Applied Machine Learning

Mohamed Salah
Ibrahim Elshenhapy
Ahmed Shafik

July 2023

1 Use the k-means algorithm and Euclidean distance to cluster the following 5 data points into 2 clusters: A1=(3,6), A2=(6,3), A3=(8,6), A4=(2,1), A5=(5,9). Suppose that the initial centroids (centers of each cluster) are A2 and A4. Using k-means, cluster the 5 points and show the followings for one iteration only:

1.1 (a) Show step-by-step the performed calculations to cluster the 5 points.

Solution

The Euclidean distance between two points in a two-dimensional space is given by:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

for A1: (distance between A1 and A2)

$$d = \sqrt{(6-3)^2 + (3-6)^2} = \sqrt{18}$$

(distance between A1 and A4)

$$d = \sqrt{(2-3)^2 + (1-6)^2} = \sqrt{26}$$

for A2: (distance between A2 and A2)

$$d = 0$$

(distance between A2 and A4)

$$d = \sqrt{(6-2)^2 + (3-1)^2} = \sqrt{20}$$

for A3: (distance between A3 and A2)

$$d = \sqrt{(8-6)^2 + (6-3)^2} = \sqrt{13}$$

(distance between A3 and A4)

$$d = \sqrt{(8-2)^2 + (6-1)^2} = \sqrt{61}$$

for A4: (distance between A4 and A2)

$$d = \sqrt{(2-6)^2 + (1-3)^2} = \sqrt{20}$$

(distance between A4 and A4)

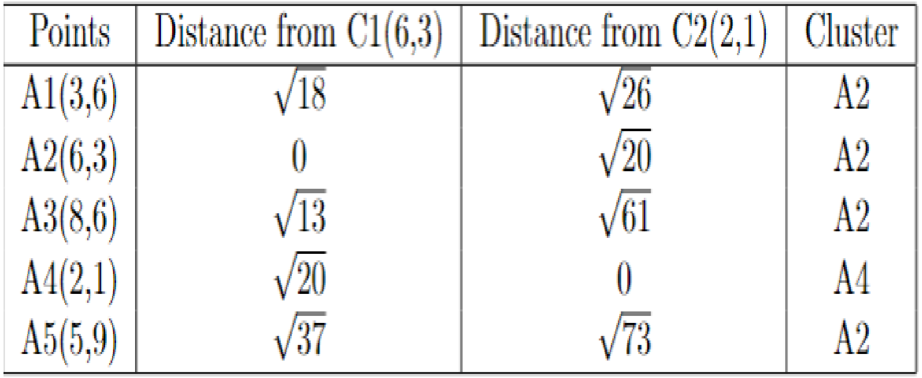
$$d = 0$$

for A5: (distance between A5 and A2)

$$d = \sqrt{(5-6)^2 + (9-3)^2} = \sqrt{37}$$

(distance between A5 and A4)

$$d = \sqrt{(5-2)^2 + (9-1)^2} = \sqrt{73}$$



Points	Distance from C1(6,3)	Distance from C2(2,1)	Cluster
A1(3,6)	$\sqrt{18}$	$\sqrt{26}$	A2
A2(6,3)	0	$\sqrt{20}$	A2
A3(8,6)	$\sqrt{13}$	$\sqrt{61}$	A2
A4(2,1)	$\sqrt{20}$	0	A4
A5(5,9)	$\sqrt{37}$	$\sqrt{73}$	A2

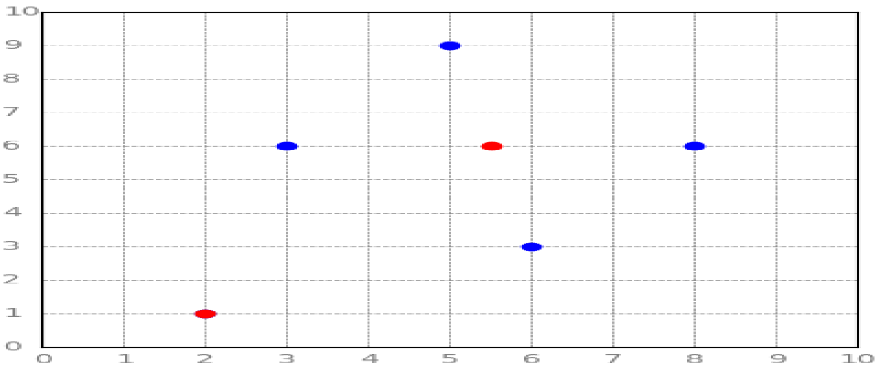
The new centroids:

c1'=($\frac{3+6+8+5}{4}$, $\frac{6+3+6+9}{4}$) =(5.5 , 6)

c2'=(2, 1)

1.2 (b) Draw a 10 by 10 space with all the clustered 5 points and the coordinates of the new centroids.

Solution



1.3 (c) Calculate the silhouette score and WSS score.

Solution

The distances between the points and the new centroids:

Points	A1(3,6)	A2(6,3)	A3(8,6)	A4(2,1)	A5(5,9)	new C1(5.5,6)	new C2(2,1)
A1(3,6)	0	4.24	5	5.1	3.61	2.5	5.1
A2(6,3)	4.24	0	3.61	4.47	6.1	3.04	4.47
A3(8,6)	5	3.61	0	7.81	4.24	2.5	7.81
A4(2,1)	5.1	4.47	7.81	0	8.54	6.1	0
A5(5,9)	3.61	6.1	4.24	8.54	0	3.04	8.54

Silhouette:

$$silhouette\_score = \frac{b-a}{\max(a,b)}$$

for A1:

$$b(A1) = 5.1$$

$$a(A1) = \frac{4.24+3.61+5}{3} = 4.28$$

for A2:

$$b(A2) = 4.47$$

$$a(A2) = \frac{4.24+3.61+6.1}{3} = 4.65$$

for A3:

$$b(A3) = 7.81$$

$$a(A3) = \frac{4.24+3.61+5}{3} = 4.28$$

for A5:

$$b(A5) = 8.54$$

$$a(A5) = \frac{3.61 + 6.1 + 4.24}{3} = 4.65$$

silhouette Score:

$$S(A1) = \frac{5.1 - 4.28}{5.1} = 0.16$$

$$S(A2) = \frac{4.47 - 4.65}{4.65} = -0.039$$

$$S(A3) = \frac{7.81 - 4.65}{7.81} = 0.405$$

$$S(A4) = \frac{6.48 - 0}{6.48} = 1$$

$$S(A5) = \frac{8.54 - 4.65}{8.54} = 0.456$$

$$AverageSilhouetteScore = \frac{\sum_{i=1}^{n} silhouettescore_i}{n}$$

$$AverageSilhouetteScore = \frac{0.16 + 0.405 + 0.456 + 1 - 0.039}{5} = 0.3964$$

$$WSS = \sum \|x - c_i\|^2$$

$$WSS = 2.5^2 + 2.5^2 + 3.04^2 + 3.04^2 + 0 = 31$$

Group 12 HW1 (Part 2)

Contents

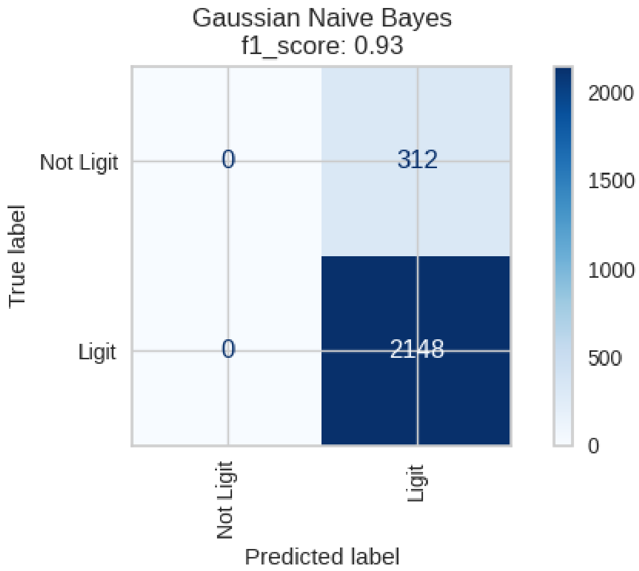
1- Naive Bayes Classifier (NB) and K-Nearest Neighbor (KNN) classifiers on the provided Mobile Crowd Sensing (MCS) dataset

a) Naive Bayes Classifier (NB) and K-Nearest Neighbor (KNN) classifiers on the provided Mobile Crowd Sensing (MCS) dataset

