ELG5255 Applied Machine Learning

REPORT of: Group Assignment 3 (Group-18)

Part 1: Calculations

Use the k-means algorithm and Euclidean distance to cluster the following 5 data points into 2 clusters: A1= (2,5), A2= (5,8), A3= (7,5), A4= (1,2), A5= (4,9). 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:

(a) Show step-by-step the performed calculations to cluster the 5 points.
(b) Draw a 10 by 10 space with all the clustered 5 points and the coordinates of the new centroids
(c) Calculate the silhouette score and WSS score.

Solution:

a) Suppose d (a, b) denotes the Euclidean distance between a and b.
It is obtained directly from the distance matrix or calculated as follows:

- d (a, b) = sqrt ((X_b - X_a)^2 + (Y_b - Y_a)^2)

One iteration-start:

A	Data	Centroid A2(5, 8)	Centroid A4(1, 2)
1	(2, 5)	d (A1, A2) = $\sqrt{(2-5)^2+(5-8)^2}$ = **4.24**	d (A1, A4) = $\sqrt{(2-1)^2+(5-2)^2}$ = **3.16**
2	(5, 8)	d (A2, A2) = $\sqrt{(5-5)^2+(8-8)^2}$ = **0**	d (A2, A4) = $\sqrt{(5-1)^2+(8-2)^2}$ = **7.21**
3	(7, 5)	d (A3, A2) = $\sqrt{(7-5)^2+(5-8)^2}$ = **3.16**	d (A3, A4) = $\sqrt{(7-1)^2+(5-2)^2}$ = **6.71**
4	(1, 2)	d (A4, A2) = $\sqrt{(1-5)^2+(2-8)^2}$ = **7.21**	d (A4, A4) = $\sqrt{(1-1)^2+(2-2)^2}$ = **0**

5	(4, 9)	d (A5, A2) = $\sqrt{(4-5)^2+(9-8)^2}$ = **1.41**	d (A5, A4) = $\sqrt{(4-1)^2+(9-2)^2}$ = **7.62**

One iteration-end

In Centroid A2 (5, 8) € Cluster1 (C1), A4 (1, 2) € Cluster2 (C2):

d (A1, A2) = 4.24, d (A1, A4) = 3.16
d (A1, A4) is less than d (A1, A2)
A1 € C2

d (A2, A2) = 0, d (A2, A4) = 7.21
d (A2, A2) is less than d (A2, A4)
A2 € C1

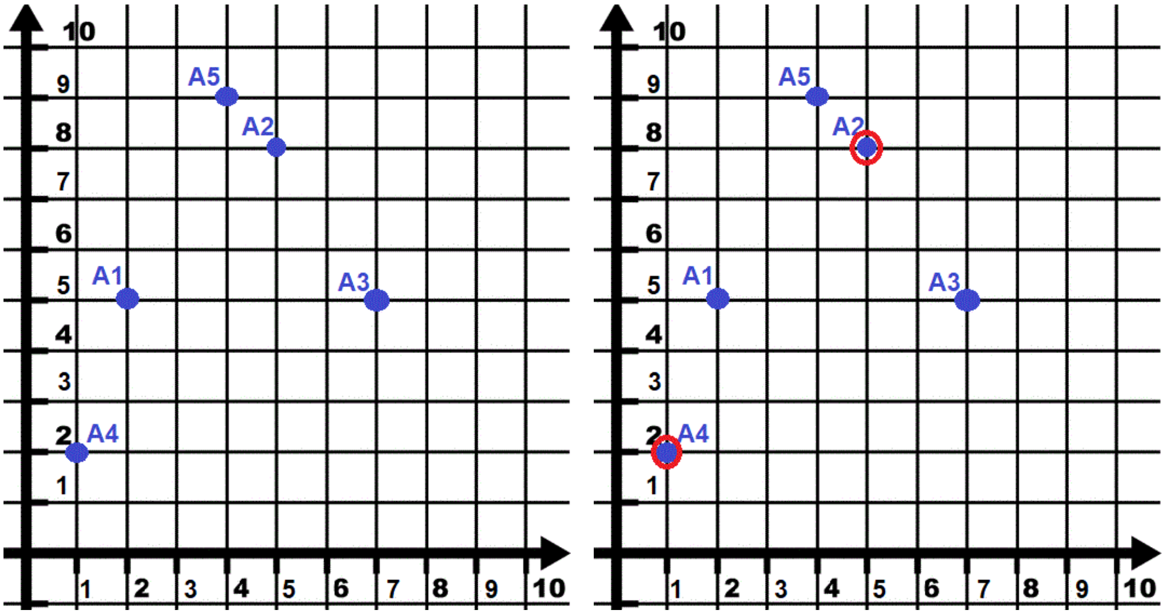
d (A3, A2) = 3.16, d (A3, A4) = 6.71
d (A3, A2) is less than d (A3, A4)
A3 € C1

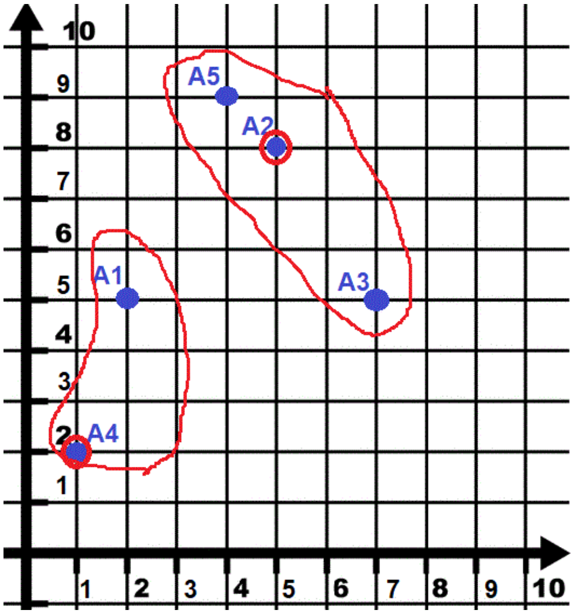
d (A4, A2) = 7.21, d (A4, A4) = 0
d (A4, A4) is less than d (A4, A2)
A4 € C2

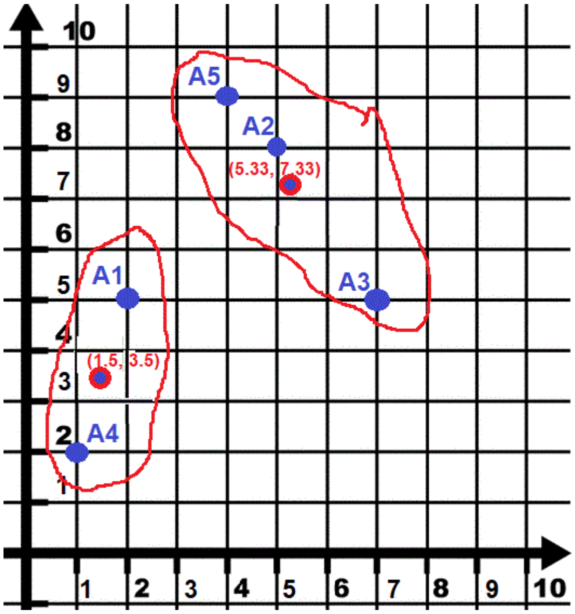
d (A5, A2) = 1.41, d (A5, A4) = 7.62
d (A5, A2) is less than d (A5, A4)
A5 € C1

C1 → {A2, A3, A5}, C2 → {A1, A4}

b)







c) C1 → {A2, A3, A5}, C2 → {A1, A4}

In C1 (A2)

Cohesion: d (A3, A2) = **3.16**, d (A5, A2) = **1.41**

a (A2) = $\mathbf{\frac{3.16+1.41}{2}}$ **= 2.51**

Separation: d (A1, A2) = **4.24** d (A4, A2) = **7.21**

b (A2) = $\mathbf{\frac{2.24+7.21}{2}}$ **= 5.73**

In C1 (A3)

Cohesion: d (A2, A3) = **3.61**, d (A5, A3) = $\sqrt{(4-7)^2+(9-5)^2}$ = **5**

a (A3) = $\mathbf{\frac{3.61+5}{2}}$ **= 4.31**

Separation: d (A1, A3) = $\sqrt{(2-7)^2+(5-5)^2}$ = **5**, d (A4, A3) = $\sqrt{(1-7)^2+(2-5)^2}$ = **6.71**

b (A3) = $\mathbf{\frac{5+6.71}{2}}$ **= 5.86**

In C1 (A5)

Cohesion: d (A2, A5) = **1.41**, d (A3, A5) = $\sqrt{(7-4)^2+(5-9)^2}$ = **5**

a (A3) = $\frac{1.41+5}{2}$ **= 3.21**

Separation: d (A1, A5) = $\sqrt{(2-4)^2+(5-9)^2}$ = **4.47**, d (A4, A5) = 7.**62**

b (A3) = $\frac{4.47+7.62}{2}$ **= 6.05**

In C2 (A1)

Cohesion: d (A4, A1) = 3.16

a (A3) = $\frac{3.16}{1}$ **= 3.16**

Separation: d (A2, A1) = **4.24**, d (A3, A1) = $\sqrt{(7-2)^2+(5-5)^2}$ = **5**

d (A5, A1) = $\sqrt{(4-2)^2+(9-5)^2}$ = **4.47**

b (A1) = $\frac{4.24+5+4.47}{3}$ **= 4.57**

In C2 (A4)

Cohesion: d (A1, A4) = 3.16

a (A4) = $\frac{3.16}{1}$ **= 3.16**

Separation: d (A2, A4) = **7.21**, d (A3, A4) = **6.71**

d (A5, A4) = **7.62**

b (A4) = $\frac{7.21+6.71+7.62}{3}$ **= 7.18**

a(A1) = 3.16, b(A1) = 4.57

a(A2) = 2.51, b(A2) = 5.73

a(A3) = 4.31, b(A3) = 5.86

a(A4) = 3.16, b(A4) = 7.18

a(A5) = 3.21, b(A5) = 6.05

S (i) = $\frac{\boldsymbol{b(i)-a(i)}}{\boldsymbol{max(a(i),b(i))}}$

S (A1) = $\frac{\boldsymbol{b(A1)-a(A1)}}{\boldsymbol{max(a(A1),b(A1))}}$ **=** $\frac{\mathbf{4.57-3.16}}{\mathbf{4.57}}$ **=** $\boxed{\mathbf{0.31}}$

S (A2) = $\frac{\boldsymbol{b(A2)-a(A2)}}{\boldsymbol{max(a(A2),b(A2))}}$ **=** $\frac{\mathbf{5.73-2.51}}{\mathbf{5.73}}$ **=** $\boxed{\mathbf{0.56}}$

S (A3) = $\frac{b(A3)-a(A3)}{max(a(A3),b(A3))} = \frac{5.86-4.31}{5.86} = \boxed{0.26}$

S (A4) = $\frac{b(A4)-a(A4)}{max(a(A4),b(A4))} = \frac{7.18-3.16}{7.18} = \boxed{0.56}$

S (A5) = $\frac{b(A5)-a(A5)}{max(a(A5),b(A5))} = \frac{6.05-3.21}{6.05} = \boxed{0.47}$

