:

e) Visualize 10 images corresponding to each cluster. **[3 Marks]**



This is the cropped photo. Since, there were many clusters which don't have 10 images. So,I have written over the image from which cluster it belongs to.

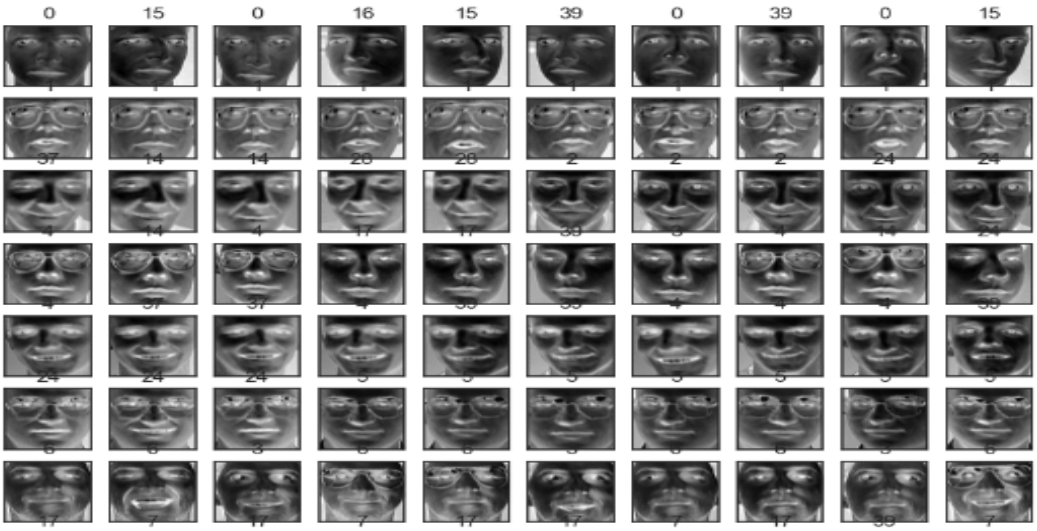
f) Train another k-means model with 10 images from each class as initializations , report the number of points in each cluster and visualize the cluster centers. **[5 Marks]**

Here is the number of points in each cluster:

Counter({14: 26, 24: 22, 22: 18, 26: 18, 29: 17, 38: 16, 37: 15, 4: 14, 3: 13, 18: 13, 39: 12, 17: 12, 2: 11, 20: 11, 23: 11, 1: 10, 8: 10, 10: 10, 13: 10, 32: 10, 33: 10, 36: 10, 21: 9, 0: 8, 9: 8, 30: 8, 5: 7, 6: 7, 27: 7, 35: 7, 28: 6, 11: 6, 16: 5, 25: 5, 15: 4, 7: 4, 19: 4, 34: 4, 12: 1, 31: 1})

Here is the plot for the cluster centers:

g) Visualize 10 images corresponding to each cluster. **[2 Marks]**



Here is the cropped photo having cluster numbers on each image.

h) Evaluate Clusters of part c and part f with Sum of Squared Error (SSE) method. Report the scores and comment on which case is a better clustering. **[4 Marks]**

Here are the SSE scores for both the models:

SSE for part c : 13799.403483402302

SSE for part f : 12722.161775846207

We know that SSE scores should be minimum for a model to work better. So, from here we can see that SSE for part f is less than part c. So, clustering in part f is better than that of part c. Because we know that in part c we have passed random initial centroids and in part f we have passed correct centroids.

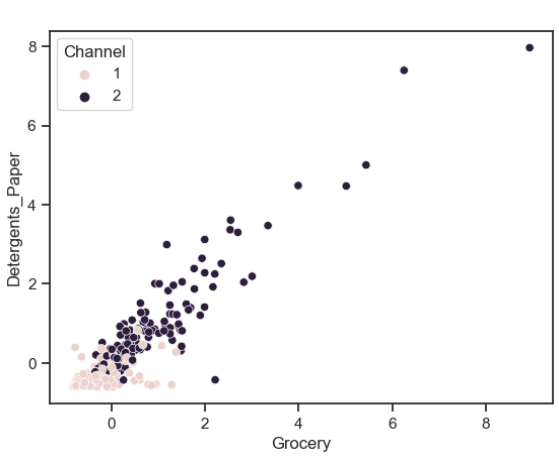
QUESTION 3

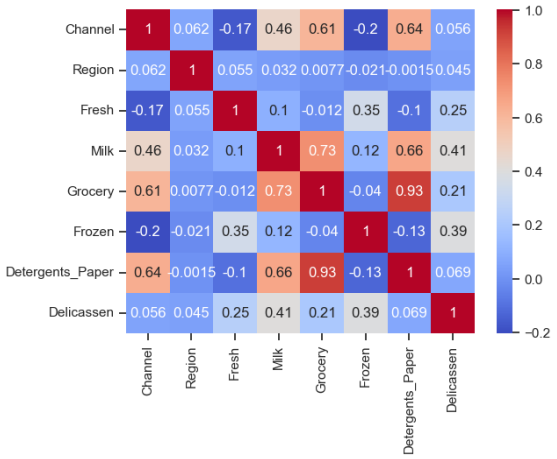
A. Check out the dataset & preprocess the data so that the scale of each variable will be the same. **[5 Marks]**

