Assignment 3

GROUP-4 MEMBERS:

- AMIRA ABU ISSA
- HEBA MOSTAFA
- AYA METWALLY

7/6/2023

Part 1: Calculations

(a)

Step 1: Calculate the Euclidean distances between each data point and each centroid.

Distance between A1 and A2: sqrt((3-6)^2 + (6-3)^2) = sqrt(18) = 4.243

Distance between A1 and A4: sqrt((3-2)^2 + (6-1)^2) = sqrt(26) = 5.099

Distance between A2 and A2: 0

Distance between A2 and A4: sqrt((6-2)^2 + (3-1)^2) = sqrt(32) = 4.472

Distance between A3 and A2: sqrt((8-6)^2 + (6-3)^2) = sqrt(13) = 3.606

Distance between A3 and A4: sqrt((8-2)^2 + (6-1)^2) = sqrt(85) = 7.810

Distance between A4 and A4: 0

Distance between A5 and A2: sqrt((5-6)^2 + (9-3)^2) = sqrt(52) = 6.083

Distance between A5 and A4: sqrt((5-2)^2 + (9-1)^2) = sqrt(83) = 8.544

Step 2: Assign each data point to the closest centroid.

A1 is closer to A2 than to A4, so it is assigned to cluster 1 (centered around A2).

A2 is a centroid, so it is assigned to cluster 1 (centered around A2).

A3 is closer to A2 than to A4, so it is assigned to cluster 1.

A4 is a centroid, so it is assigned to cluster 2 (centered around A4).

A5 is closer to A2 than to A4, so it is assigned to cluster 1.

Step 3: Recalculate the centroids of each cluster.

The centroid of cluster 1 is the mean of the coordinates of A1, A2, A3, and A5: ((3+6+8+5)/4, (6+3+6+9)/4) = (5.5, 6.00)

new_centroide of cluster 1=(5.5, 6.00)

The centroid of cluster 2 is the same as the initial centroid A4: (2, 1)

Repeat steps 1,2 and 3 until all points converge and cluster centers stop moving.

Step 1: Calculate the Euclidean distances between each data point and each centroid.

Distance between A1 and new_centroide: sqrt((3-5.5)^2 + (6-6)^2) = 2.5

Distance between A1 and A4: sqrt((3-2)^2 + (6-1)^2) = 5.099

Distance between A2 and new_centroide: sqrt((5.5-6)^2 + (6-3)^2) = 3.041

Distance between A2 and A4: sqrt((6-2)^2 + (3-1)^2) = 4.472

Distance between A3 and new_centroide: sqrt((5.5-8)^2 + (6-6)^2) = 2.5

Distance between A3 and A4: sqrt((8-2)^2 + (6-1)^2) = 7.810

Distance between A4 and A4: 0

Distance between A5 and new_centroide: sqrt((5.5-5)^2 + (6-9)^2) = 3.041

Distance between A5 and A4: sqrt((5-2)^2 + (9-1)^2) = 8.544

Step 2: Assign each data point to the closest centroid.

A1 is closer to A2 than to A4, so it is assigned to cluster 1 (centered around A2).

A2 is a centroid, so it is assigned to cluster 1 (centered around A2).

A3 is closer to A2 than to A4, so it is assigned to cluster 1.

A4 is a centroid, so it is assigned to cluster 2 (centered around A4).

A5 is closer to A2 than to A4, so it is assigned to cluster 1.

Step 3: Recalculate the centroids of each cluster.

The centroid of cluster 1 is the mean of the coordinates of A1, A2, A3, and A5: ((3+6+8+5)/4, (6+3+6+9)/4) = (5.5, 6.00)

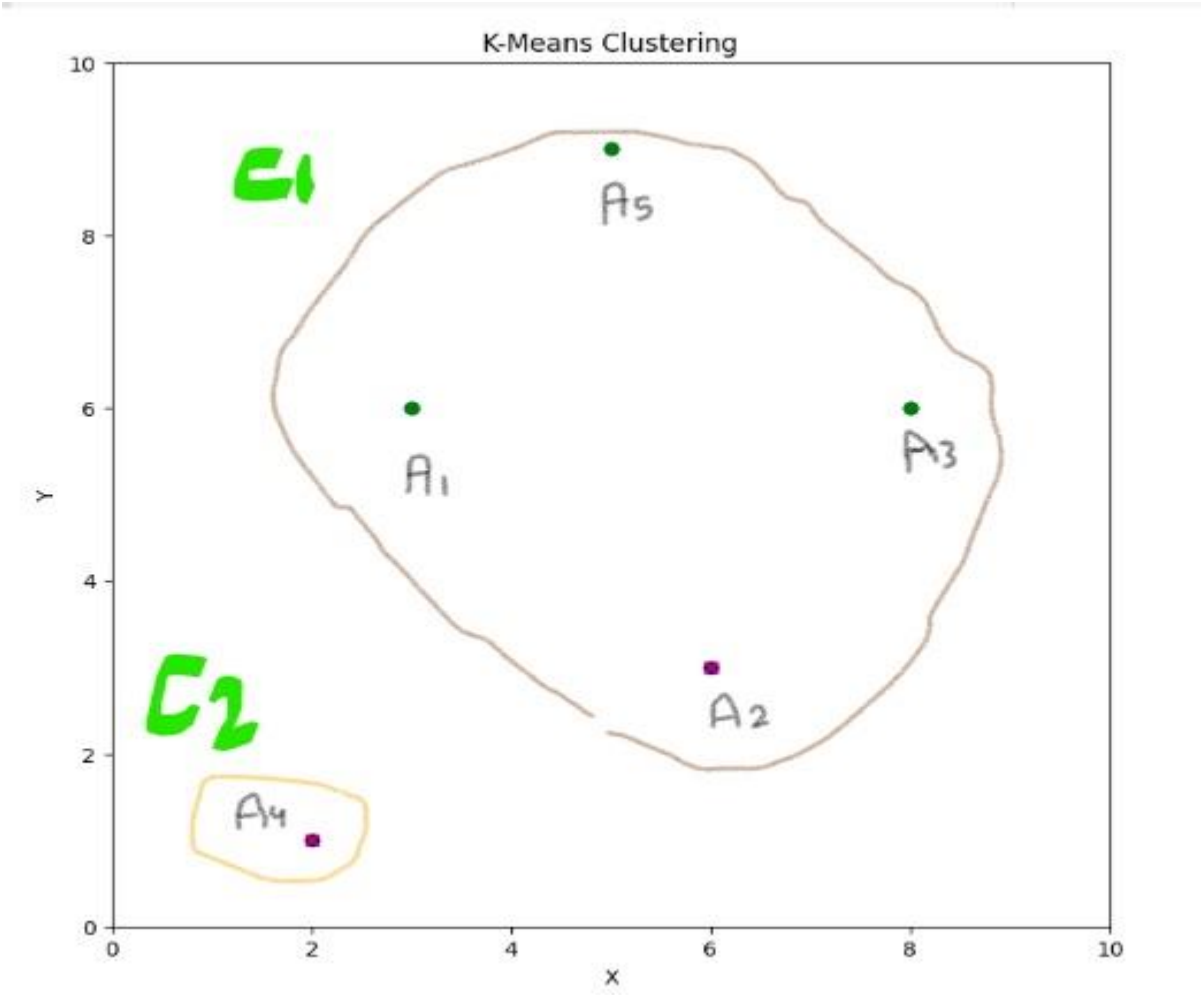
new_centroide of cluster 1=(5.5, 6.00)