$\boxed{\textbf{The silhouette score} = \frac{0.31+0.56+0.26+0.56+0.47}{5} = \mathbf{0.43}}$

C1 → {A2, A3, A5} {(5, 8), (7, 5), (4, 9)}

$X_{c1} = \frac{5+7+4}{3} = \mathbf{5.33}, \mathbf{Y_{c1}} = \frac{8+5+9}{3} = \mathbf{7.33}$

New Centroid (C_{c1}) = (5.33, 7.33)

C2 → {A1, A4} {(2, 5), (1, 2)}

$X_{c2} = \frac{2+1}{2} = \mathbf{1.5}, \mathbf{Y_{c2}} = \frac{5+2}{2} = \mathbf{3.5}$

New Centroid (C_{c2}) = (1.5, 3.5)

WSS = $\sum_{i=1}^{m}(A_i - C_i)^2$

WSS = $(5-5.33)^2 + (8-7.33)^2 + (7-5.33)^2 + (5-7.33)^2$

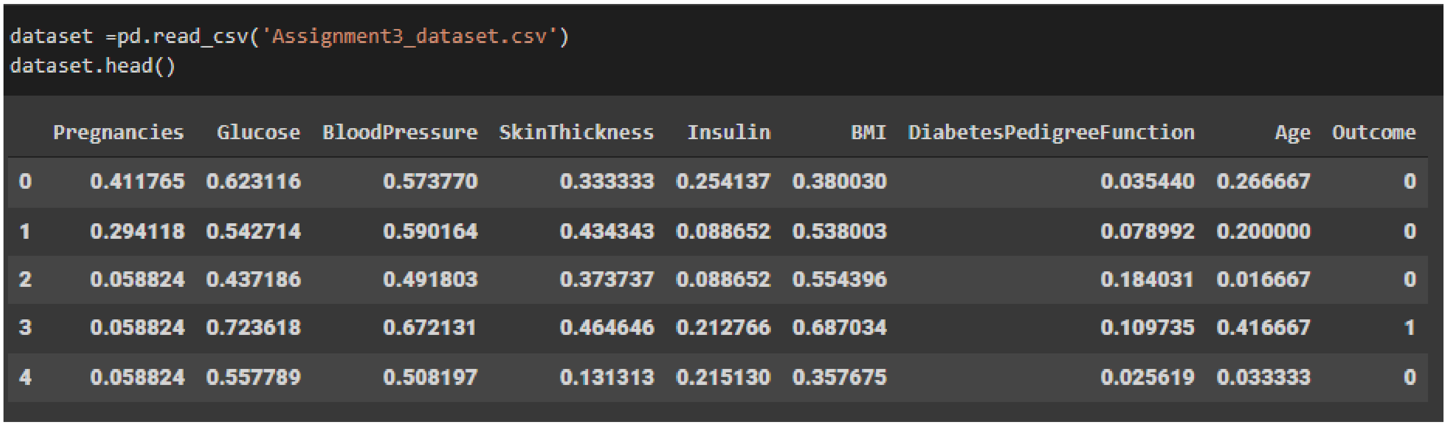
$+ (4-5.33)^2 + (9-7.33)^2 + (2-1.5)^2 + (5-3.5)^2$

$+ (1-1.5)^2 + (2-3.5)^2$ = 18.33.

$\boxed{\textbf{WSS score = 18.33}}$

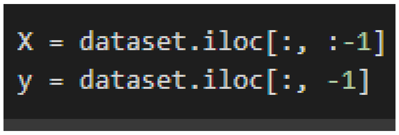
Part 2: Programming

1. Use scikit-learn to implement Logistic Regression (LR) and K-Nearest Neighbor (KNN).

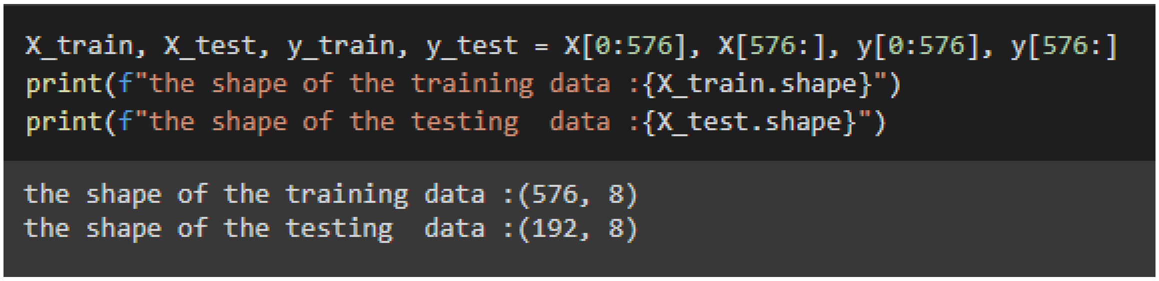


```
dataset =pd.read_csv('Assignment3_dataset.csv')
dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0.411765	0.623116	0.573770	0.333333	0.254137	0.380030	0.035440	0.266667	0
1	0.294118	0.542714	0.590164	0.434343	0.088652	0.538003	0.078992	0.200000	0
2	0.058824	0.437186	0.491803	0.373737	0.088652	0.554396	0.184031	0.016667	0
3	0.058824	0.723618	0.672131	0.464646	0.212766	0.687034	0.109735	0.416667	1
4	0.058824	0.557789	0.508197	0.131313	0.215130	0.357675	0.025619	0.033333	0

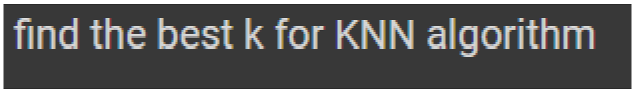


```
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]
```

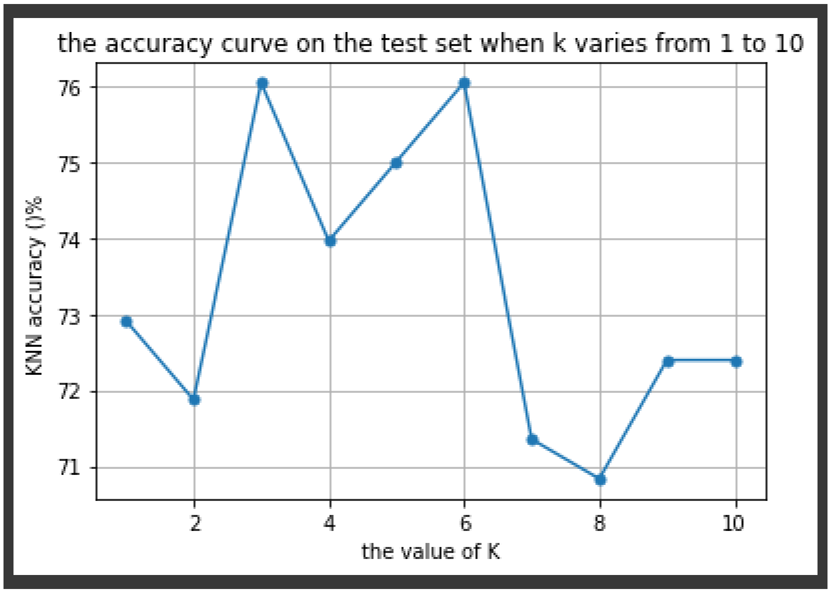


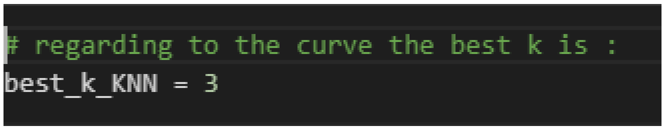
```
X_train, X_test, y_train, y_test = X[0:576], X[576:], y[0:576], y[576:]
print(f"the shape of the training data :{X_train.shape}")
print(f"the shape of the testing  data :{X_test.shape}")
```

```
the shape of the training data :(576, 8)
the shape of the testing  data :(192, 8)
```



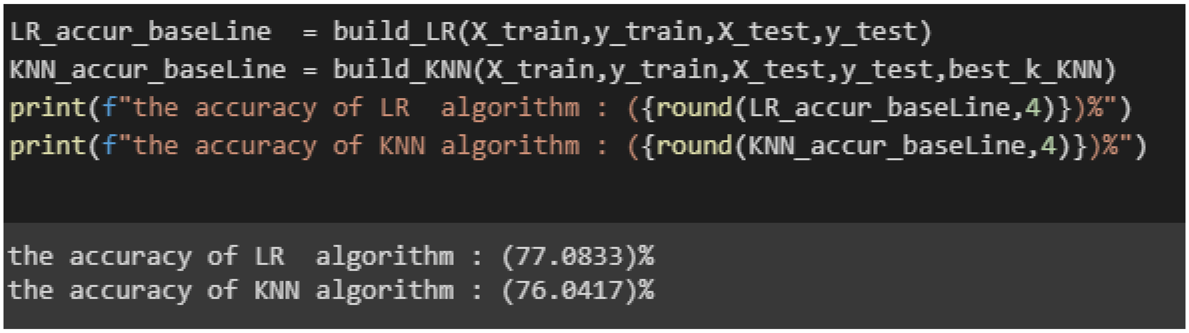
find the best k for KNN algorithm





```
# regarding to the curve the best k is :
best_k_KNN = 3
```

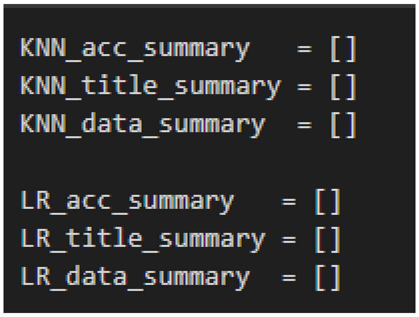
(a) Provide the accuracy of LR and K-NN classifier as baseline performances.



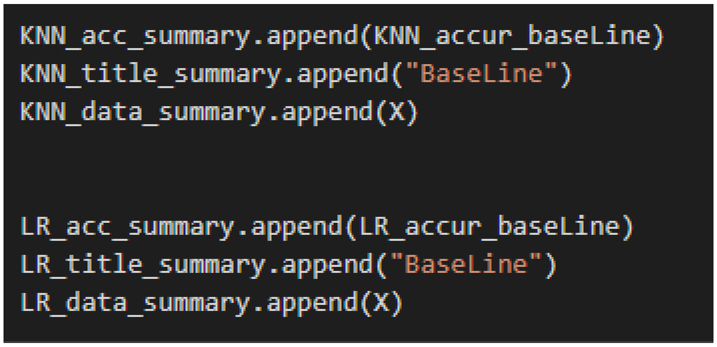
```
LR_accur_baseLine  = build_LR(X_train,y_train,X_test,y_test)
KNN_accur_baseLine = build_KNN(X_train,y_train,X_test,y_test,best_k_KNN)
print(f"the accuracy of LR  algorithm : ({round(LR_accur_baseLine,4)})%")
print(f"the accuracy of KNN algorithm : ({round(KNN_accur_baseLine,4)})%")
```