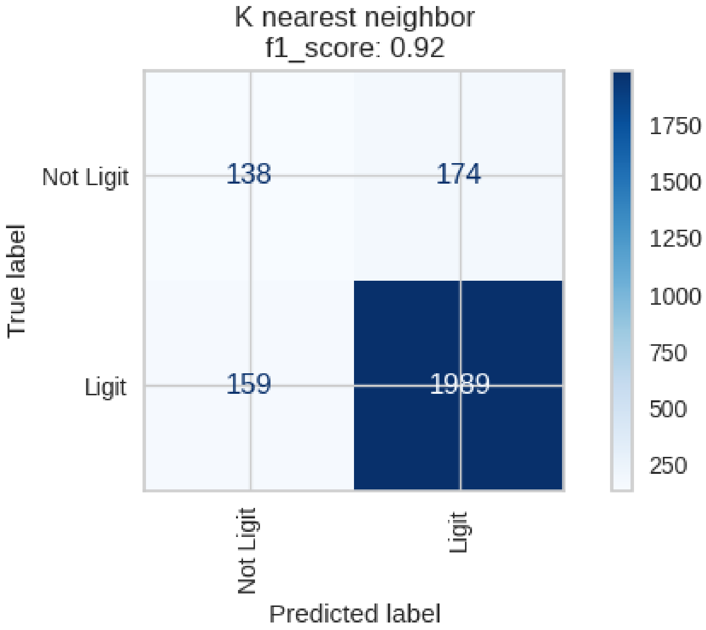
```
train_data = data.loc[data.Day.isin([0, 1, 2])]
test_data = data[data.Day == 3]
train_data = train_data.drop(['ID','Day'],axis = 1)
test_data = test_data.drop(['ID','Day'],axis = 1)
x_train = train_data.drop('Ligitimacy',axis = 1)
y_train = train_data.Ligitimacy
x_test = test_data.drop('Ligitimacy',axis = 1)
y_test = test_data.Ligitimacy
```

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

F1 score for Gaussian Naive Bayes : 0.9322916666666667

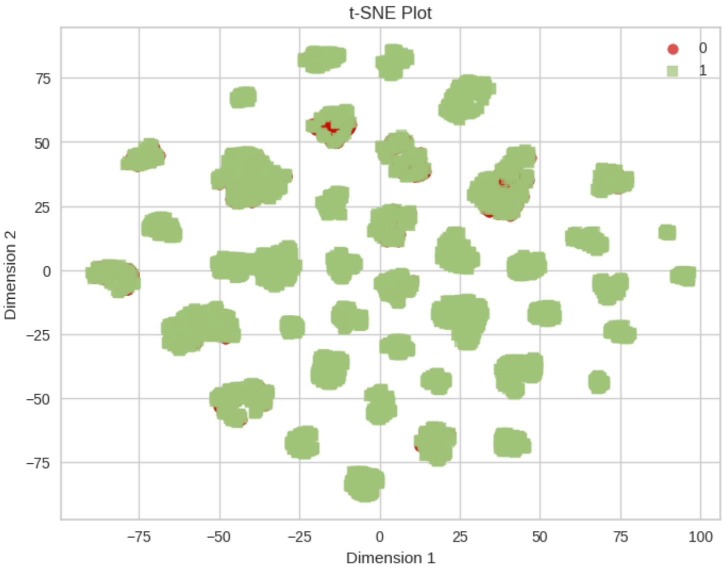


F1 score for KNN: 0.9227557411273487

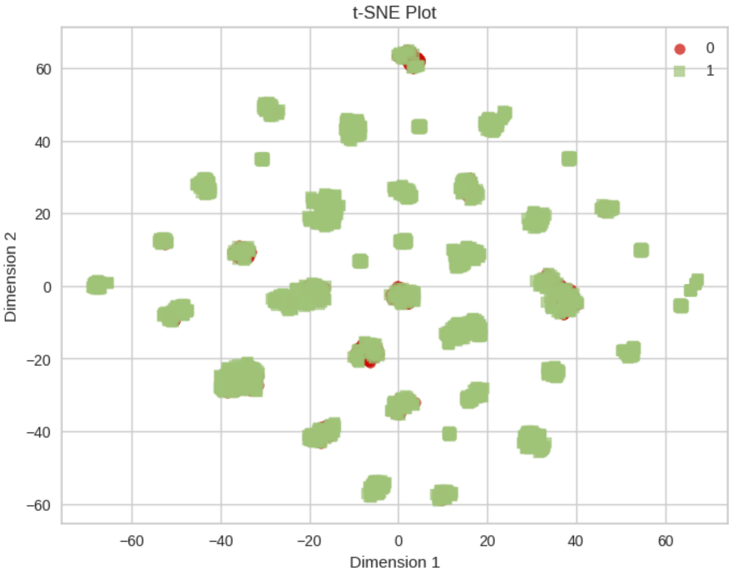


c) Provide 2D TSNE plots, one for the training set and one for the test set.

TSNE for train data



TSNE for testdata



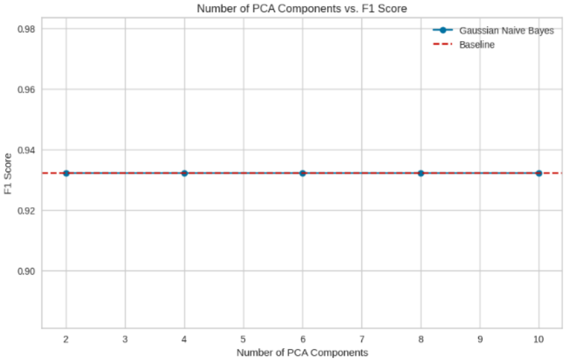
2- Apply the following Dimensionality Reduction (DR) methods: PCA and Auto Encoder (AE).

a) Find the best reduced dimensions of PCA and AE based on f1 score of test dataset using both classifiers (NB and KNN)

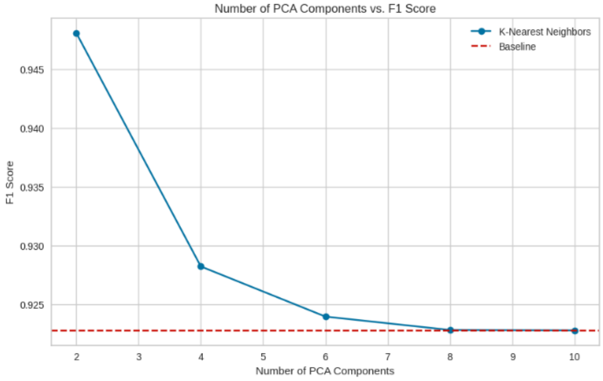
PCA

```
Best n_components for Naive Bayes: 2
Best F1 score for Naive Bayes: 0.9322916666666667
Best n_components for KNN: 2
Best F1 score for KNN: 0.9481165600568585
```

Gauss NB



KNN



Auto Encoder

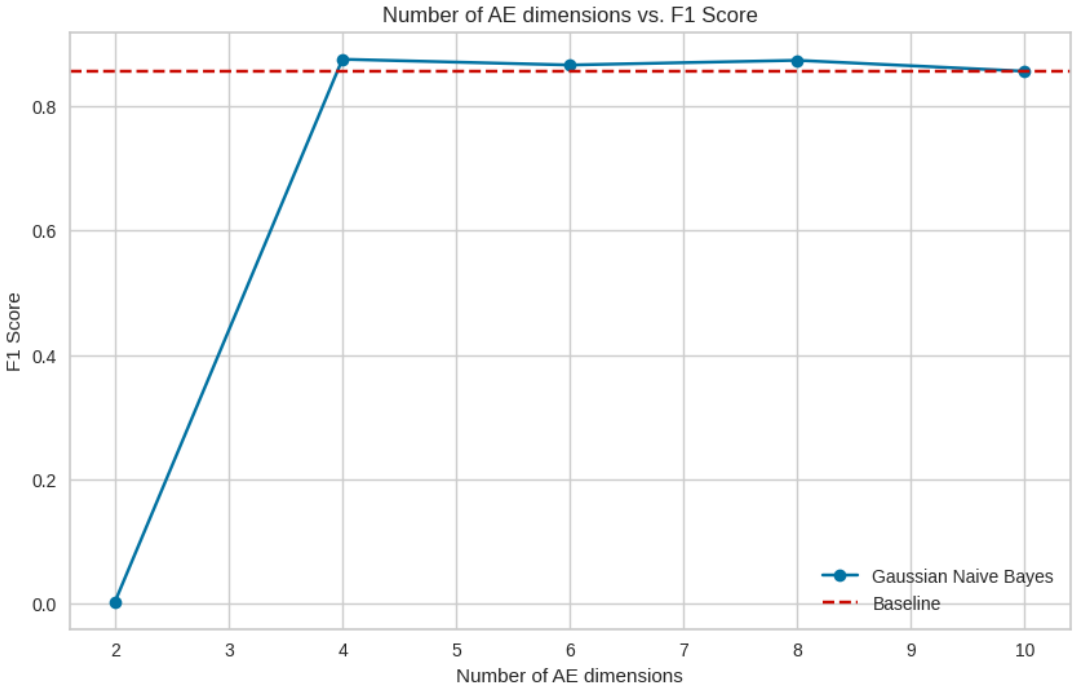
Best F1 score for Naive Bayes: 0.8752455795677799