The centroid of cluster 2 is the same as the initial centroid A4: (2, 1)

So we have 2 clusters :cluster1(A1,A2,A3andA5) , cluster2(A4)

(b)

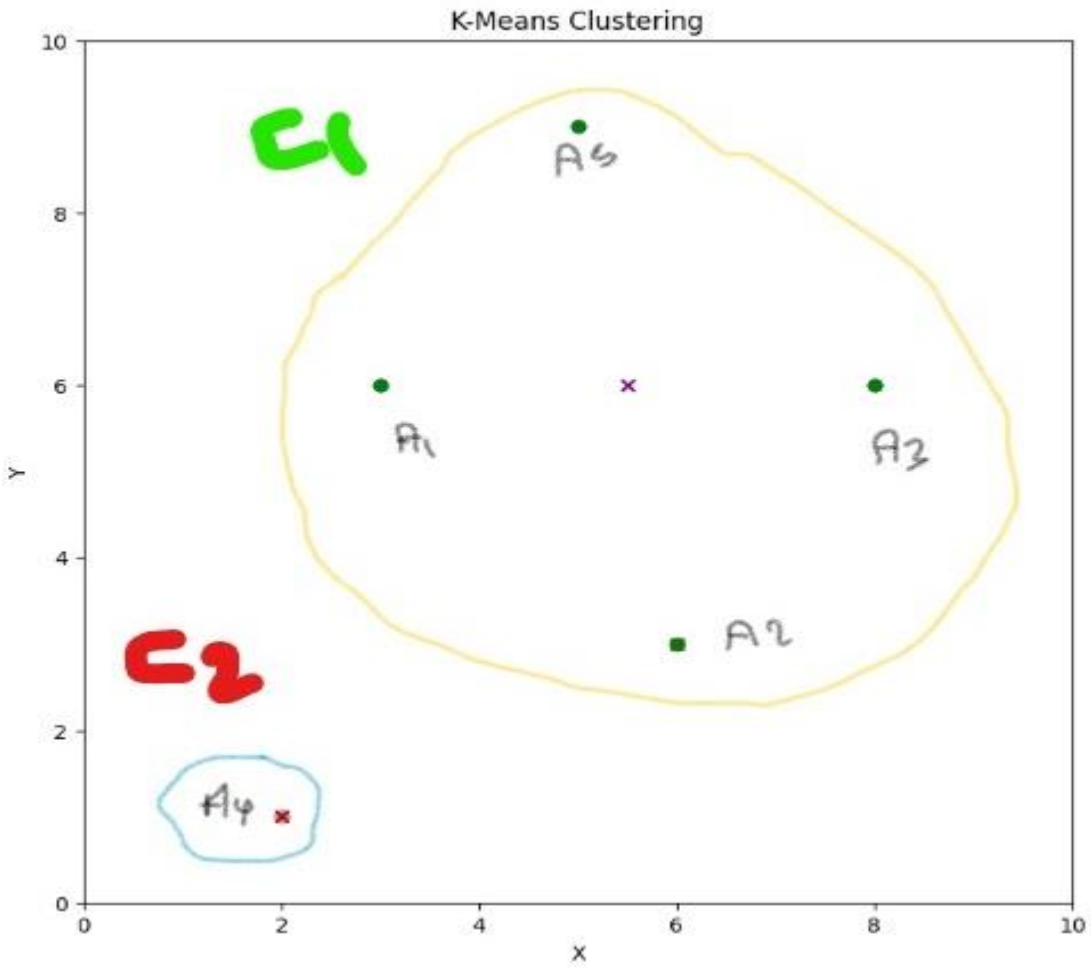
First we have 5 points A1,A2,A3,A4 and A5 the initial centroids (centers of each cluster) are A2 and A4



After Calculate the Euclidean distances A1,A3,A5 are cluster1 and A2 is the centroied of cluster1 ,A4 is centroied of cluster2 and the only point in it.

Then we calculate the new centroied of cluster1 is (5.5, 6.00) and of cluster2 is the same of A4(2,1)

After Calculate the Euclidean distances A1,A2,A3,A5 are cluster1 and new centroied (5.5, 6.00) is the centroied of cluster1 ,A4 is centroied of cluster2 and the only point in it.