```
the accuracy of LR  algorithm : (77.0833)%
the accuracy of KNN algorithm : (76.0417)%
```

through this assignment we will face many numbers of accuracies and different training sets with two classifiers (LR and KNN) So , we have built 6 lists to keep track all of these information

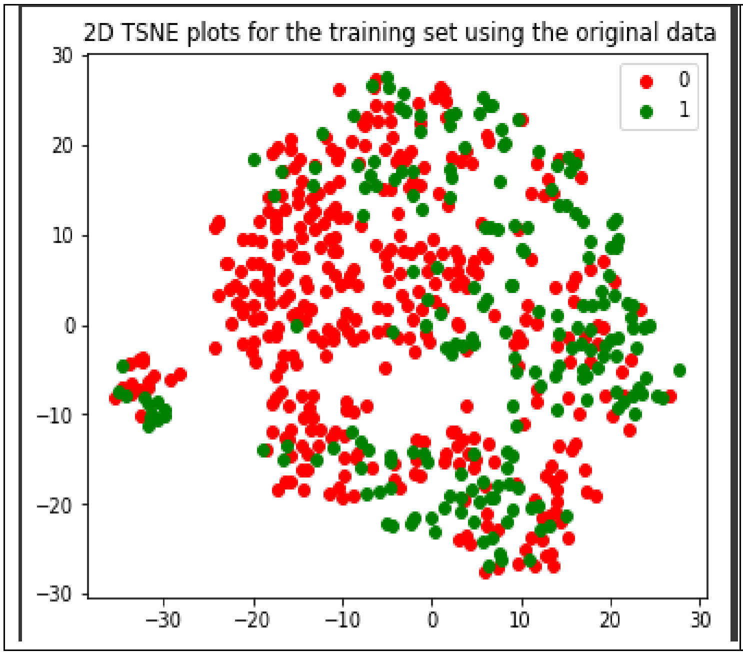


```
KNN_acc_summary   = []
KNN_title_summary = []
KNN_data_summary  = []

LR_acc_summary   = []
LR_title_summary = []
LR_data_summary  = []
```

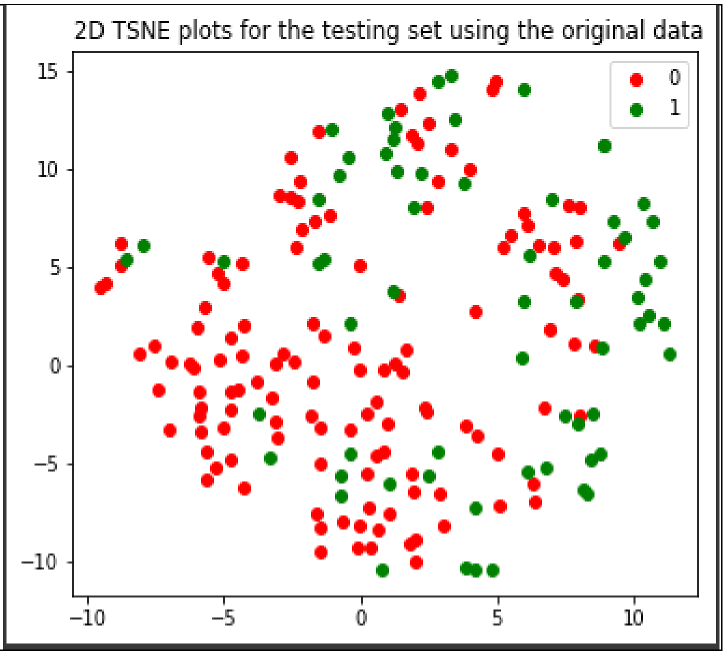


```
KNN_acc_summary.append(KNN_accur_baseLine)
KNN_title_summary.append("BaseLine")
KNN_data_summary.append(X)


LR_acc_summary.append(LR_accur_baseLine)
LR_title_summary.append("BaseLine")
LR_data_summary.append(X)
```

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



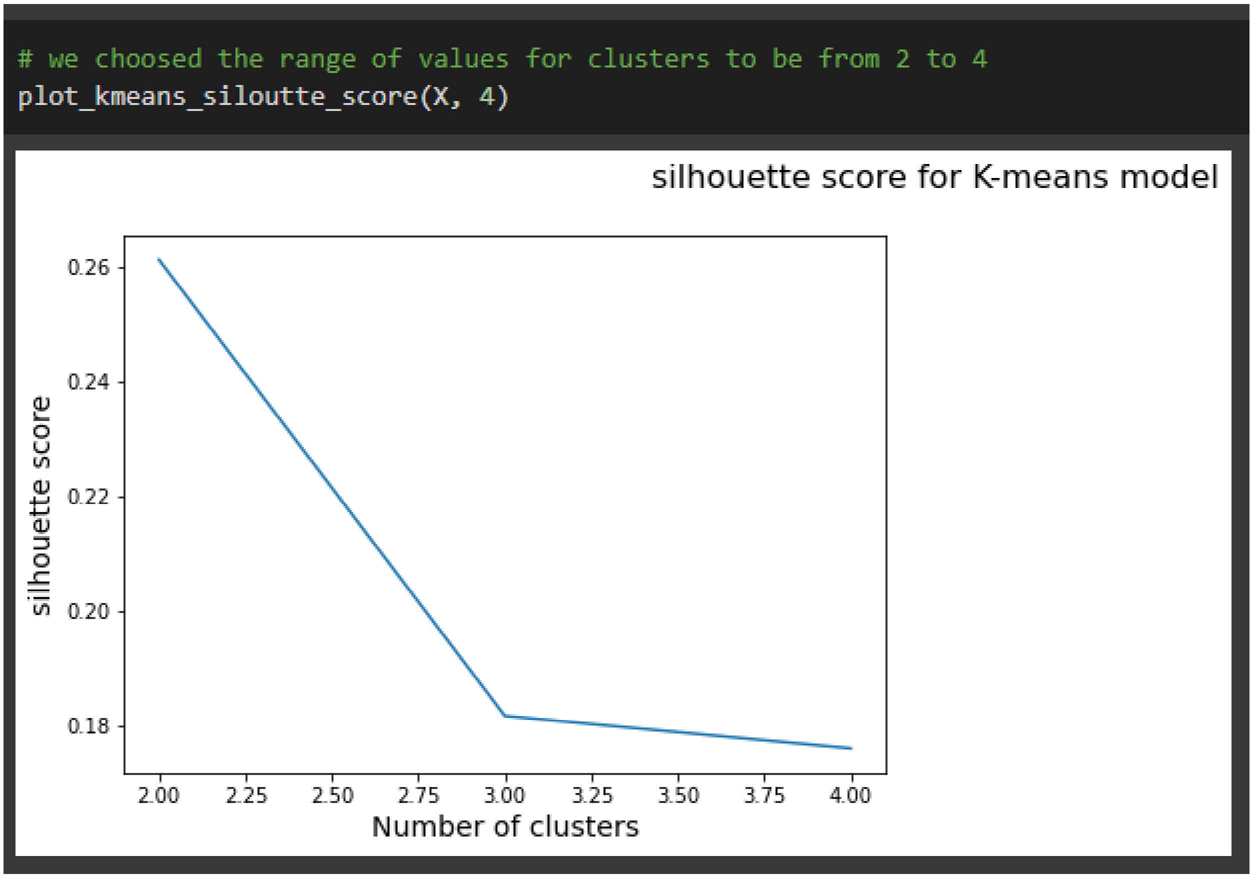
2D TSNE plots for the training set using the original data



2D TSNE plots for the testing set using the original data

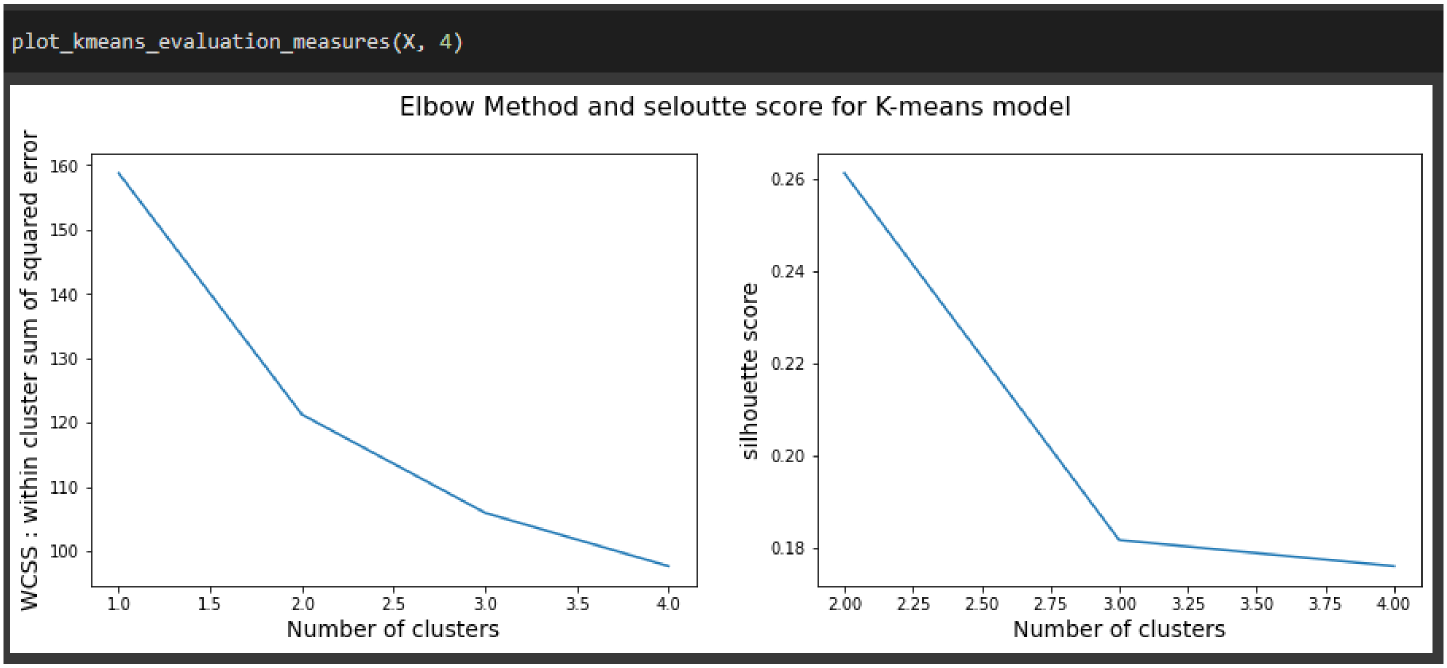
2. Choose the best number of clusters for k-means clustering algorithm

(a) Plot the silhouette score vs the number of clusters.