Number of features for best F1 score (Naive Bayes): 4

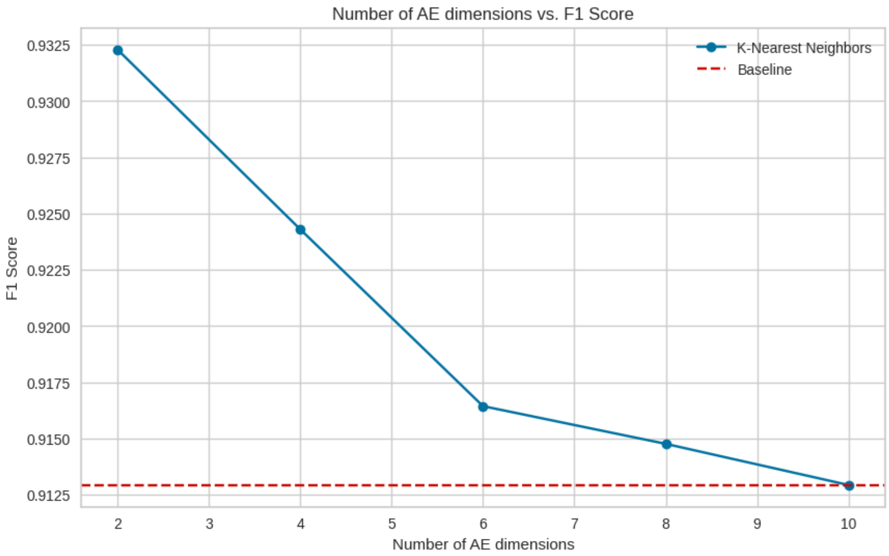
Best F1 score for KNN: 0.9322916666666667

Number of features for best F1 score (KNN): 2

Gauss naïve bayes

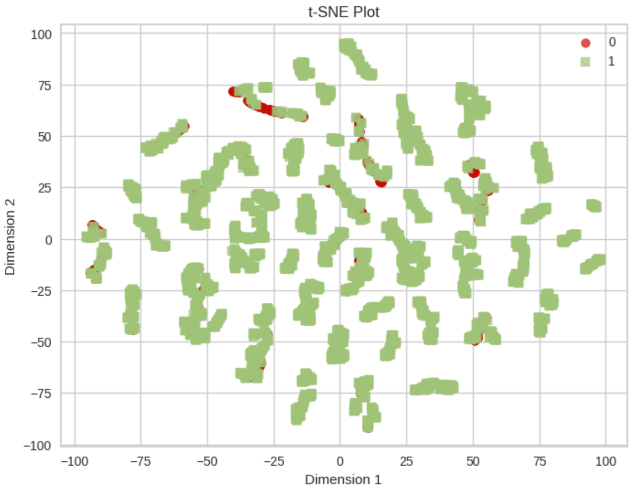


KNN

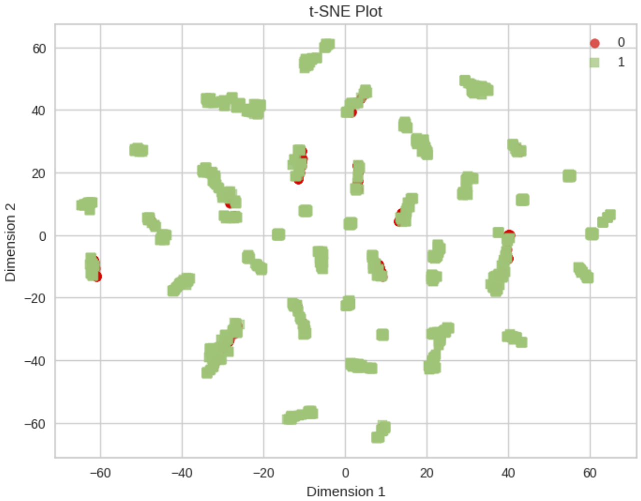


b) Provide 2D TSNE plots for the best performance in previous part

One for the training set



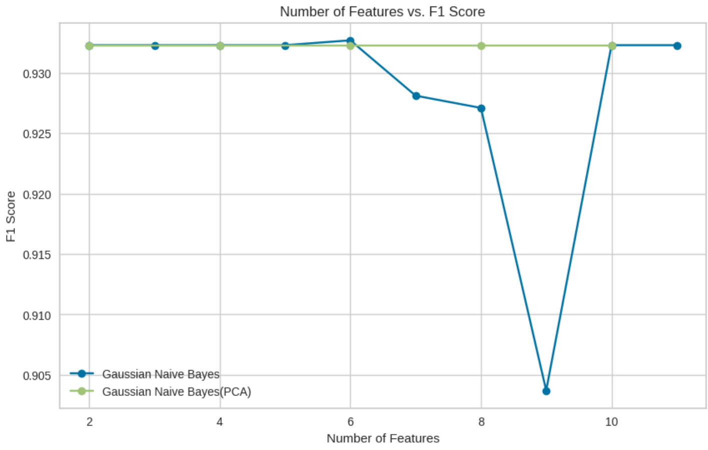
One for the test set.



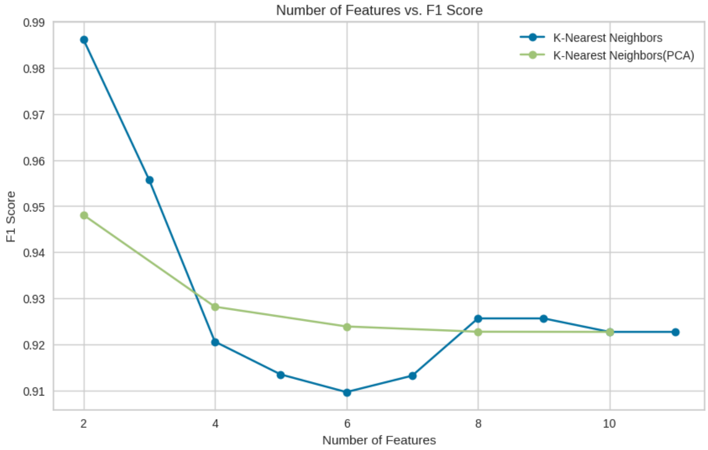
3- Use the Filter and wrapper Feature Selection methods.

a) Filter Methods. Plot the number of features vs f1 score with the improved baseline performance.

Gauss Naïve Bayes

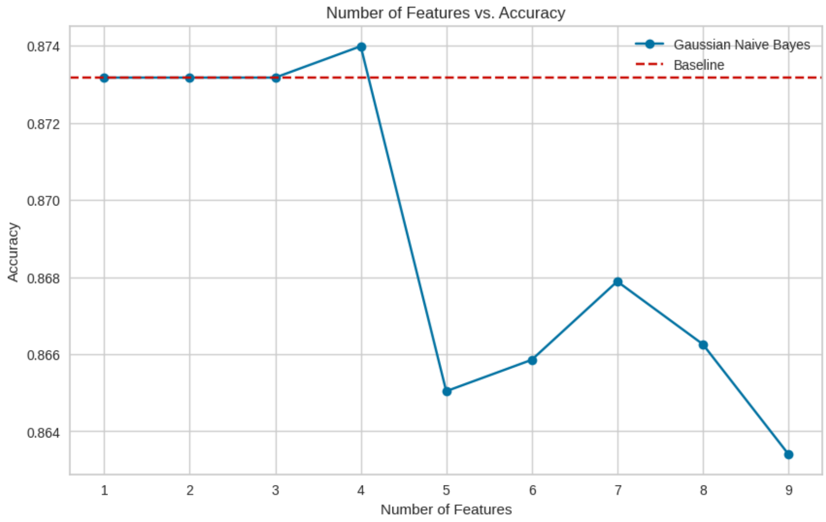


KNN

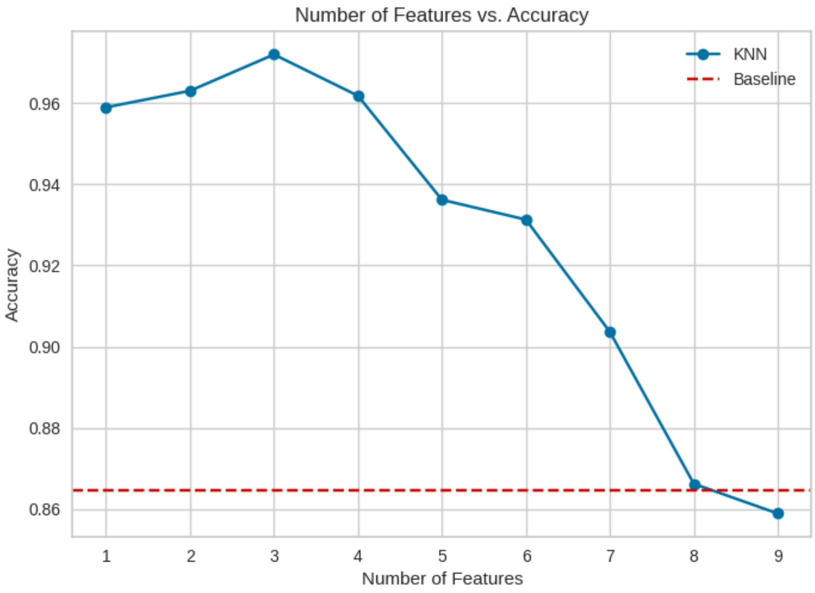


b) Wrapper Methods (Forward or Backward Feature Elimination, Recursive Feature Elimination etc.). Plot the number of features versus accuracy graph with the baseline performance.