(c) Calculate the silhouette score and WSS score

To calculate silhouette score

we must calculate each distance between to point and we use the Euclidean distances between each data point

Distance between A1 and A2: sqrt((3-6)^2 + (6-3)^2) = 4.243

Distance between A1 and A3: sqrt((8-3)^2 + (6-6)^2) = 5.00

Distance between A1 and A4: sqrt((3-2)^2 + (6-1)^2) = 5.099

Distance between A1 and A5: sqrt((5-3)^2 + (9-6)^2) = 3.606

Distance between A2 and A3: sqrt((6-8)^2 + (3-6)^2) = 3.606

Distance between A2 and A4: sqrt((2-6)^2 + (1-3)^2) = 4.472

Distance between A2 and A5: sqrt((6-5)^2 + (3-9)^2) = 6.083

Distance between A3 and A4: sqrt((2-8)^2 + (1-6)^2) = 7.810

Distance between A3 and A5: sqrt((5-8)^2 + (9-6)^2) = 4.243

Distance between A4 and A5: sqrt((2-5)^2 + (1-9)^2) = 8.544

the Euclidean distances between each data point

distance	A1	A2	A3	A4	A5
A1	0	4.243	5	5.099	3.606
A2	4.243	0	3.606	4.472	6.083
A3	5.00	3.606	0	7.810	4.243
A4	5.099	4.472	7.810	0	8.544
A5	3.606	6.083	4.243	8.544	0

S(i)=b(i)-a(i)/Max{a(i),b(i)}

points	a(i)	b(i)	S(i)
A1	$\frac{A2+A3+A5}{3}=4.283$	5.099	5.099-4.283/5.099=0.160
A2	$\frac{A1+A3+A5}{3}=4.644$	4.472	4.472-4.644/4.644= -0.037
A3	$\frac{A1+A2+A5}{3}=4.283$	7.810	7.810-4.283/7.810=0.452
A4	0	$\frac{A1+A2+A3+A5}{4}=6.481$	6.481-0/6.481=1
A5	$\frac{A1+A2+A3}{3}=4.644$	8.544	8.544-4.644/8.544=0.457

Average silhouette=mean{S(i)} $=\frac{0.160-0.037+0.452+1+0.457}{5}=0.41$

To calculate WSS score

Wss=$\sum_{i=1}^{m}(xi - ci)^2$

So we calculate the centroid of cluster 1 and 2 in (a)

point	Centroid cluster1(5.5,6)	Centroid cluster2(2,1)
A1	$(3-5.5)^2+(6-6)^2=6.25$	----------
A2	$(6-5.5)^2+(3-6)^2=9.25$	----------
A3	$(8-5.5)^2+(6-6)^2=6.25$	----------
A4	----------	0
A5	$(5-5.5)^2+(9-6)^2=9.25$	----------
$\sum(xi-ci)^2$	31	0

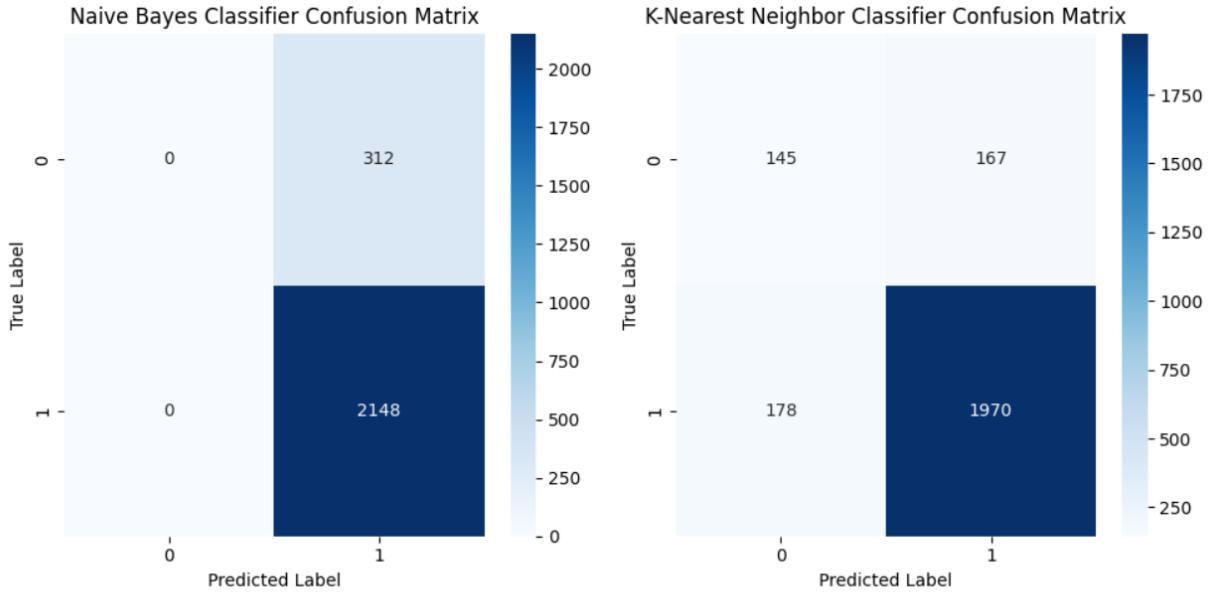
Wss=$\sum_{i=1}^{m}(xi - ci)^2 = 31$

Part 2: Programing

1- (a) Create training and test datasets for remaining parts according to day feature in the dataset (column:“day”)

```
#1(a)

# Split the dataset into training and test datasets based on the day feature
train_data = data[data['Day'].isin([0, 1, 2])]
test_data = data[data['Day'] == 3]

# Separate the features and target variable for training and test datasets
train_features = train_data.drop(['ID', 'Day', 'Ligitimacy'], axis=1)
train_target = train_data['Ligitimacy']
test_features = test_data.drop(['ID', 'Day', 'Ligitimacy'], axis=1)
test_target = test_data['Ligitimacy']

# Split the remaining parts of the dataset into training and test datasets based on the day feature
remaining_data = data[data['Day'].isin([4, 5, 6, 7])]
remaining_train_data, remaining_test_data = train_test_split(remaining_data, test_size=0.2, random_state=42)

# Separate the features and target variable for remaining training and test datasets
remaining_train_features = remaining_train_data.drop(['ID', 'Day', 'Ligitimacy'], axis=1)
remaining_train_target = remaining_train_data['Ligitimacy']
remaining_test_features = remaining_test_data.drop(['ID', 'Day', 'Ligitimacy'], axis=1)
remaining_test_target = remaining_test_data['Ligitimacy']
```