```
# we choosed the range of values for clusters to be from 2 to 4
plot_kmeans_siloutte_score(X, 4)
```

(b) Determine the optimal number of clusters for k-Means



```
plot_kmeans_evaluation_measures(X, 4)
```

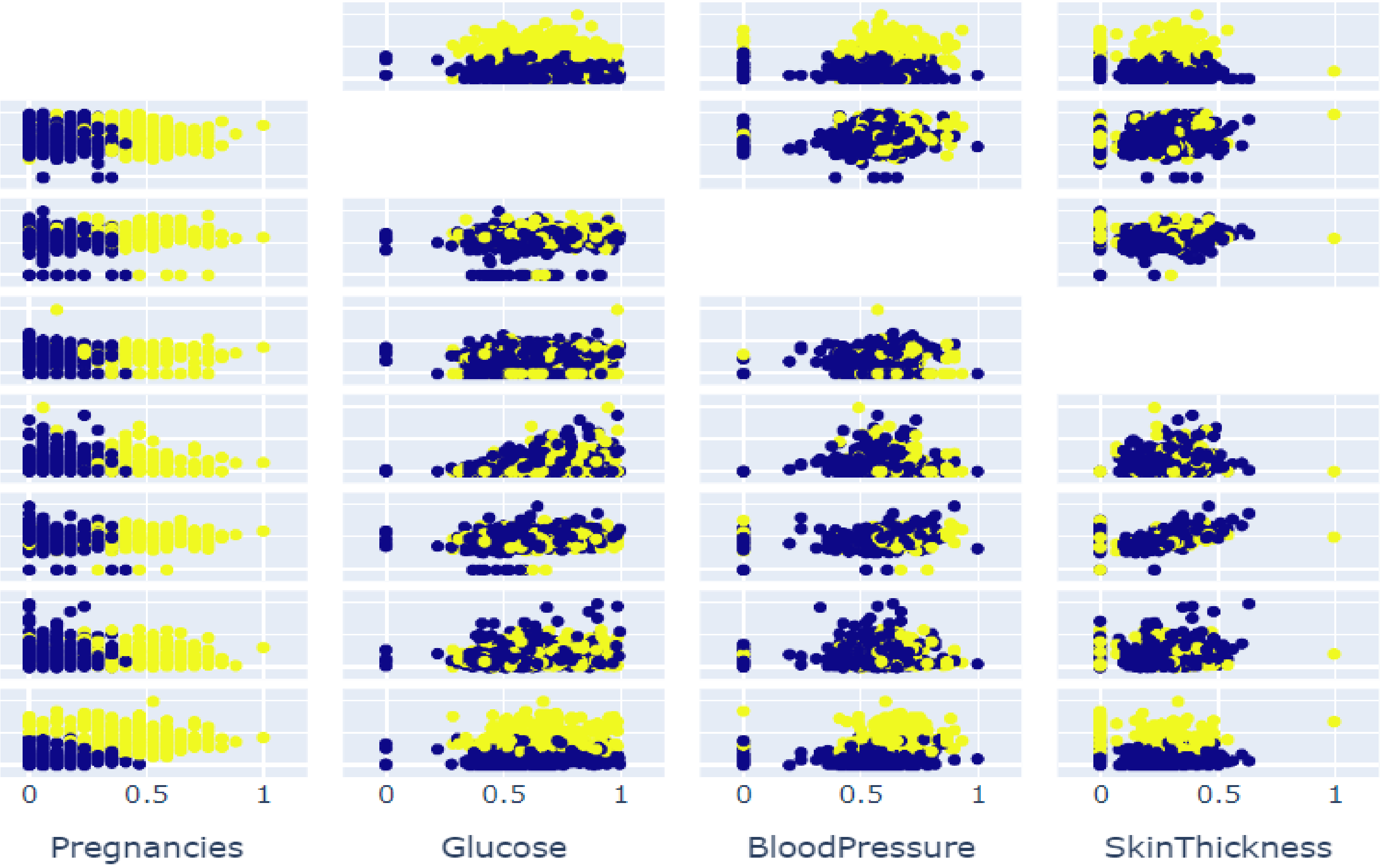
we chose the clusters number with the highest silhouette score (to minimize the distances between the data points in the same clusters and maximize the distances between data points in different clusters)

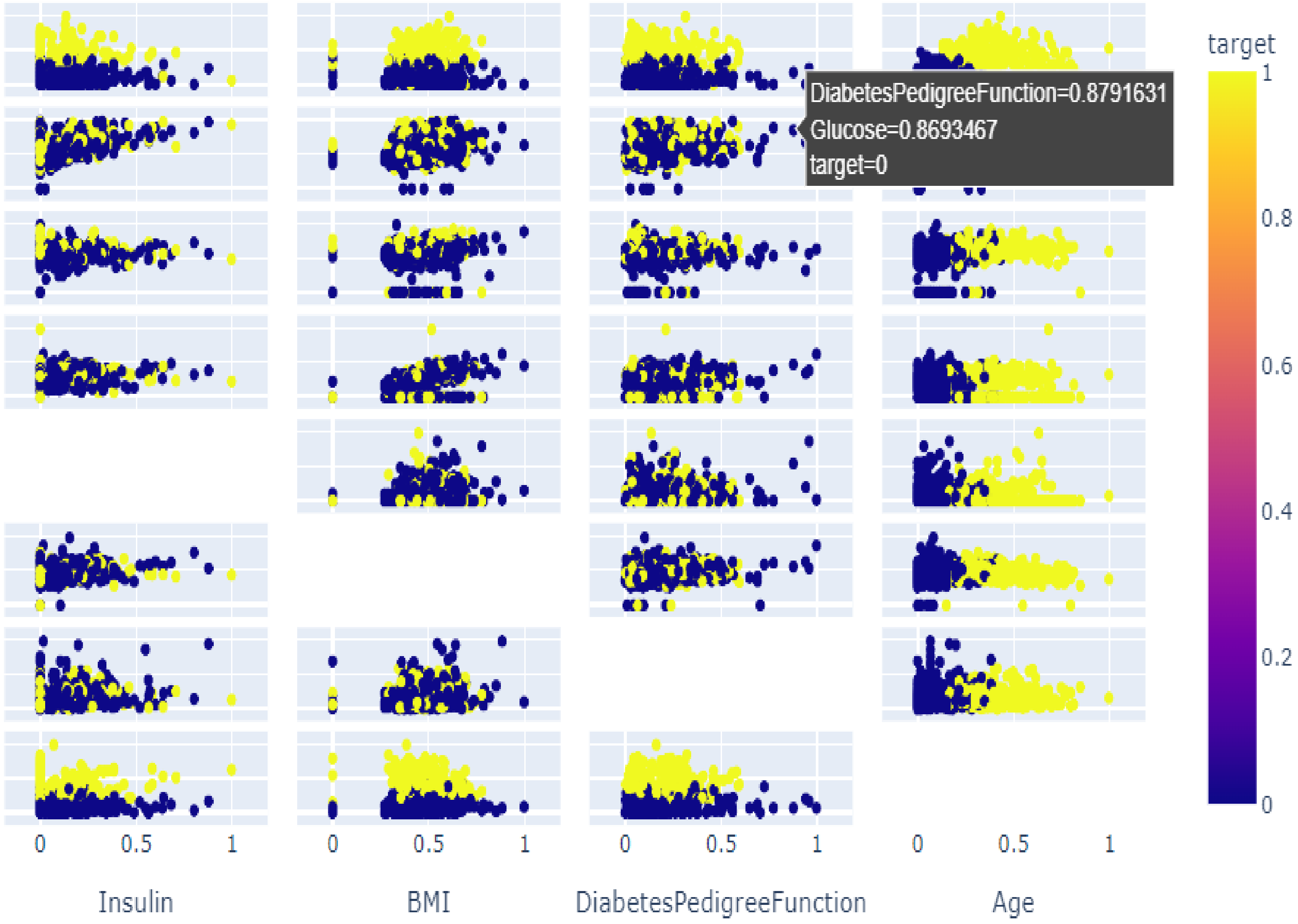
---

to make sure we plotted the elbow to keep track of the sum of squared error within clusters with different number of K, we were hesitated between k = 2 and k = 3, so we looked to the silhouette score so we chose k = 2, and logically it makes sense because we have two classes in our dataset

(c) Plot the clustered data with optimum number of clusters.