Gauss Naïve Bayes



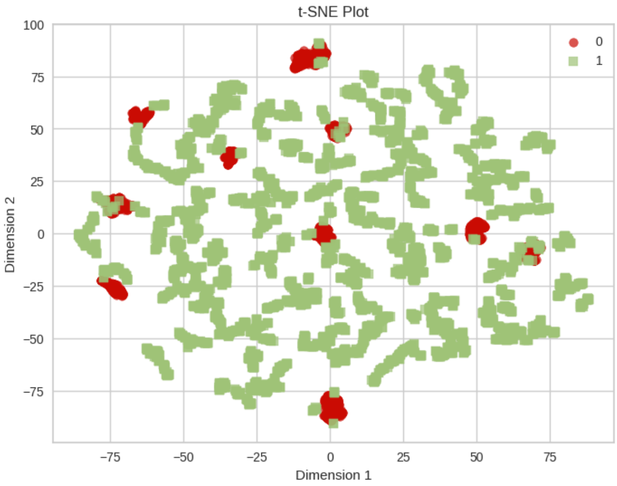
KNN



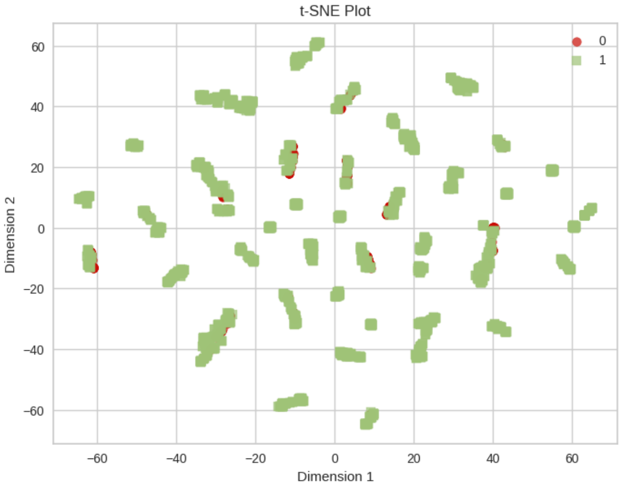
c) Provide 2D TSNE plots, using only the best method (either the filter or wrapper).

The Best Method is filter Method

One for the train set.

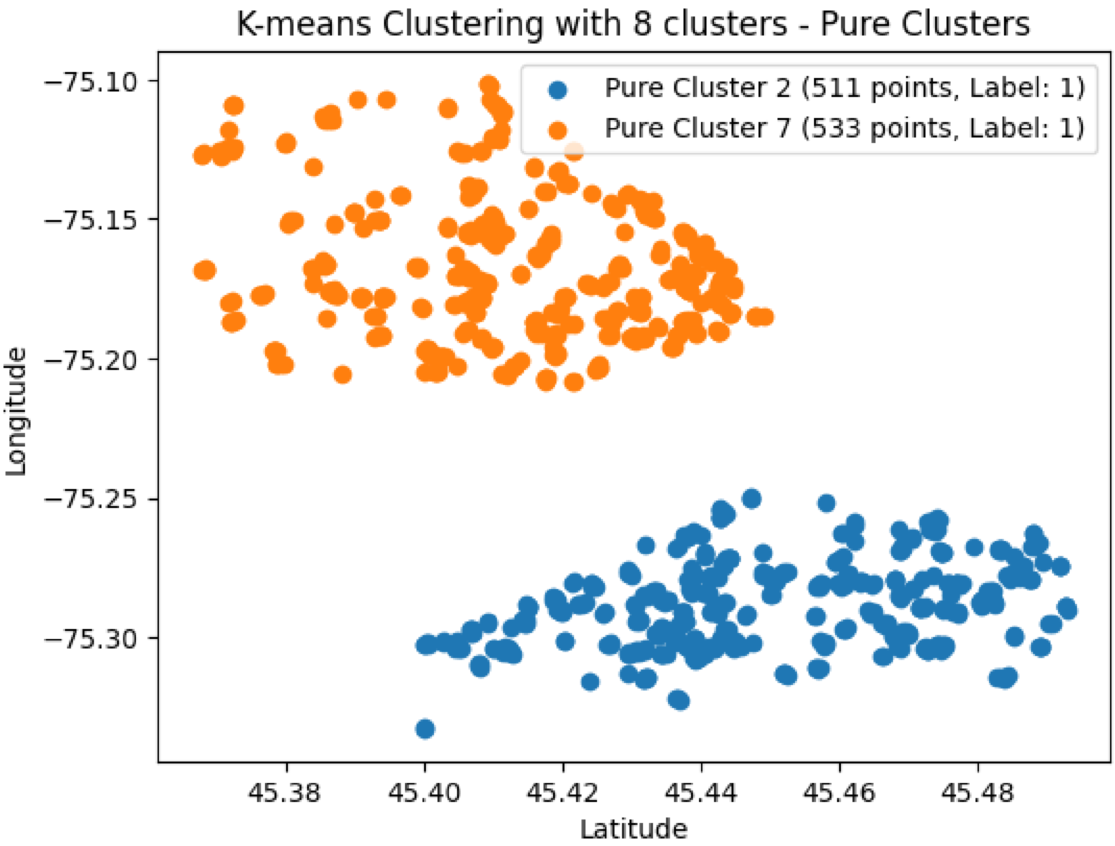


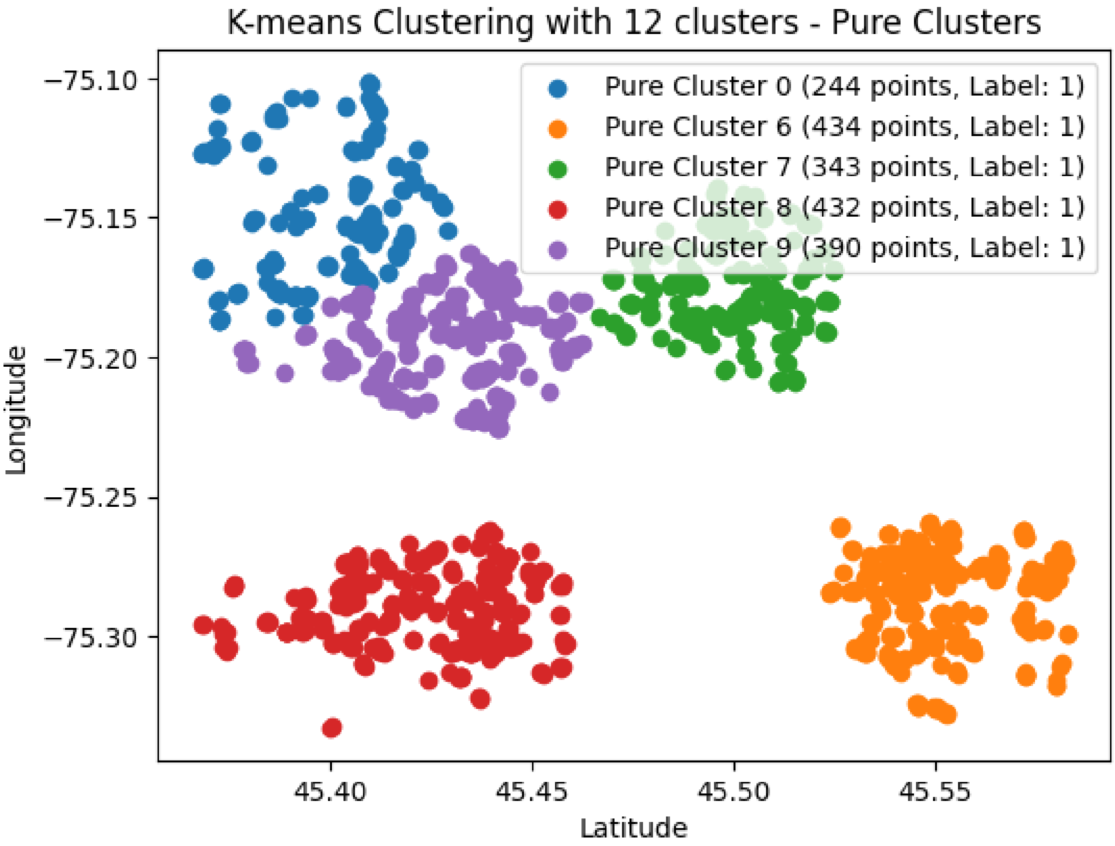
One for the test set.