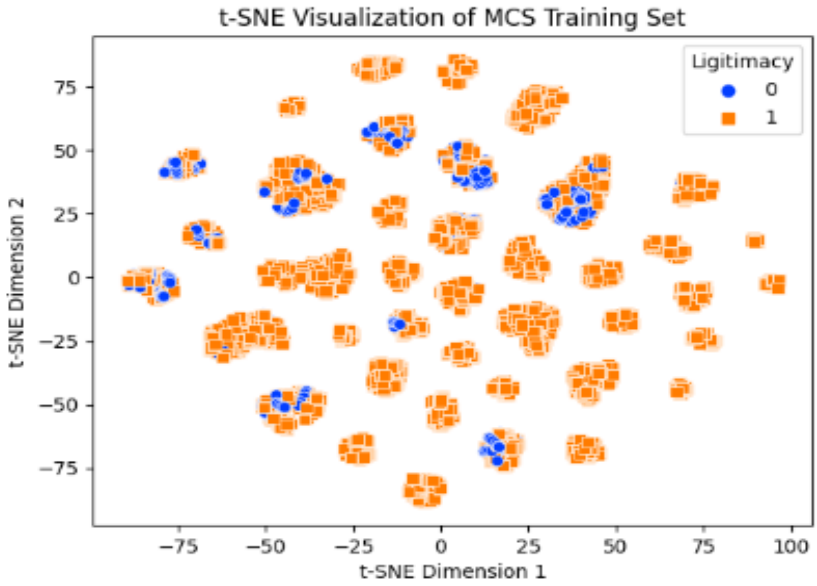
(b) Provide confusion matrixes and F1 scores of NB and KNN classifier as baseline performances.

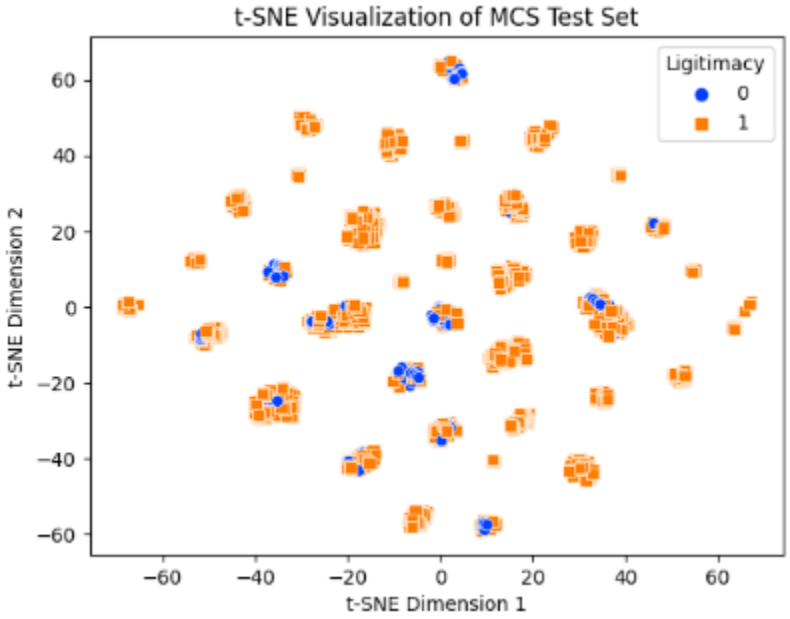



```
Naive Bayes Classifier F1 Score: 0.9322916666666667
K-Nearest Neighbor Classifier F1 Score: 0.9194865810968496
```

F1 scores of NB 0.93 is better than of KNN 0.919

(c) Provide 2D TSNE plots, one for the training set and one for the test set.





2- (a) To find the best reduced dimensions of PCA and AE based on f1 score of test dataset using both classifiers (NB and KNN), plot the number of components (dimension) vs f1 score together with baseline performances for each classifier. The Graph should be plotted based on the f1 score of test dataset. The total number of figures will be 4 in this part.