Visualize all the original dimensions

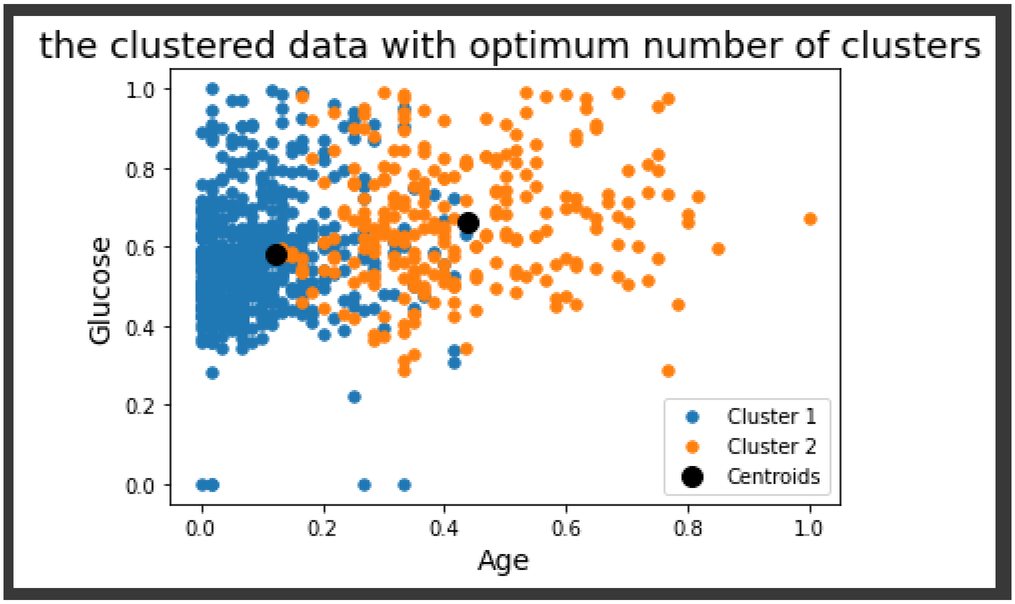




Some feature is able to represent our classes in the dataset. And others are not.

For example, if we used age and any other feature it could represent the two clusters.

(notice: more details in the conclusion)

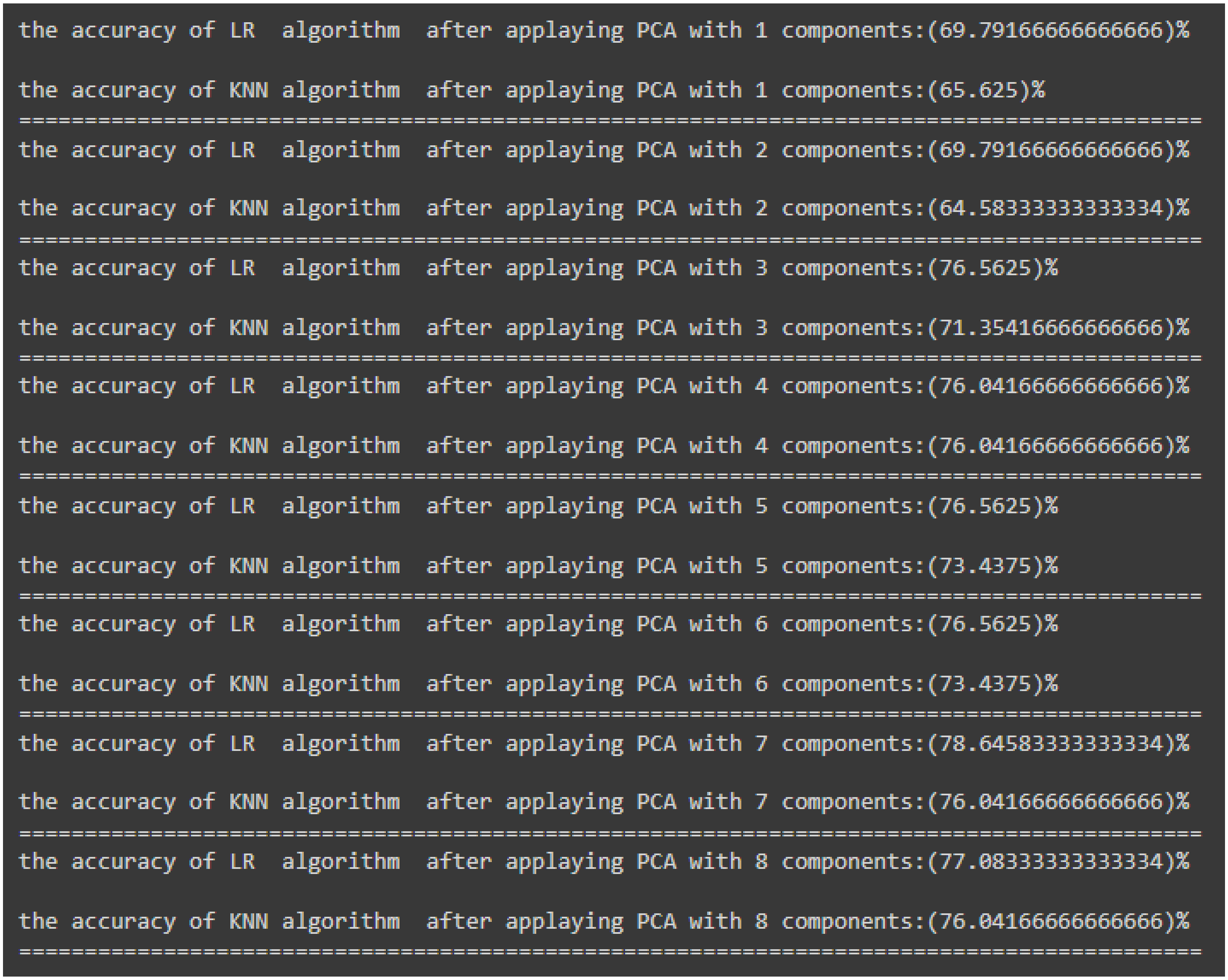


Actually, the previous visualization let us understand more about our dataset.

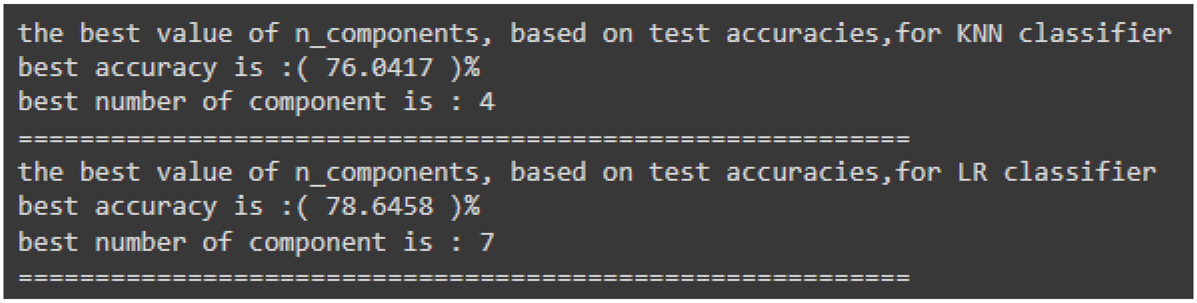
[1] “This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes.”

3. Apply the following Dimensionality Reduction (DR) methods:

(a) Find the best value of n components, based on test accuracies, for both classifiers (LR and K-NN).

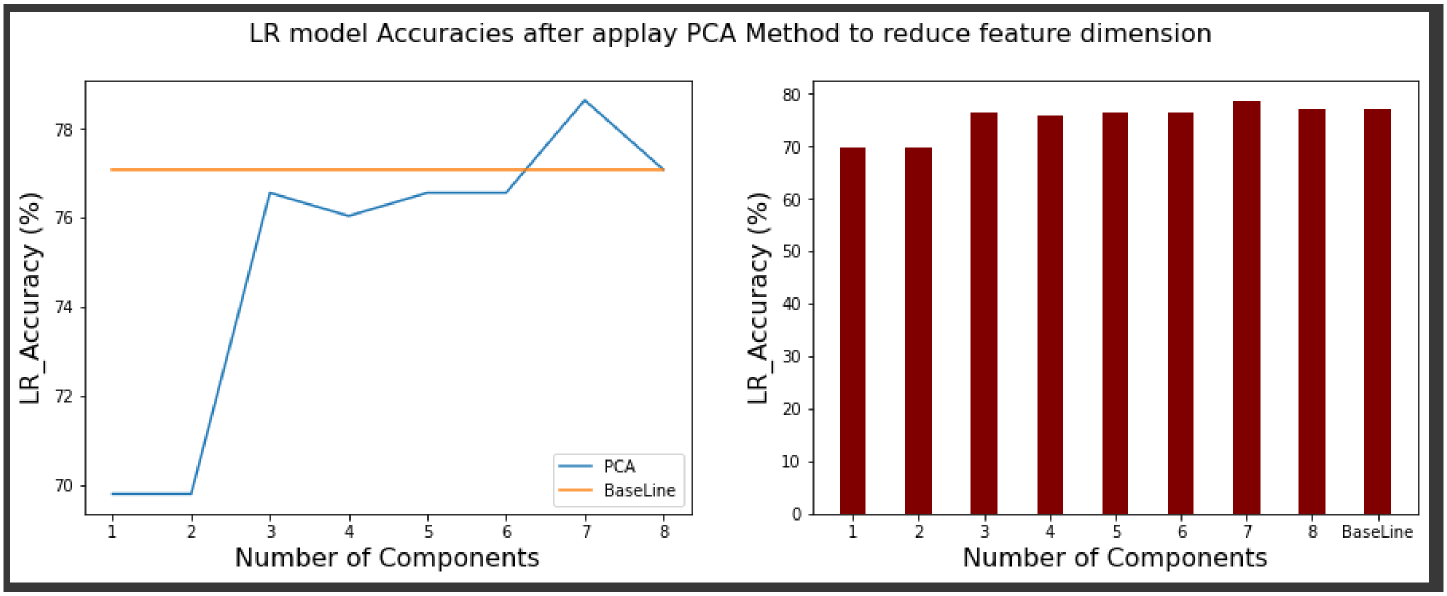


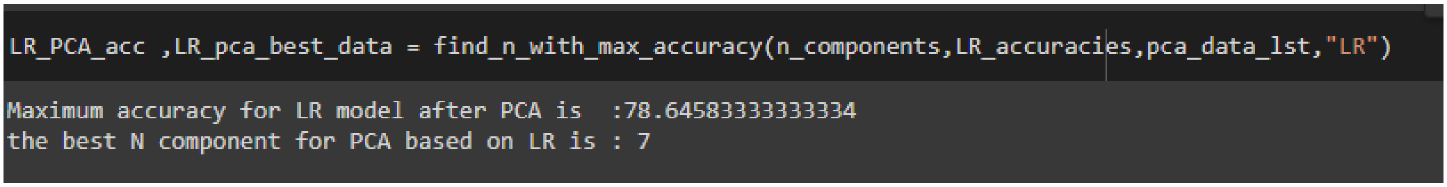
```
the accuracy of LR  algorithm  after applaying PCA with 1 components:(69.79166666666666)%