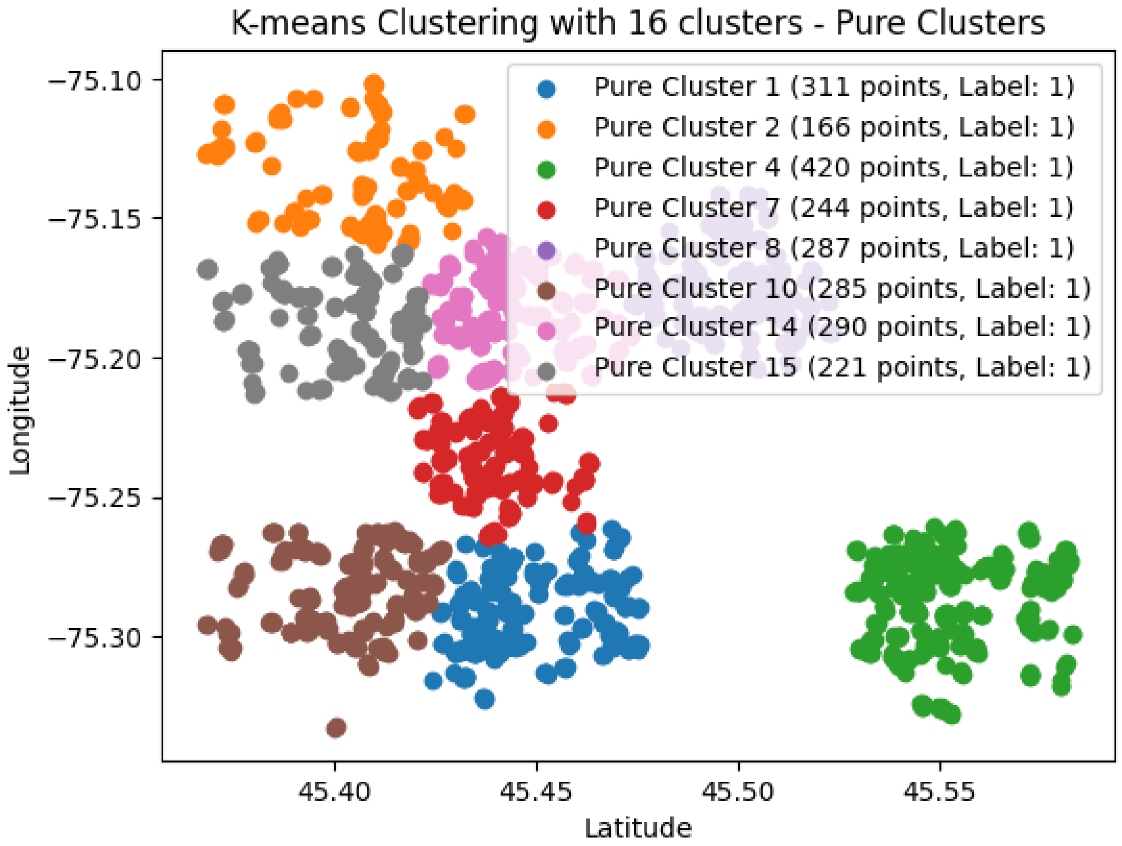


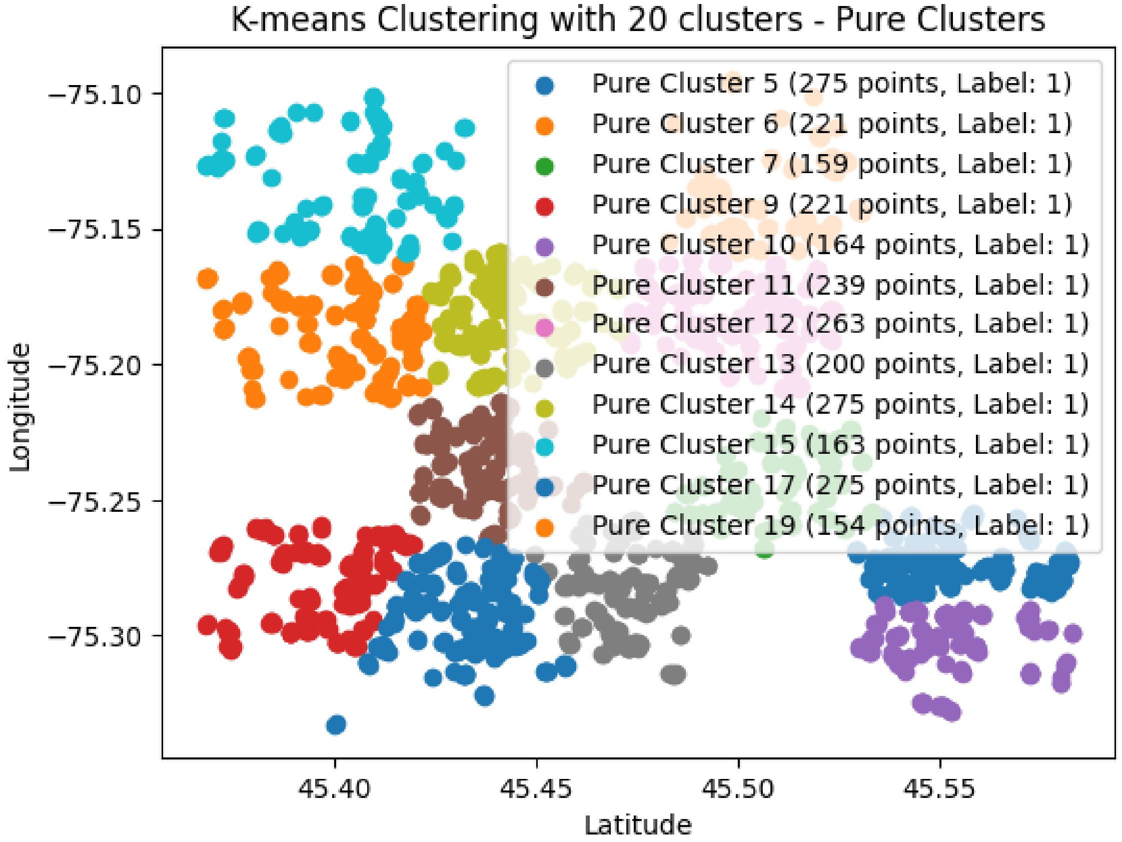
4- Latitude and longitude features should be considered for clustering based methods.