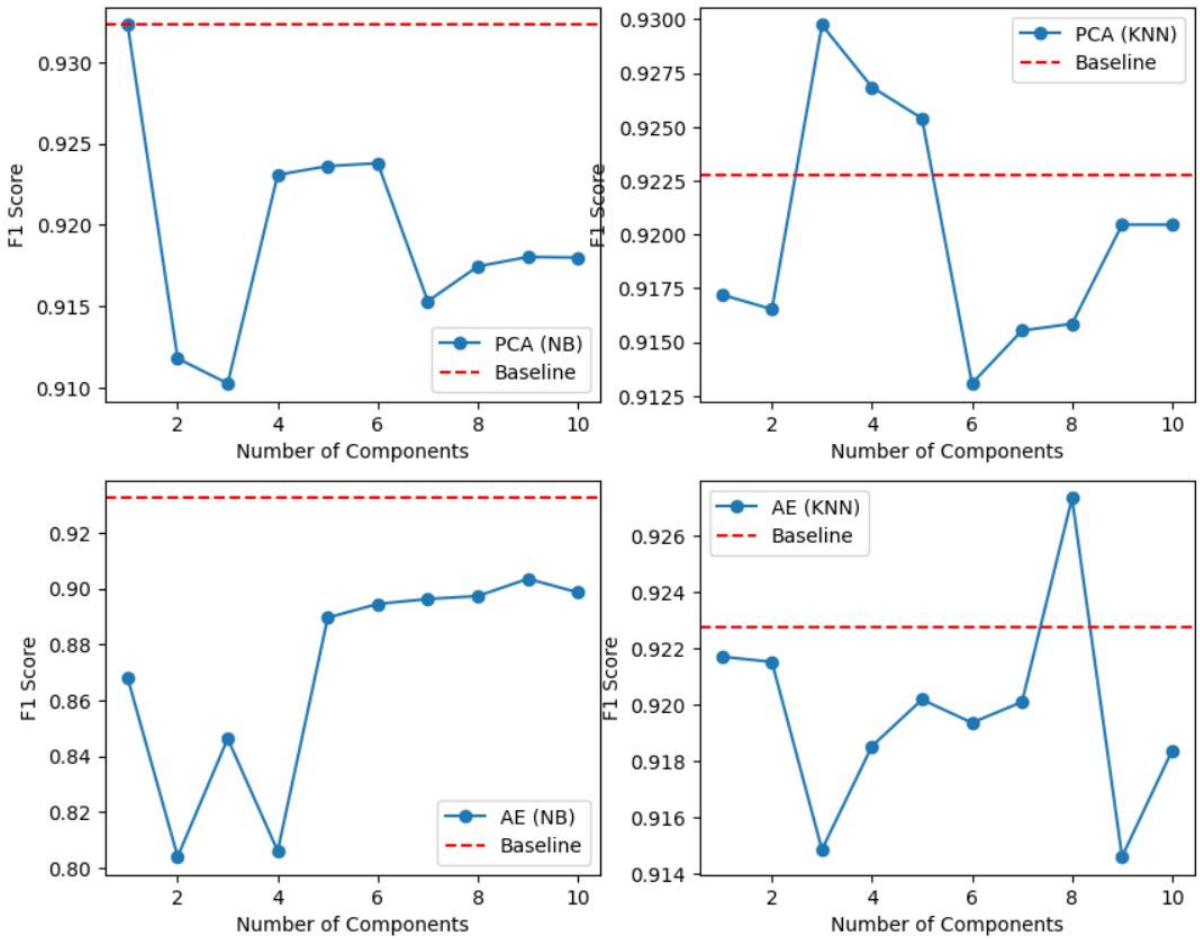
Applying PCA when n-components=1 for Naive Bayes and n-components=3 for K-Nearest Neighbors

```
F1 score for Naive Bayes with n-components=1 is  0.9322916666666667
F1 score for K-Nearest Neighbors with n-components=3 is  0.9297396913153652
```

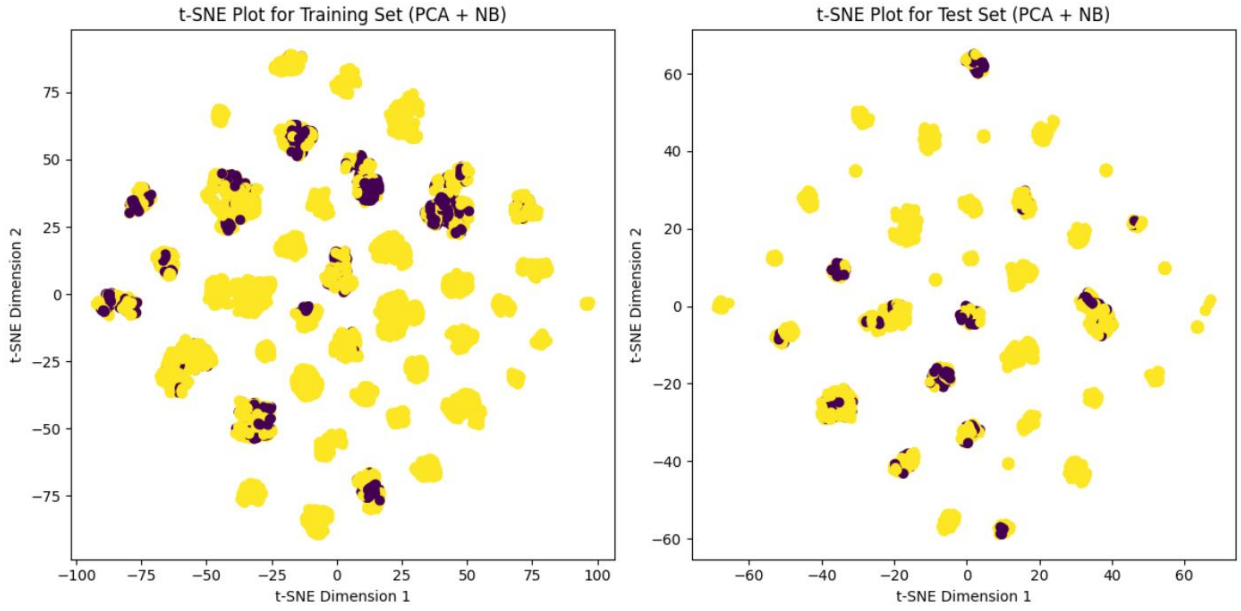
Find the best value of hidden layer size of Autoencoder

```
Best value of hidden_layer_sizes for Naive Bayes (AE): 9
Best value of hidden_layer_sizes for K-Nearest Neighbors (AE): 8

[0.8679338434954333, 0.8840201005025125, 0.8462311557788944, 0.8060413354531001, 0.8895320791123975, 0.8944457802356336, 0.8962219033955044, 0.8972586412395709, 0.9034400948991697, 0.898571428571428
[0.9216962082118016, 0.9215077605321508, 0.9148550724637681, 0.9185185185185186, 0.920172845121674, 0.9193511537582819, 0.9200913242009132, 0.9273182957393483, 0.9146005509641874, 0.918362680082323
```