the accuracy of KNN algorithm  after applaying PCA with 1 components:(65.625)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 2 components:(69.79166666666666)%

the accuracy of KNN algorithm  after applaying PCA with 2 components:(64.58333333333334)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 3 components:(76.5625)%

the accuracy of KNN algorithm  after applaying PCA with 3 components:(71.35416666666666)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 4 components:(76.04166666666666)%

the accuracy of KNN algorithm  after applaying PCA with 4 components:(76.04166666666666)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 5 components:(76.5625)%

the accuracy of KNN algorithm  after applaying PCA with 5 components:(73.4375)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 6 components:(76.5625)%

the accuracy of KNN algorithm  after applaying PCA with 6 components:(73.4375)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 7 components:(78.64583333333334)%

the accuracy of KNN algorithm  after applaying PCA with 7 components:(76.04166666666666)%
==========================================================================================
the accuracy of LR  algorithm  after applaying PCA with 8 components:(77.08333333333334)%

the accuracy of KNN algorithm  after applaying PCA with 8 components:(76.04166666666666)%
==========================================================================================
```



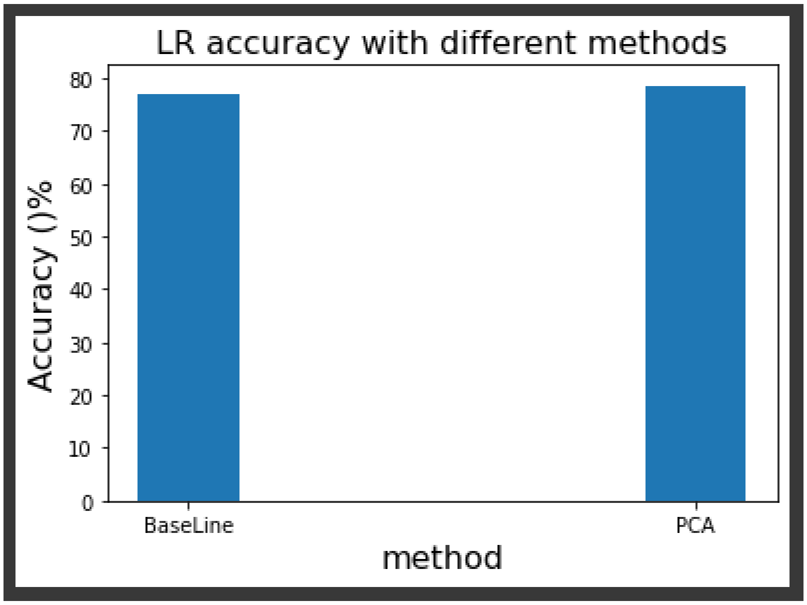
```
the best value of n_components, based on test accuracies,for KNN classifier
best accuracy is :( 76.0417 )%
best number of component is : 4
============================================================
the best value of n_components, based on test accuracies,for LR classifier
best accuracy is :( 78.6458 )%
best number of component is : 7
============================================================
```

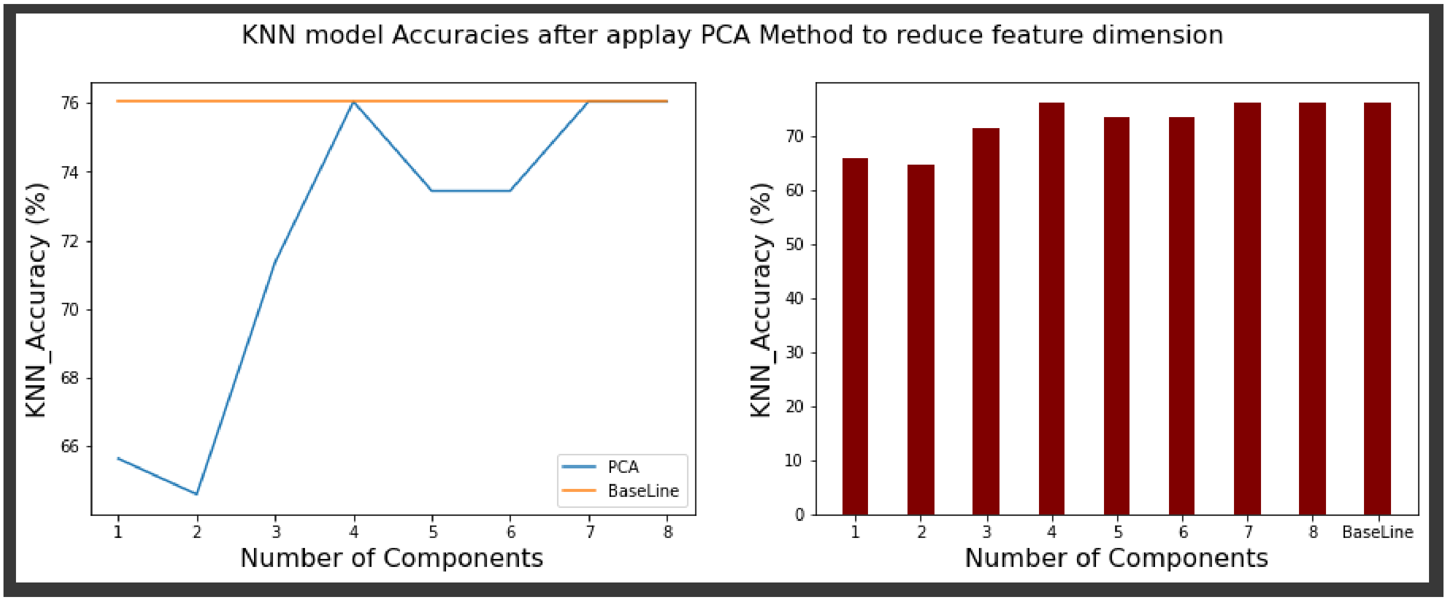
(b) Plot the Number of Components-Accuracy graph with baseline performances for each classifier.

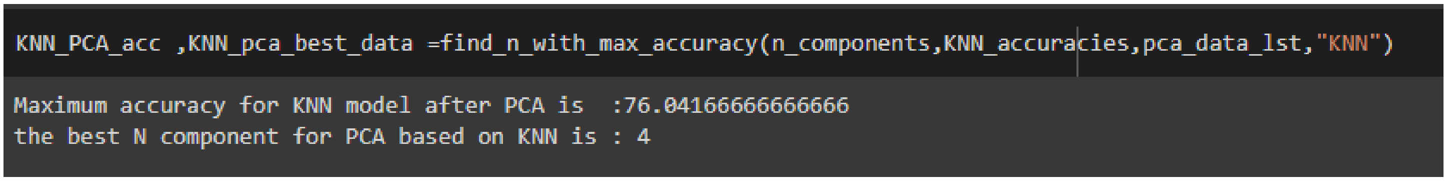




```
LR_PCA_acc ,LR_pca_best_data = find_n_with_max_accuracy(n_components,LR_accuracies,pca_data_lst,"LR")

Maximum accuracy for LR model after PCA is  :78.64583333333334
the best N component for PCA based on LR is : 7
```

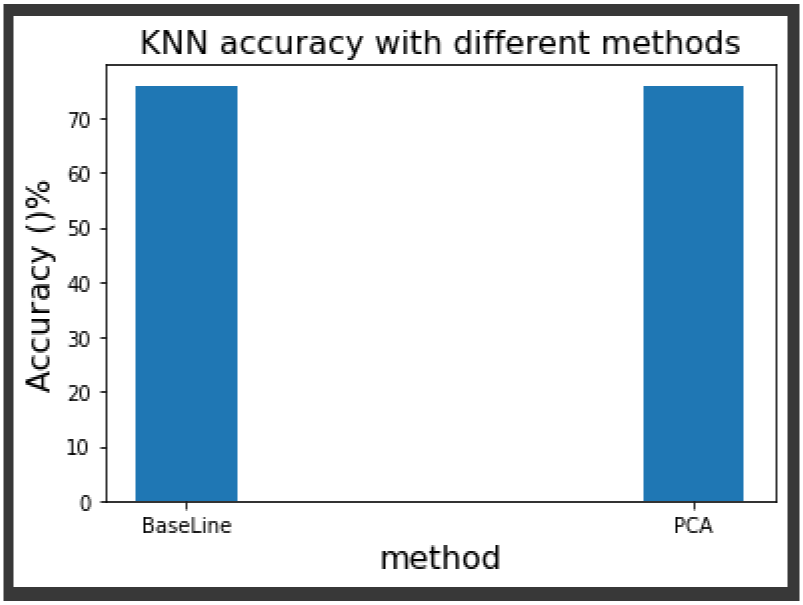


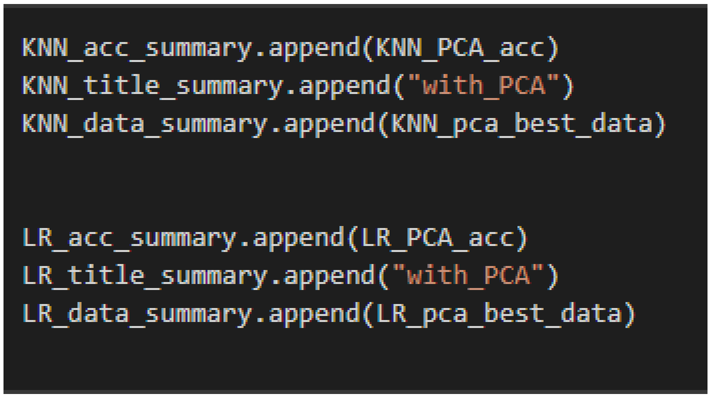




```
KNN_PCA_acc ,KNN_pca_best_data =find_n_with_max_accuracy(n_components,KNN_accuracies,pca_data_lst,"KNN")
```

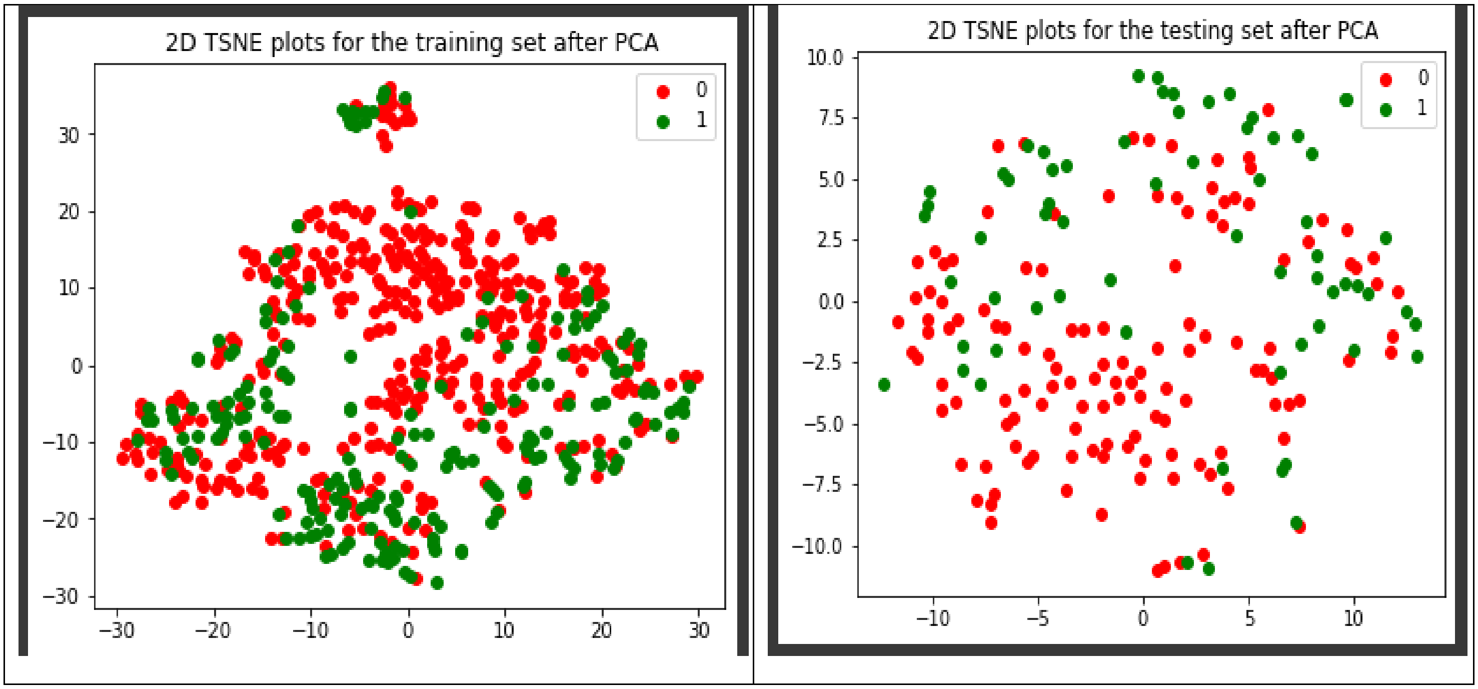
```
Maximum accuracy for KNN model after PCA is  :76.04166666666666
the best N component for PCA based on KNN is : 4
```





```
KNN_acc_summary.append(KNN_PCA_acc)
KNN_title_summary.append("with_PCA")
KNN_data_summary.append(KNN_pca_best_data)