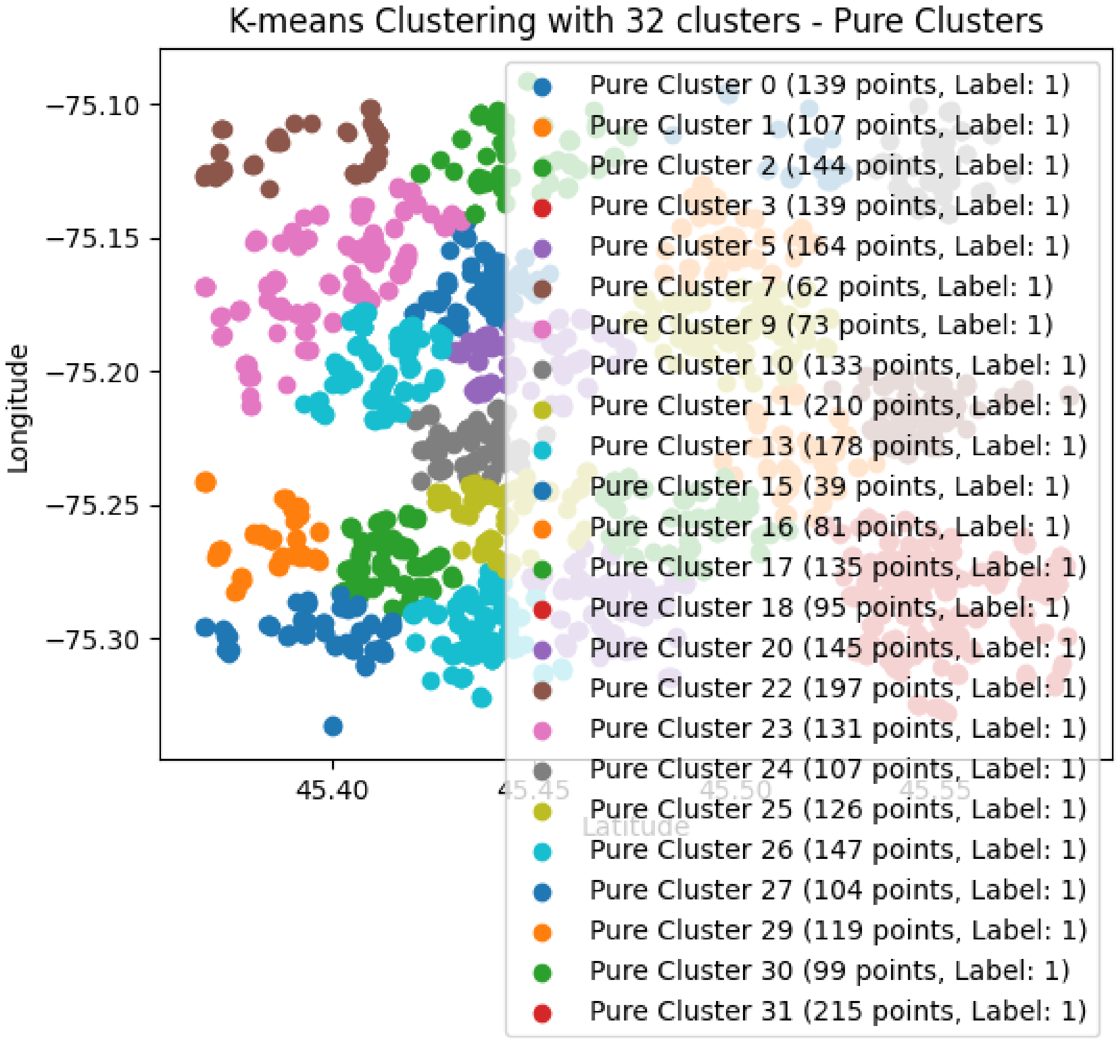
a) Apply K-means algorithm to plot the number of clusters (8,12,16,20 and 32) vs the total number of legitimate only members inside the legitimate only clusters.


K-means Clustering with 8 clusters - Pure Clusters

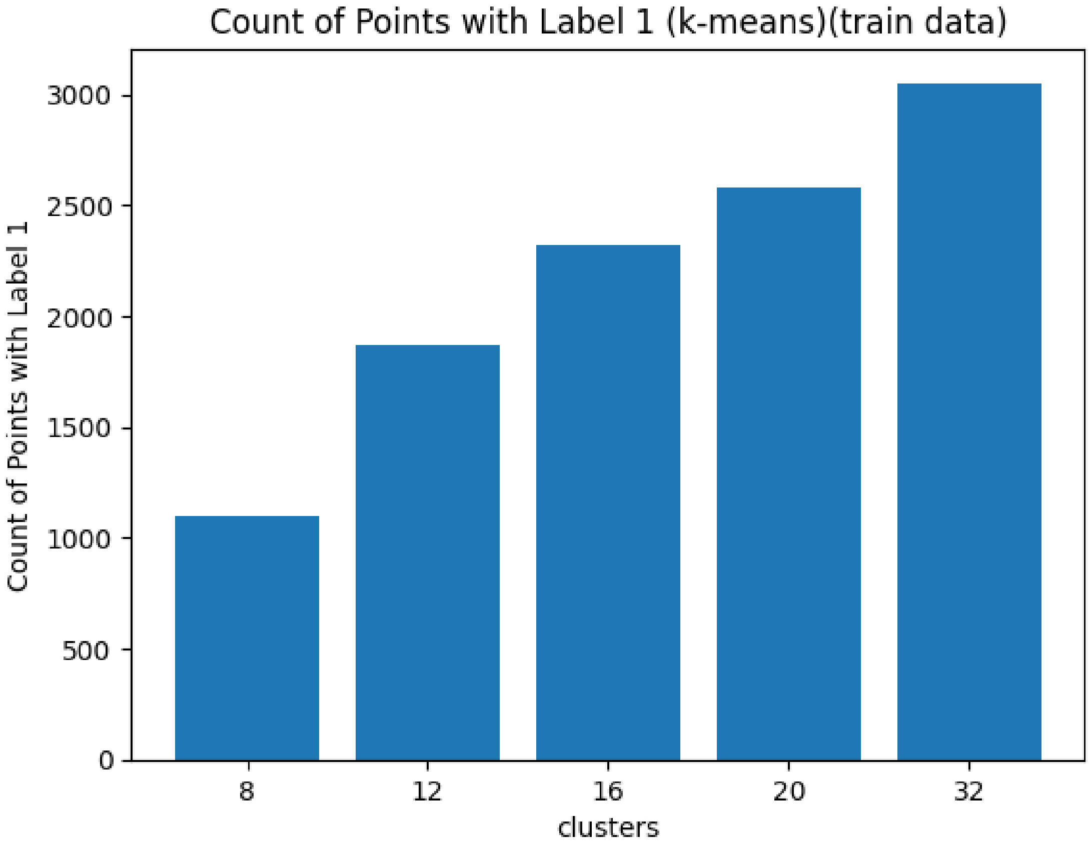

K-means Clustering with 12 clusters - Pure Clusters


K-means Clustering with 16 clusters - Pure Clusters
Pure Cluster 1 (311 points, Label: 1)
Pure Cluster 2 (166 points, Label: 1)
Pure Cluster 4 (420 points, Label: 1)
Pure Cluster 7 (244 points, Label: 1)
Pure Cluster 8 (287 points, Label: 1)
Pure Cluster 10 (285 points, Label: 1)
Pure Cluster 14 (290 points, Label: 1)
Pure Cluster 15 (221 points, Label: 1)
−75.10
−75.15
−75.20
−75.25
−75.30
Longitude
45.40
45.45
45.50
45.55
Latitude

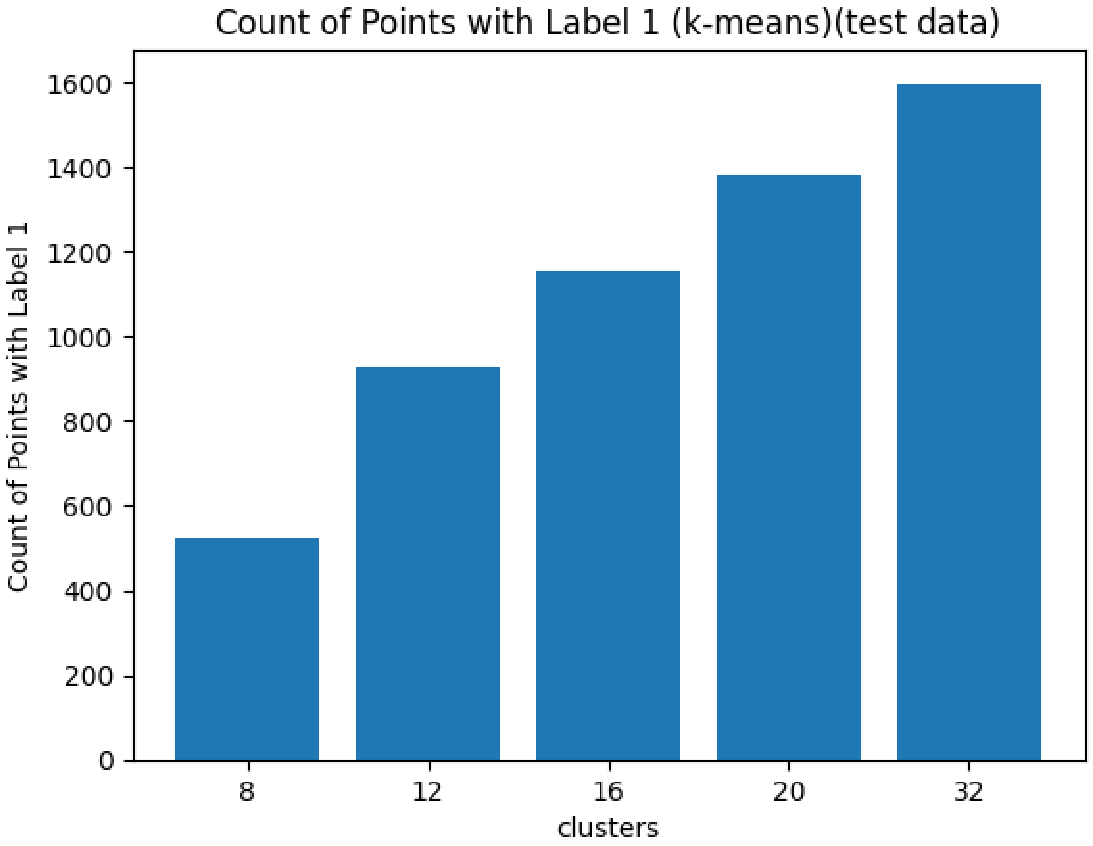

K-means Clustering with 20 clusters - Pure Clusters
Pure Cluster 5 (275 points, Label: 1)
Pure Cluster 6 (221 points, Label: 1)
Pure Cluster 7 (159 points, Label: 1)
Pure Cluster 9 (221 points, Label: 1)
Pure Cluster 10 (164 points, Label: 1)
Pure Cluster 11 (239 points, Label: 1)
Pure Cluster 12 (263 points, Label: 1)
Pure Cluster 13 (200 points, Label: 1)
Pure Cluster 14 (275 points, Label: 1)
Pure Cluster 15 (163 points, Label: 1)
Pure Cluster 17 (275 points, Label: 1)
Pure Cluster 19 (154 points, Label: 1)
−75.10
−75.15
−75.20
−75.25
−75.30
Longitude
45.40
45.45
45.50
45.55
Latitude