The total number of figures



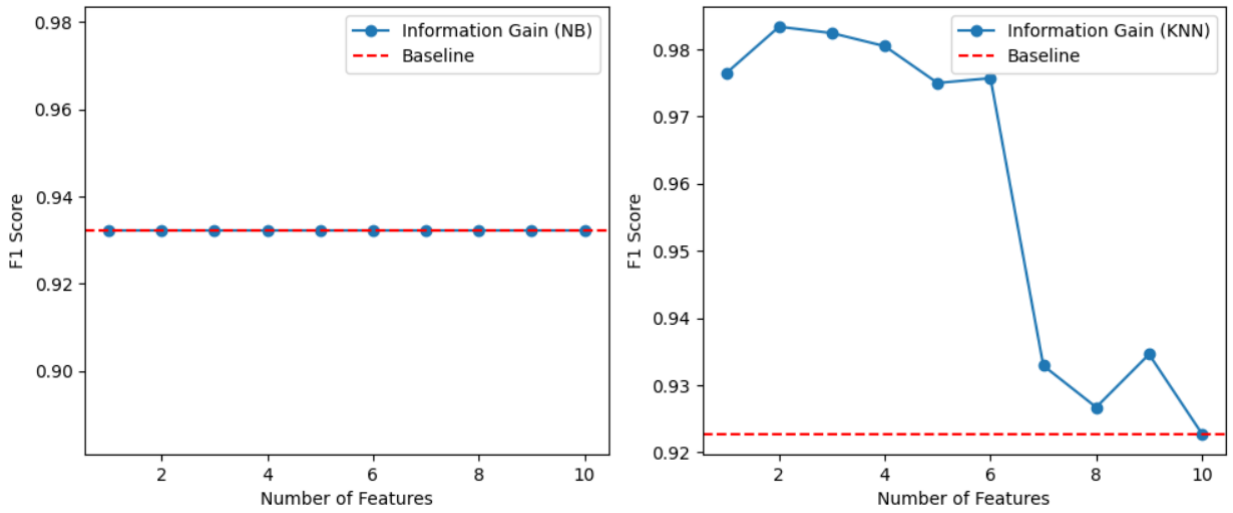
(b) Provide 2D TSNE plots for the best performance in previous part (The best dimensionality reduction performance using one of the NB and KNN classifiers) one for the training set and one for the test set



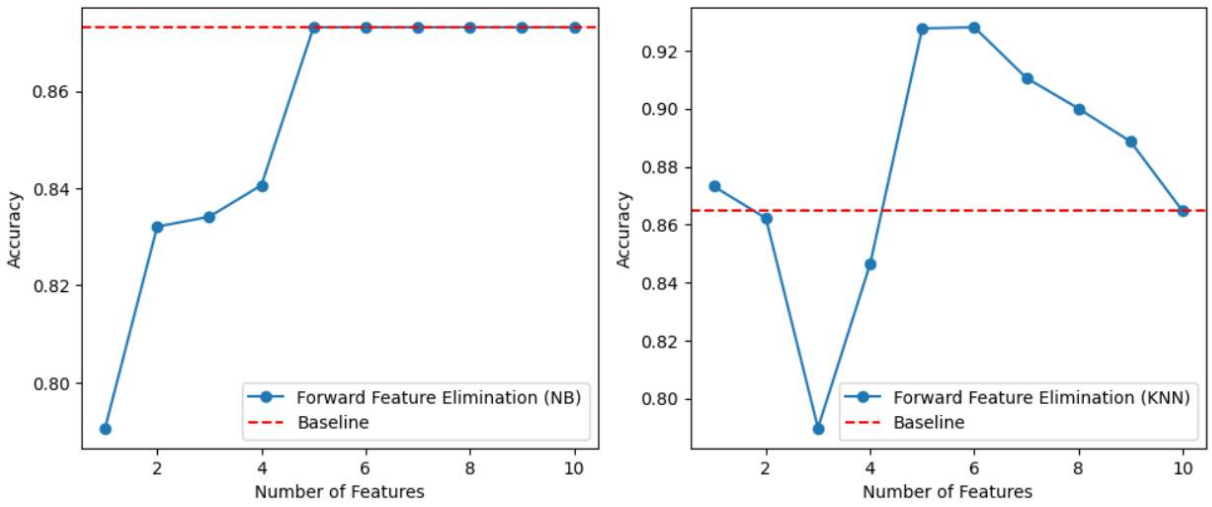
3- Find the best number of features based on both, the NB and KNN classifiers f1 scores

```
Best value of num_features_range for Naive Bayes (IG): 1
Best value of num_features_range for K-Nearest Neighbors (IG): 2

[0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667, 0.9322916666666667]
[0.9764513872697598, 0.9833684703677676, 0.9824232481837357, 0.9804751823100447, 0.9749882019820669, 0.9757132751709503, 0.9328891004283675, 0.9266903914590747, 0.9360112097150864, 0.9227557411273487]
```

3-(a) Filter Methods (Information Gain, Variance Threshold etc.). Plot the number of features vs f1 score with the improved baseline performance.