LR_acc_summary.append(LR_PCA_acc)
LR_title_summary.append("with_PCA")
LR_data_summary.append(LR_pca_best_data)
```

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

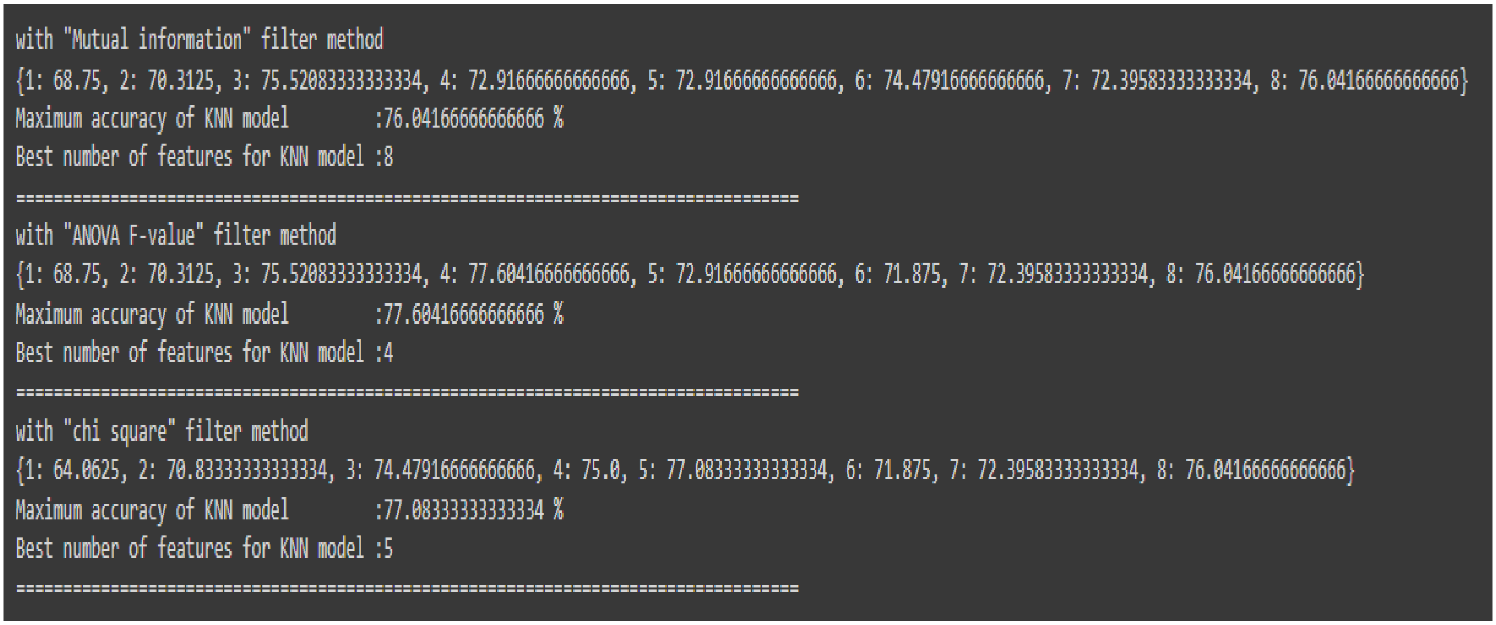


2D TSNE plots for the training set after PCA

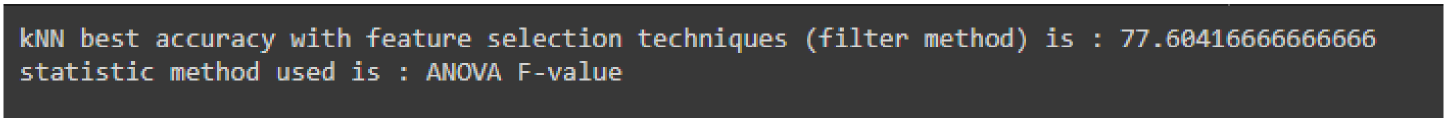
2D TSNE plots for the testing set after PCA

4. Use the following Feature Selection methods (one for each method). Find the best number of features based on both, the LR and K-NN classifiers' test accuracies.

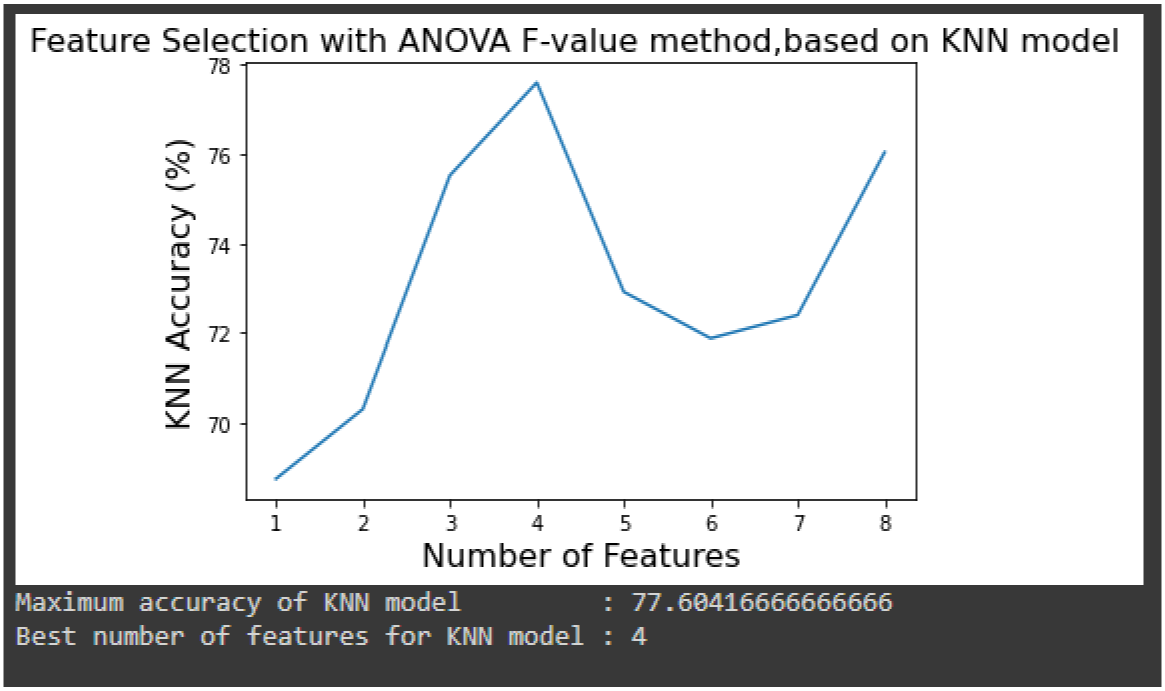
(a) Filter Methods (Information Gain, Variance Threshold etc.). Plot the number of features versus accuracy graph with the improved baseline performance as shown in Q3, using only the method that gives you the best test accuracy.



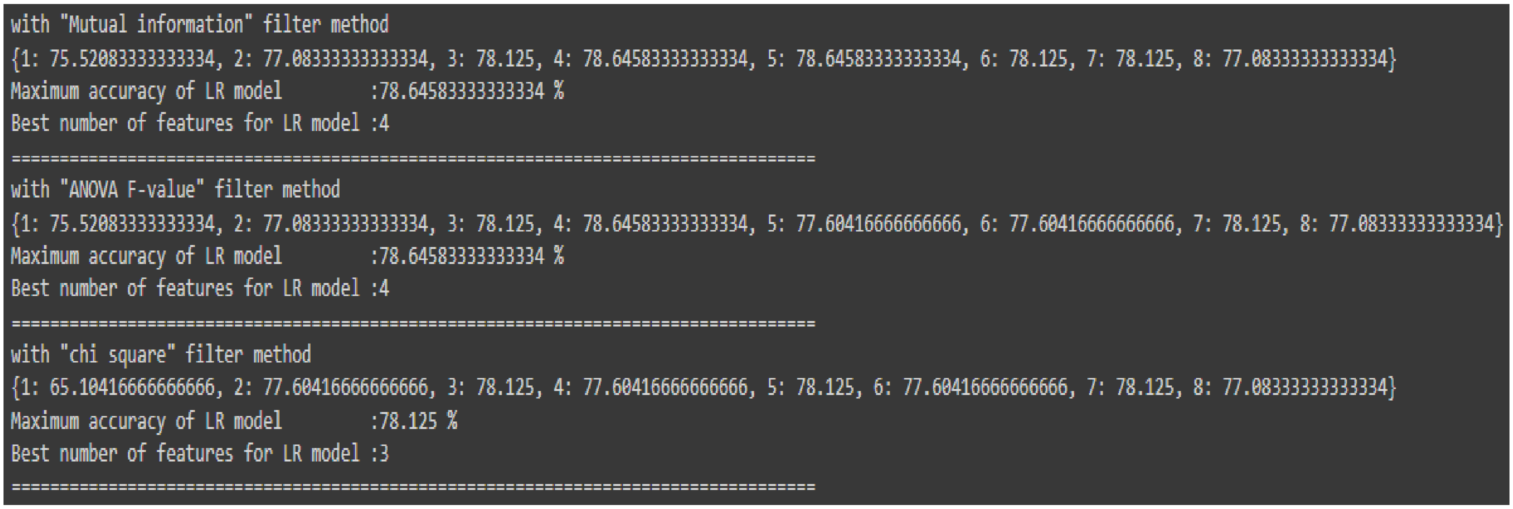
```
with "Mutual information" filter method
{1: 68.75, 2: 70.3125, 3: 75.52083333333334, 4: 72.91666666666666, 5: 72.91666666666666, 6: 74.47916666666666, 7: 72.39583333333334, 8: 76.04166666666666}
Maximum accuracy of KNN model        :76.04166666666666 %
Best number of features for KNN model :8
================================================================================
with "ANOVA F-value" filter method
{1: 68.75, 2: 70.3125, 3: 75.52083333333334, 4: 77.60416666666666, 5: 72.91666666666666, 6: 71.875, 7: 72.39583333333334, 8: 76.04166666666666}
Maximum accuracy of KNN model        :77.60416666666666 %
Best number of features for KNN model :4
================================================================================
with "chi square" filter method
{1: 64.0625, 2: 70.83333333333334, 3: 74.47916666666666, 4: 75.0, 5: 77.08333333333334, 6: 71.875, 7: 72.39583333333334, 8: 76.04166666666666}
Maximum accuracy of KNN model        :77.08333333333334 %
Best number of features for KNN model :5
================================================================================
```



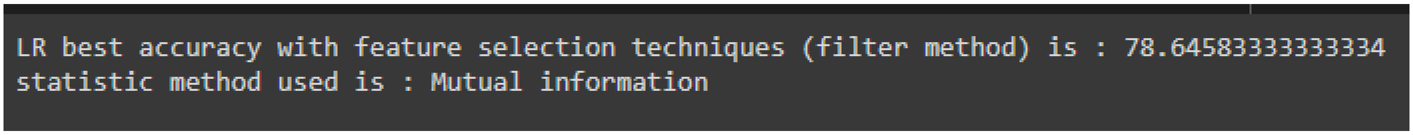
```
kNN best accuracy with feature selection techniques (filter method) is : 77.60416666666666
statistic method used is : ANOVA F-value
```



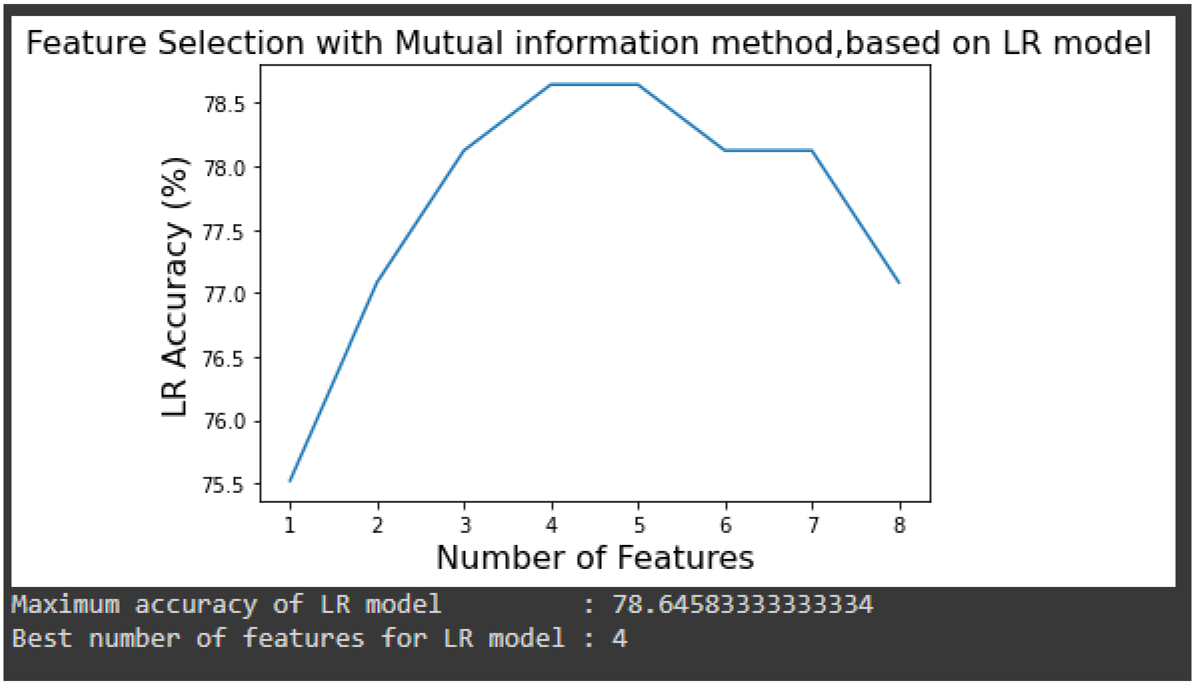
```
Maximum accuracy of KNN model         : 77.60416666666666
Best number of features for KNN model : 4
```



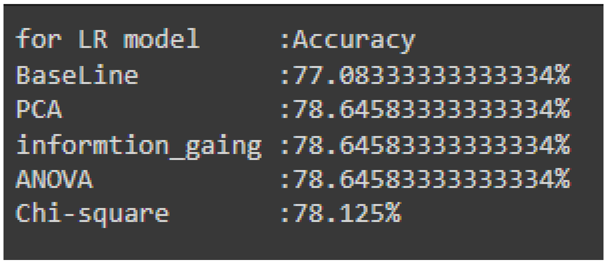
```
with "Mutual information" filter method
{1: 75.52083333333334, 2: 77.08333333333334, 3: 78.125, 4: 78.64583333333334, 5: 78.64583333333334, 6: 78.125, 7: 78.125, 8: 77.08333333333334}
Maximum accuracy of LR model         :78.64583333333334 %
Best number of features for LR model :4
===================================================================================
with "ANOVA F-value" filter method
{1: 75.52083333333334, 2: 77.08333333333334, 3: 78.125, 4: 78.64583333333334, 5: 77.60416666666666, 6: 77.60416666666666, 7: 78.125, 8: 77.08333333333334}
Maximum accuracy of LR model         :78.64583333333334 %
Best number of features for LR model :4
===================================================================================
with "chi square" filter method
{1: 65.10416666666666, 2: 77.60416666666666, 3: 78.125, 4: 77.60416666666666, 5: 78.125, 6: 77.60416666666666, 7: 78.125, 8: 77.08333333333334}
Maximum accuracy of LR model         :78.125 %
Best number of features for LR model :3
===================================================================================
```



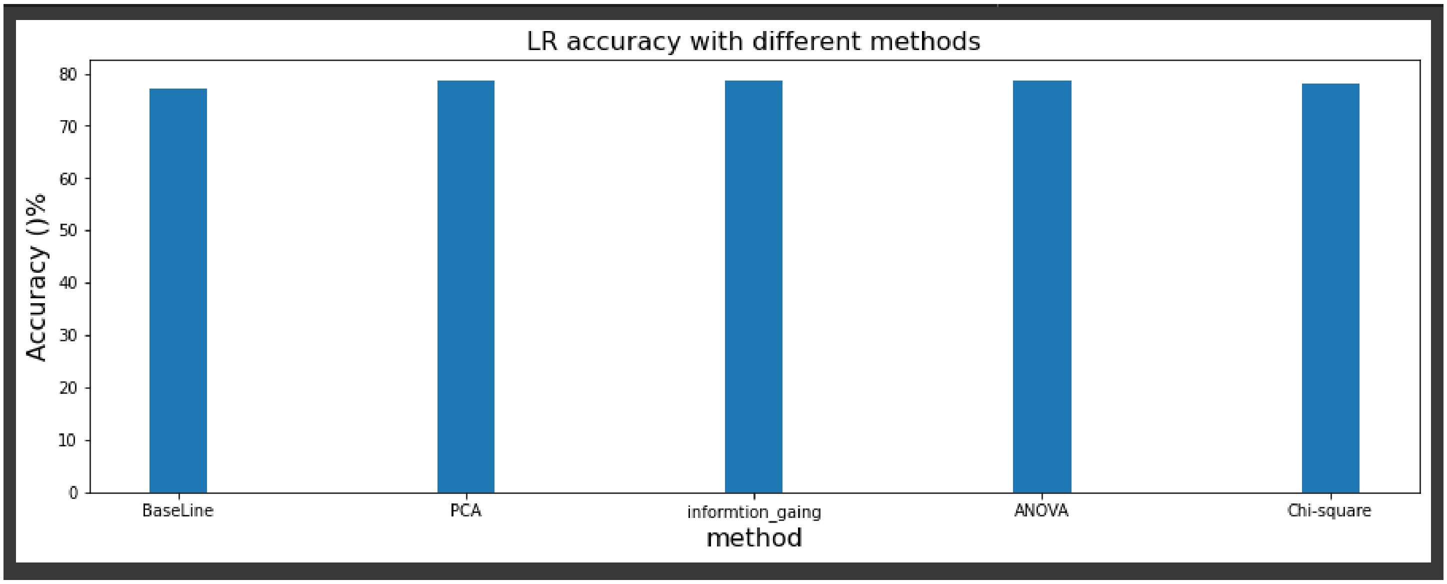
```
LR best accuracy with feature selection techniques (filter method) is : 78.64583333333334
statistic method used is : Mutual information
```

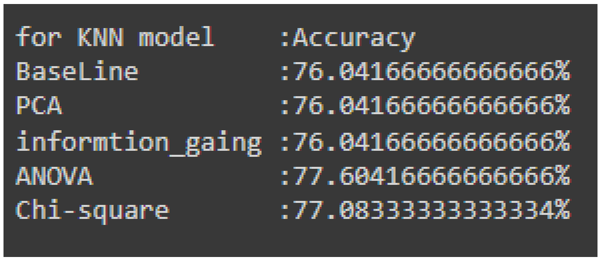