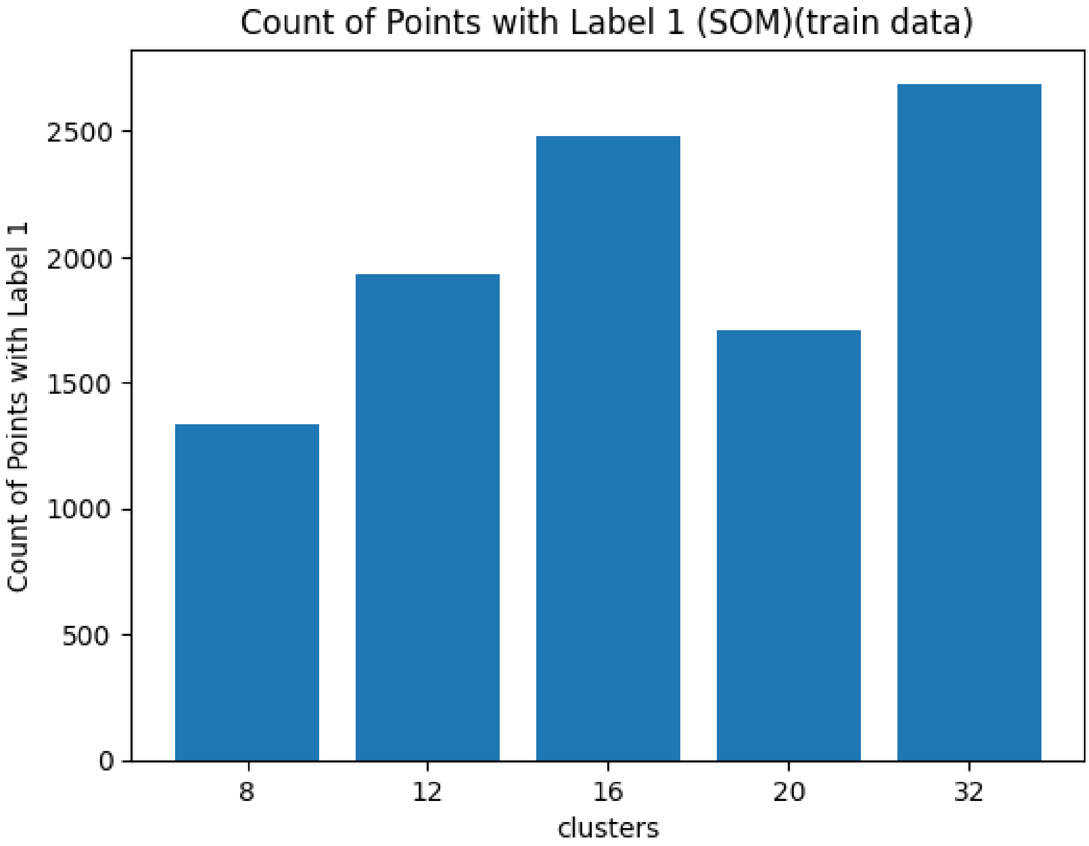
K-means Clustering with 32 clusters - Pure Clusters

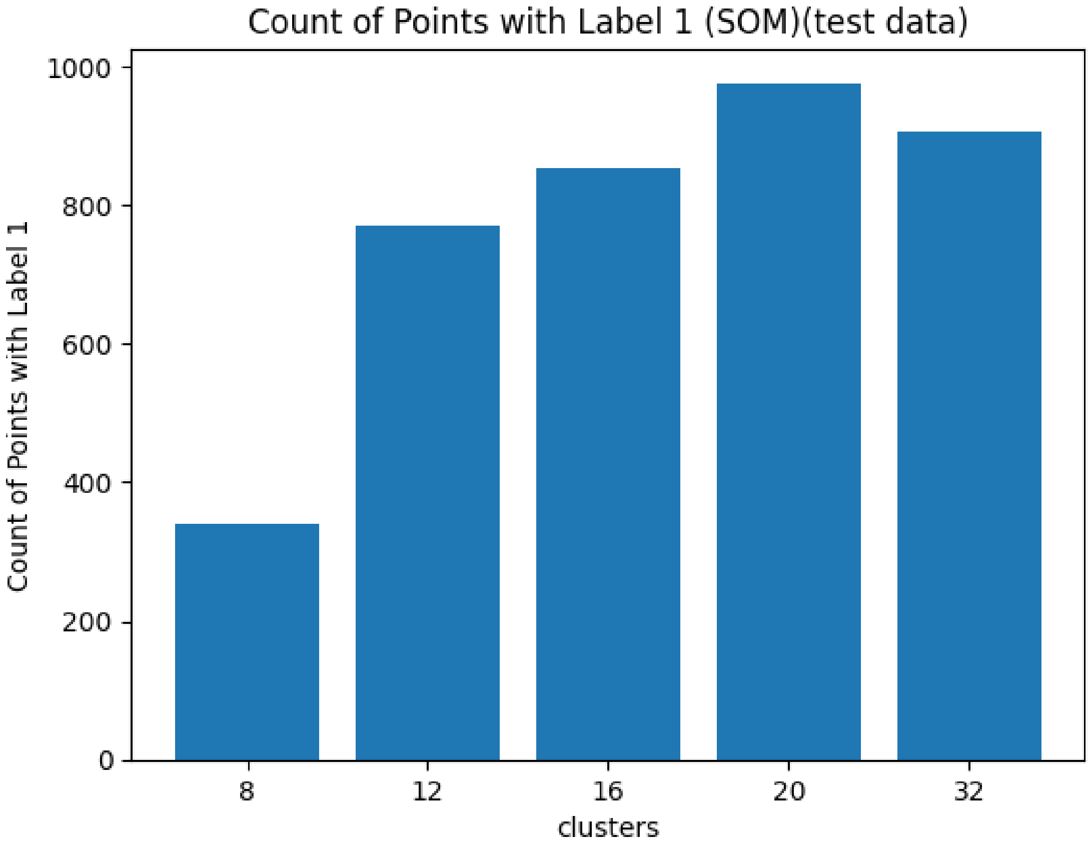


Count of Points with Label 1 (k-means)(train data)