In this part I have scaled the dataset using Standard Scaler so that the scale of each variable will be the same.

B. Find out the covariance between the pair of features with which you can best visualize the outliers. Also, visualize the same set of features. **[7 Marks]**

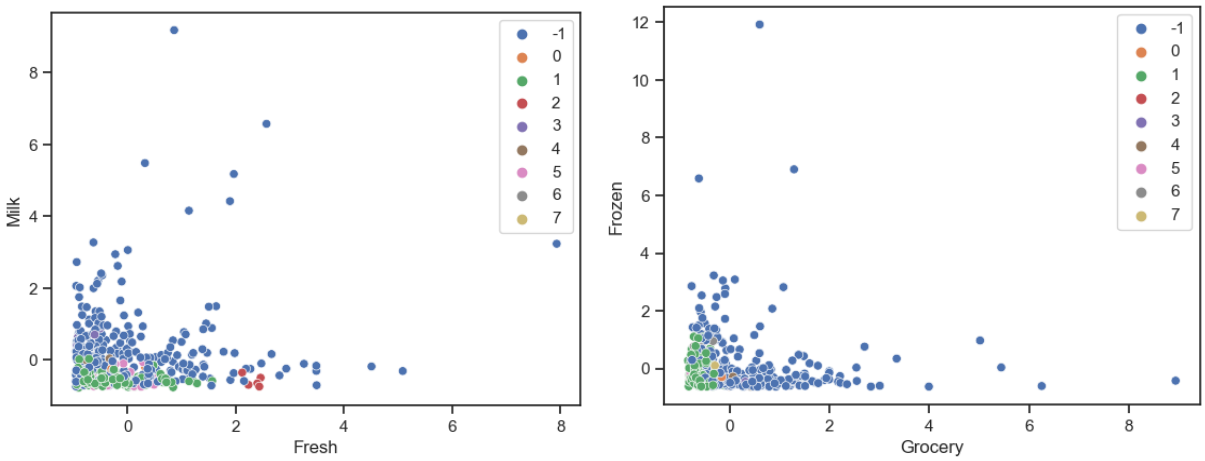
In this part I have plotted the heatmap for the covariance between all the variables of the dataset. From this we can clearly see that the features Detergents_Paper and Grocery are having maximum covariance. So, these two can be used to visualize the outliers best. Here are the plots for the same:





C. Apply DBSCAN to cluster the data points and visualize the same. **[5 Marks]**

In this part I have applied DBSCAN on the dataset for values epsilon = 0.5 and min_samples = 5. Here are the plots for the same for two - two features:

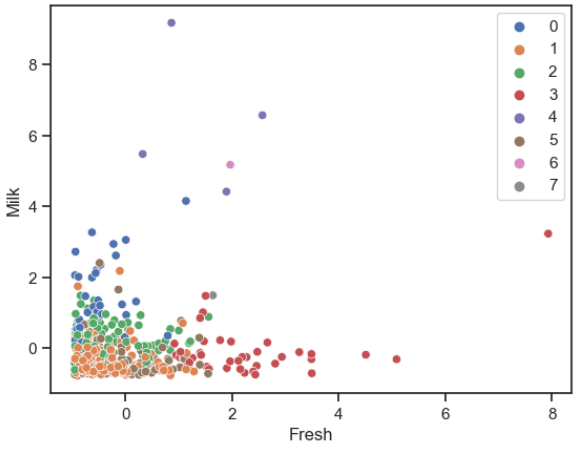


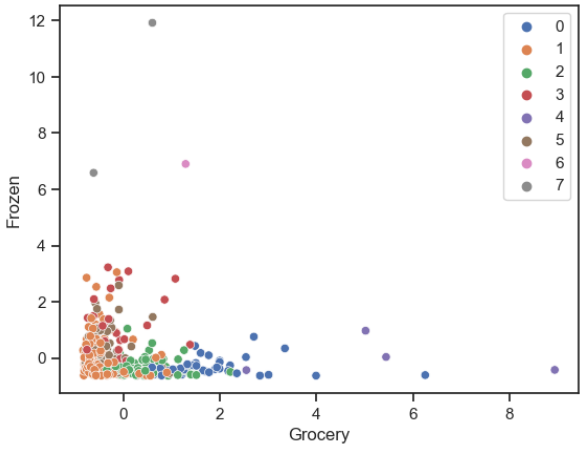
D. Apply KNN on the same dataset and compare the visualization with DBSCAN. Comment on what you observe with reason in the report. **[5 Marks]**

In this part I have applied the K-Means algorithm for the same Dataset. I have given k = 8 for this to maintain the same number of clusters for both methods.