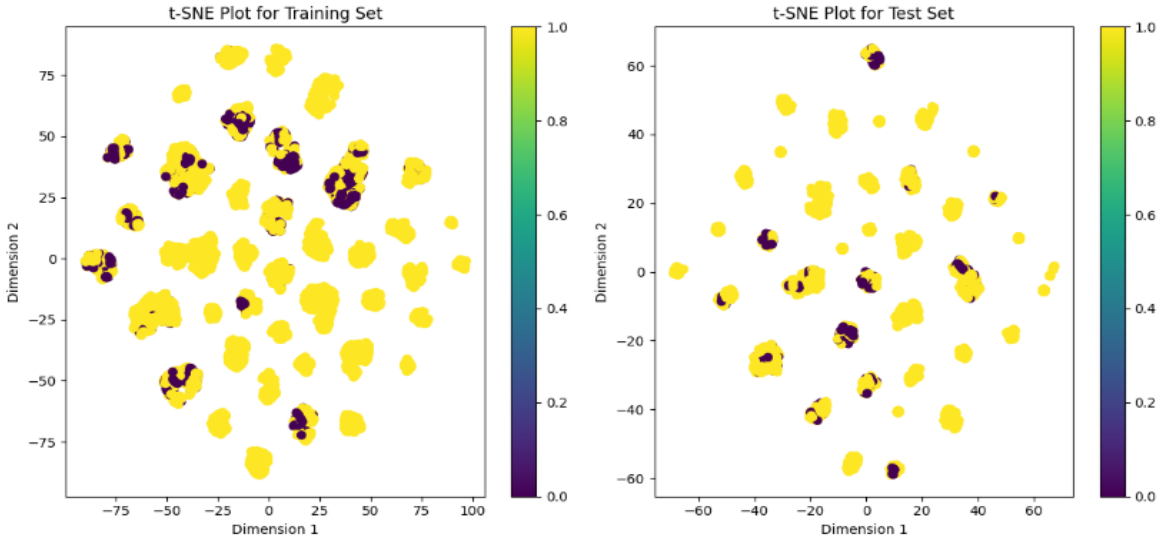


3-(b) Wrapper Methods (Forward or Backward Feature Elimination, Recursive Feature Elimination etc.). Plot the number of features versus accuracy graph with the baseline performance.



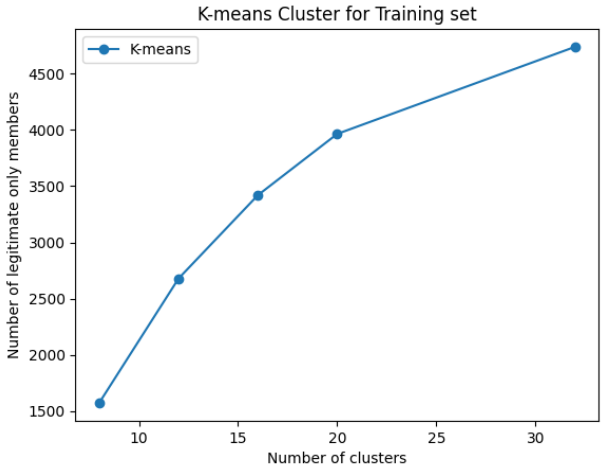
Based on the previous graphs Applying filter methods results the best performance especially with "Information Gain"

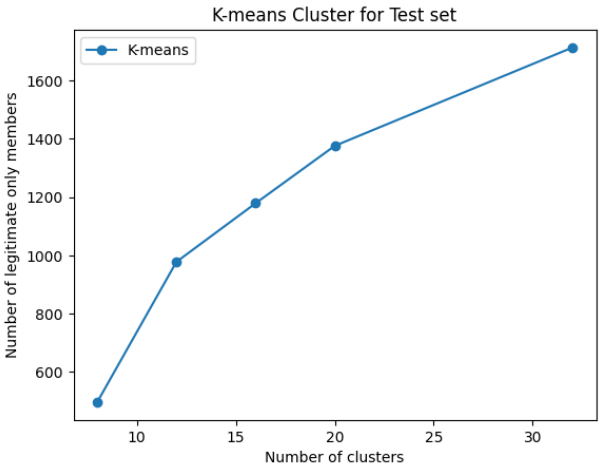
3(c) Provide 2D TSNE plots, one for the training set and one for the test set, using only the best method (either the filter or wrapper).



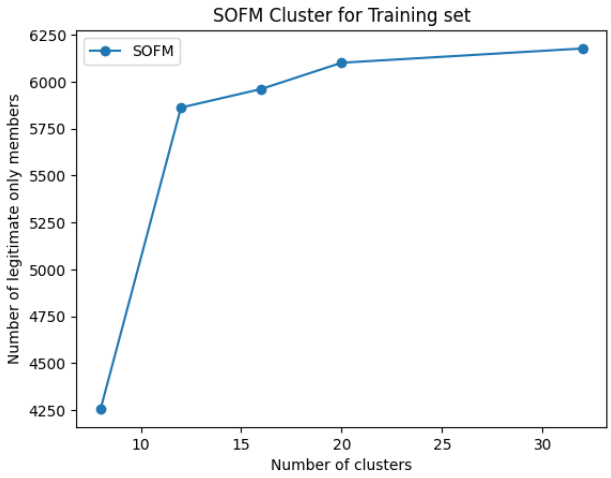
4- (a) Apply K-means algorithm to plot the number of clusters (8,12,16,20 and 32) vs the total number of legitimate only members inside the legitimate only clusters

For part 2 in question 4, we performed clustering on both training and test data set, as the professor told us.



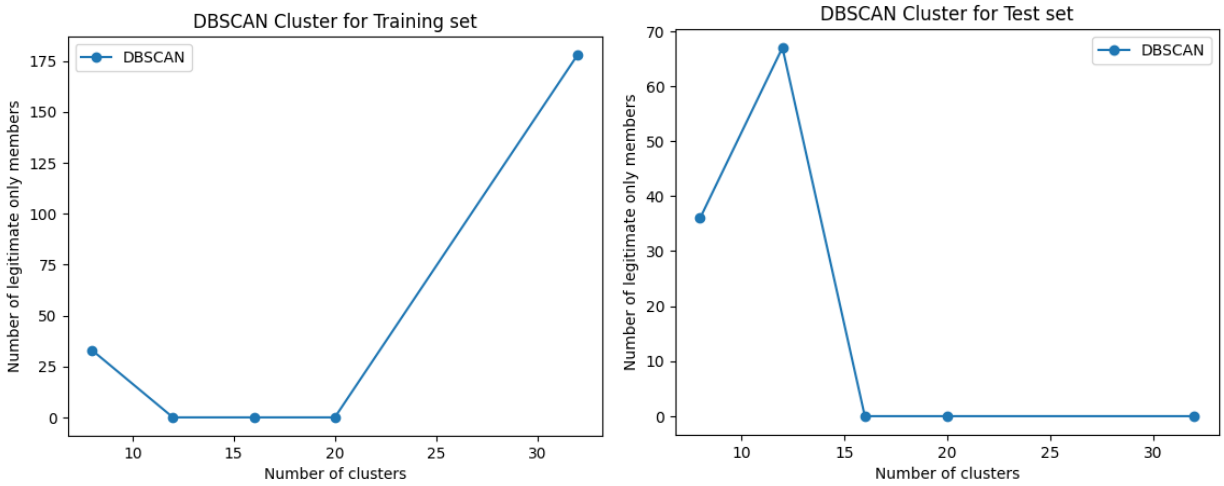


4- (b) Apply SOFM algorithm to plot the number of clusters (8,12,16,20 and 32) vs the total number of legitimate only members inside the legitimate only clusters.