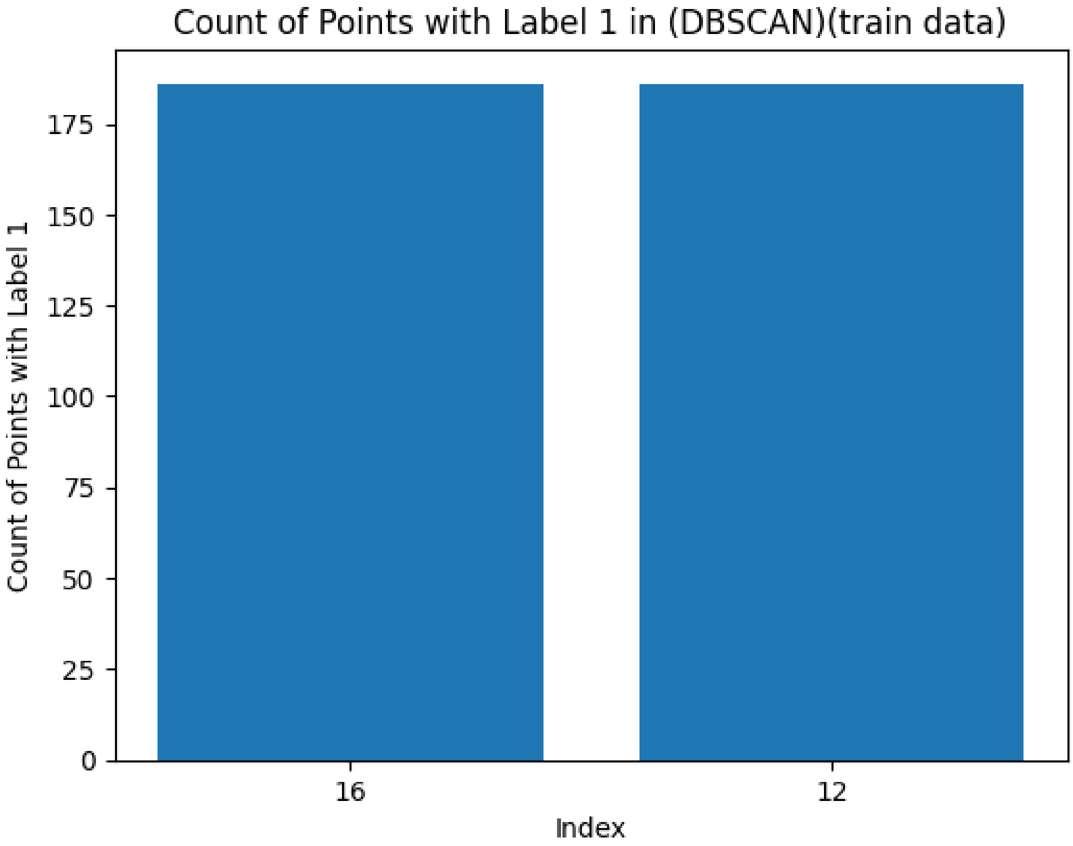

Count of Points with Label 1 (k-means)(test data)
Count of Points with Label 1
0
200
400
600
800
1000
1200
1400
1600
8
12
16
20
32
clusters

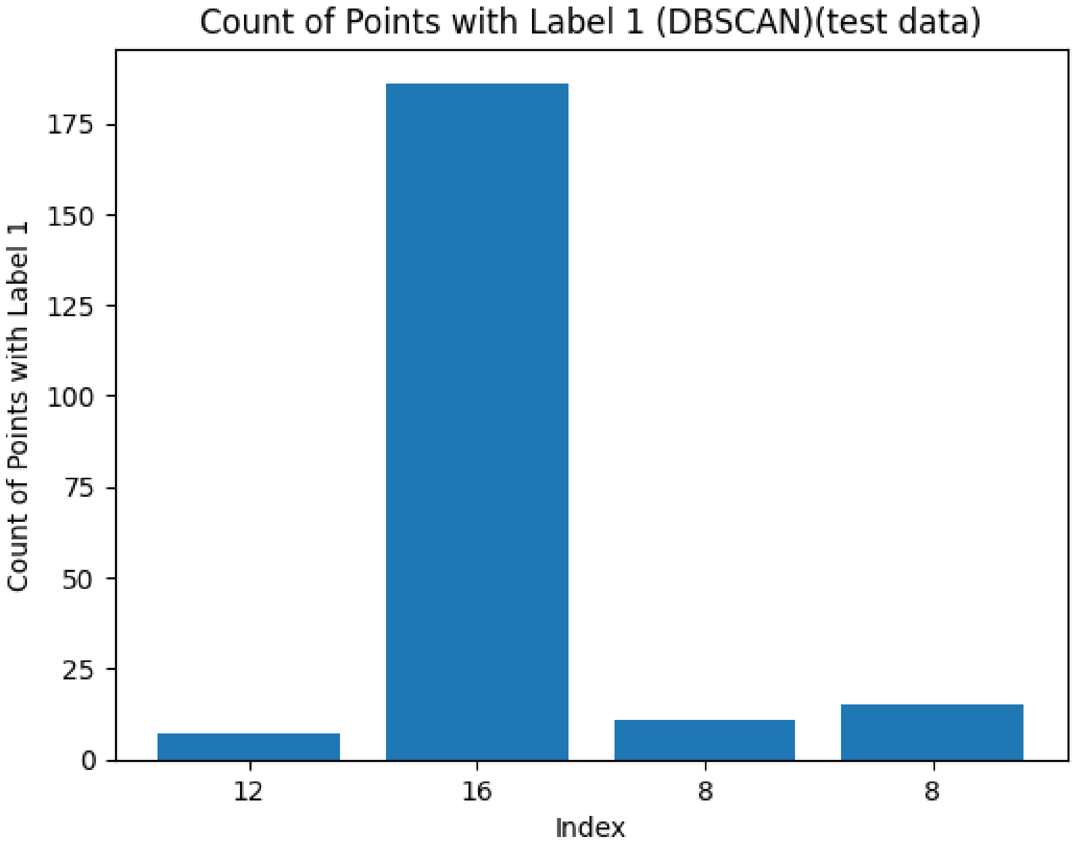
b) Apply SOFM algorithm to plot the number of clusters (8,12,16,20 and 32) vs the total number of legitimate only members inside the legitimate only clusters.





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.





5- The conclusion.

Problem 1 conclusions

- After processing the data according to the instructions, it was observed that the dataset had a small proportion of fake values (marked as '0') in the "Legitimacy" feature. As a result, the Gaussian Naive Bayes classifier classified all elements as legitimate ('1'), while the K-Nearest Neighbor (KNN) classifier showed better performance by correctly identifying some fake instances, although it also made some false predictions.
- The confusion matrices for both classifiers confirmed these findings, with the Gaussian Naive Bayes classifier having high precision but low recall for the "fake" class, while the KNN classifier exhibited a more balanced performance.
- The high F1 scores obtained for both classifiers were primarily driven by their ability to correctly predict the majority class (legitimate instances), rather than their effectiveness in detecting the minority class (fake instances). Therefore, the F1 score alone may not be a reliable measure of overall classifier performance in this imbalanced dataset scenario.

Problem 2 conclusions

- For the PCA dimensionality reduction method:

When using the Gaussian Naive Bayes classifier, the F1 score of the test dataset did not change with varying numbers of PCA components. For the K-Nearest Neighbor (KNN) classifier, the highest F1 score was achieved when the number of PCA components was 2.

- For the Autoencoder (AE) dimensionality reduction method:

With the Gaussian Naive Bayes classifier, the best performance was obtained with 4 AE components. As the number of components increased, the performance slightly changed, but the optimal choice remained at 4 components. For the KNN classifier, the highest F1 score occurred when the number of AE components was 2, similar to the PCA performance and as the number of components increased further, the F1 score started to decrease.

- It was observed that the highest F1 score was obtained when using PCA with 2 components. This suggests that PCA was able to capture the relevant information in the dataset more effectively compared to the Autoencoder.

Problem 3 conclusions

- For the Filter method:

Using the Gaussian Naive Bayes classifier, the best F1 score was achieved with 6 features. The F1 score slightly increased compared to 2 and 4 features, but then started to decrease until reaching its lowest value at 9 features. For the K-Nearest Neighbor (KNN) classifier, the best F1 score occurred with 2 features, which differed from the optimal number of features for the Gaussian Naive Bayes model.

- For the Wrapper method:

When employing the Gaussian Naive Bayes classifier, the highest F1 score was obtained with 4 features. However, it was still lower than the F1 score achieved by the Filter method. For the KNN classifier, the best F1 score was obtained with 3 features, but it slightly trailed behind the results obtained from the Filter method.

- Based on these observations, it is evident that the Filter method outperformed the Wrapper method in terms of F1 scores for both the Gaussian Naive Bayes and K-Nearest Neighbor classifiers. Additionally, the KNN classifier yielded better results in both methods compared to the Gaussian Naive Bayes model.
- The Filter method proved to be more effective in selecting the optimal number of features for both classifiers, resulting in improved F1 scores.

Problem 4 conclusions

- For the Kmeans:

The K-means clustering algorithm was applied to the test dataset using different numbers of clusters (n_clusters). The resulting counts of points with label 1 were stored in the c_num list. The analysis showed that the count of points with label 1 varied based on the number of clusters used, indicating the impact of cluster choice on identifying points with label 1 in both training and test datasets. Therefore, it is crucial to consider the number of clusters as a parameter in K-means clustering to accurately identify points with specific labels.

- For the SOM:

The Self-Organizing Maps (SOM) algorithm was used to cluster the given dataset with different numbers of clusters (n_clusters). The counts of points with label 1 in the resulting clusters were calculated and stored in the c_num list. The analysis showed that the counts of points with label 1 varied based on the number of clusters used. The c_num list contained arrays representing the count of points with label 1 for each specific number of clusters. The bar chart effectively visualized the distribution of these counts across different numbers of clusters.

- For the DBSCAN:

The DBSCAN clustering algorithm was utilized with different combinations of epsilon (eps) and minimum samples (min_samples) on the given dataset. The resulting clusters consistently contained 186 points with label 1 for all specified combinations of eps and min_samples. The counts of points with label 1 for each combination were stored in the c_num list. The findings highlight the effectiveness of the chosen combinations of eps and min_samples in accurately detecting and grouping a specific group of points with label 1 using DBSCAN clustering. This demonstrates the algorithm's capability to successfully identify and cluster points with a particular label.