DBSCAN is better than KMeans because it will find the core points and then it will find the neighbors of the core points and then it will assign the data points to the cluster of the core points and can also find the outliers which Kmeans is not able to find out.

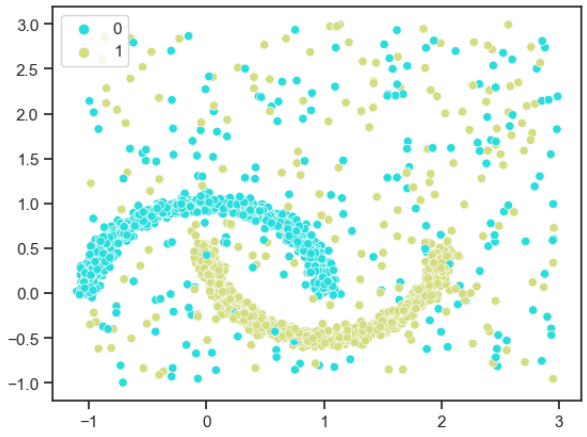
Here are the plots for the visualization of clustering using k-means clustering.

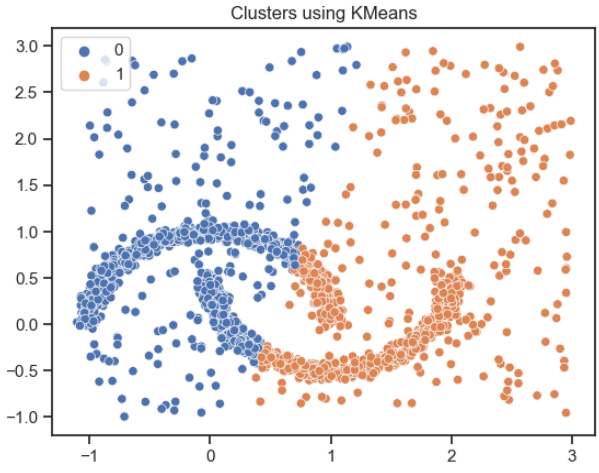


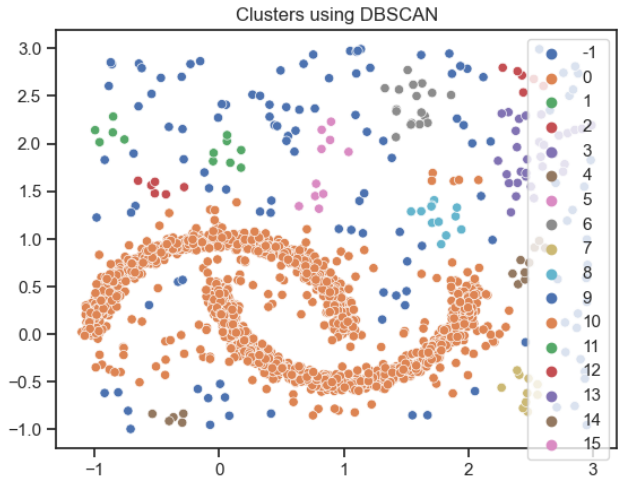


E. Use the make_moons function of sklearn to create a datasat of 2000 points. Add some noise to the plot, i.e., randomly add data points to the plot with a 20% probability. Apply DBSCAN and KNN to cluster them and finally compare the plots and comment on which one is better.**[8 Marks]**

In this part I have generated the dataset using probability of 0.2 to add the noise to the dataset. And here are the plots for the same after applying the K-means , DBSCAN and initial generated dataset.







As we can see from the plots also DBSCAN is way better than K-MEANS because DBSCAN is able to find the clusters properly and it is able to find the outliers also. We can also see that it is able to differentiate the noise from the original dataset while K-means is not able to differentiate between them. DBSCAN has correctly clustered the dataset into two classes but in case of the Kmeans it is not able to cluster the dataset into two classes. In short, DBSCAN is better than K-MEANS as it can differentiate between the outliers and the normal data points.