(c) Apply DBSCAN algorithm to plot the number of clusters (8,12,16,20 and 32) vs the total number of legitimate only members inside the legitimate only clusters. You need to try different midPoint and epsilon parameters to obtain the 5 different cluster numbers. If you cannot obtain specific numbers you can report approximate numbers to 8,12,16,20 and 32

Note: As asked in Announcements yesterday, (fix one parameter, and change the other parameter), we fixed midpoint = 2 and use different epsilon values.



5) The Conclusion Parts

Conclusion of Q.1

This code loads the MCS dataset, splits it into training and test datasets based on the "Day" feature, and then separates the features and target variable for the training and test datasets. It also splits the remaining parts of the dataset into training and test datasets based on the "Day" feature, using a random seed of 0.

It then trains a Naive Bayes classifier and a K-Nearest Neighbor classifier on the training data, and makes predictions on the test data using both classifiers. It also calculates the confusion matrices and F1 scores for both classifiers F1 scores of NB 0.93 is better than of KNN 0.92 and plots the confusion matrices using `seaborn`'s heatmap function.

Finally, it generates 2D t-SNE plots for both the training and test sets using `sklearn.manifold`'s `TSNE` function and `seaborn`'s `scatterplot` function.

Conclusion of Q.2:

We used both Naive Bayes Classifier (NB) and K-Nearest Neighbor (KNN) to find the best values of n-components that result the best F1 scores when applying two Dimensionality Reduction methods PCA and Auto Encoder (AE).

As mentioned in **(A-1) cell** for **PCA** after we used both classifiers (NB, KNN) we found out that **the best value of n_components for NB is equal 1 and for KNN is equal 3**.

then we took these values of n-components to get the highest F1 scores of test dataset and we got the following:

For Naive Bayes the best value of F1 score when n-components=1 is 0.9322916666666667

For K-Nearest Neighbors F1 score when n-components=3 is 0.9297396913153652

Then we plotting the results against the baseline performance and found out that we got the highest performance when using PCA+NB than PCA+KNN.

****

And as mentioned in **(A-2) cell** for **Autoencoder (AE)**, after we applied both Naive Bayes Classifier (NB) and K-Nearest Neighbor (KNN) we found out that **for Naive Bayesthe the best value of hidden layer size is equal 9 and for K-Nearest Neighbors the Best value of hidden_layer_size is equal 8**

then we took these values of hidden_layer_sizes to get the highest F1 scores and we got the following resul

*For Naive Bayes the best value of F1 score when using hidden layer size = 9 is 0.9034400948991697

For K-Nearest Neighbors the best value of F1 score when using hidden layer size = 8 is 0.9273182957393483

Then we plotting the results against the baseline performance and found out that we got the highest performance when using AE+KNN than AE+NB.

Finally, we were plotting 2D TSNE based on the best performance (PCA+NB) for both training set and test set and observed that the data wasn't separated well.

Conclusion of Q.3:

For both Feature Selection methods, we used Naive Bayes Classifier (NB) and K-Nearest Neighbor (KNN) to find the best numbers of features.

when applying **filter methods with Information Gain** we found out **that the best value of num_features_range for Naive Bayes is equal 1 and the best value of num_features_range for K-Nearest Neighbors is equal 2**.then we took these values of features to get the highest F1 scores and we got the following results:

For Naive Bayes the best value of F1 score with 1 selected feature is 0.9322916666666667

For K-Nearest Neighbors the best value of F1 score with 2 selected features is 0.9833684703677676

Then we plotting the results against the baseline performance and found out that we got the highest performance when using filter Methods (Information Gain)

****

And when applying **Wrapper Methods with Forward Feature Elimination** we found out that **the best value of features_range for Naive Bayes is equal 5 and the best value of features_range for K-Nearest Neighbor is equal 6**. then we took these values of features to get the highest F1 scores and we got the following results:

For Naive Bayes the best value of F1 score with 5 selected features is 0.9322916666666667

For K-Nearest Neighbors the best value of F1 score with 6 selected features is 0.9584214235377027

Then we plotting the results against the baseline performance and found out that we got changeable performance when using Wrapper Methods (Forward Feature Elimination)

Conclusion of Q.4:

For both training and test dataset, we used three clustering algorithms, K-means, Self-Organizing Feature Map (SOFM), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), to cluster the latitude and longitude features of a dataset and find the best number of clusters among 8, 12, 16, 20, and 32 clusters that maximizes the number of legitimate only members inside the legitimate only clusters.

For K-means and SOFM algorithms, we tried different numbers of clusters and calculated the number of legitimate only members inside the legitimate only clusters for each number of clusters. Based on the plots, for training & test set, we found that the best number of clusters is 32 for both algorithms, which achieved the highest number of legitimate only members inside the legitimate only clusters.

For the DBSCAN algorithm, we tried different combinations of hyperparameters and calculated the number of legitimate only members inside the legitimate only clusters for each number of clusters. Based on the plot, we found that the best number of clusters is 12 for Training and 32 for Test set, which achieved the highest number of legitimate only members inside the legitimate only clusters.

To sum up, the results show that the K-means and SOFM algorithms perform best when 32 clusters are used in training & test set, while the DBSCAN algorithm performs best with 12 clusters in training and 32 in test set. The maximum number of legitimate only members in legitimate only clusters for 32 clusters is (4740 training, 1713 test set) for K-means, and (6182 training, 2130 test) for SOFM, while the maximum number of legitimate only members in legitimate only clusters for (32 clusters is 178 for training set), and (12 cluster is 67) for test set for DBSCAN.

****