```
Maximum accuracy of LR model         : 78.64583333333334
Best number of features for LR model : 4
```

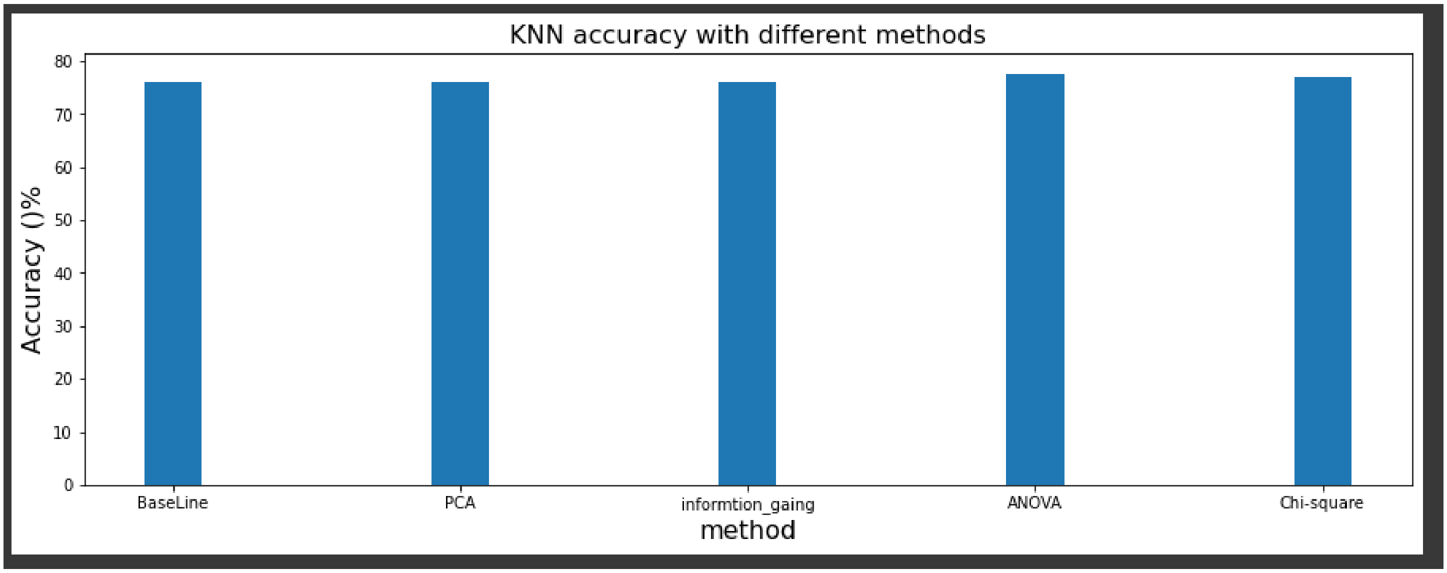


```
for LR model      :Accuracy
BaseLine          :77.08333333333334%
PCA               :78.64583333333334%
informtion_gaing  :78.64583333333334%
ANOVA             :78.64583333333334%
Chi-square        :78.125%
```

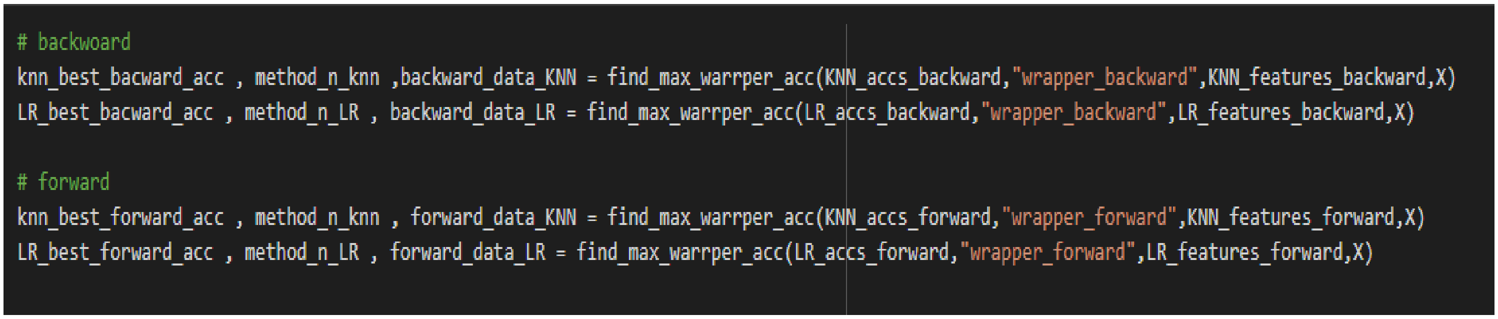




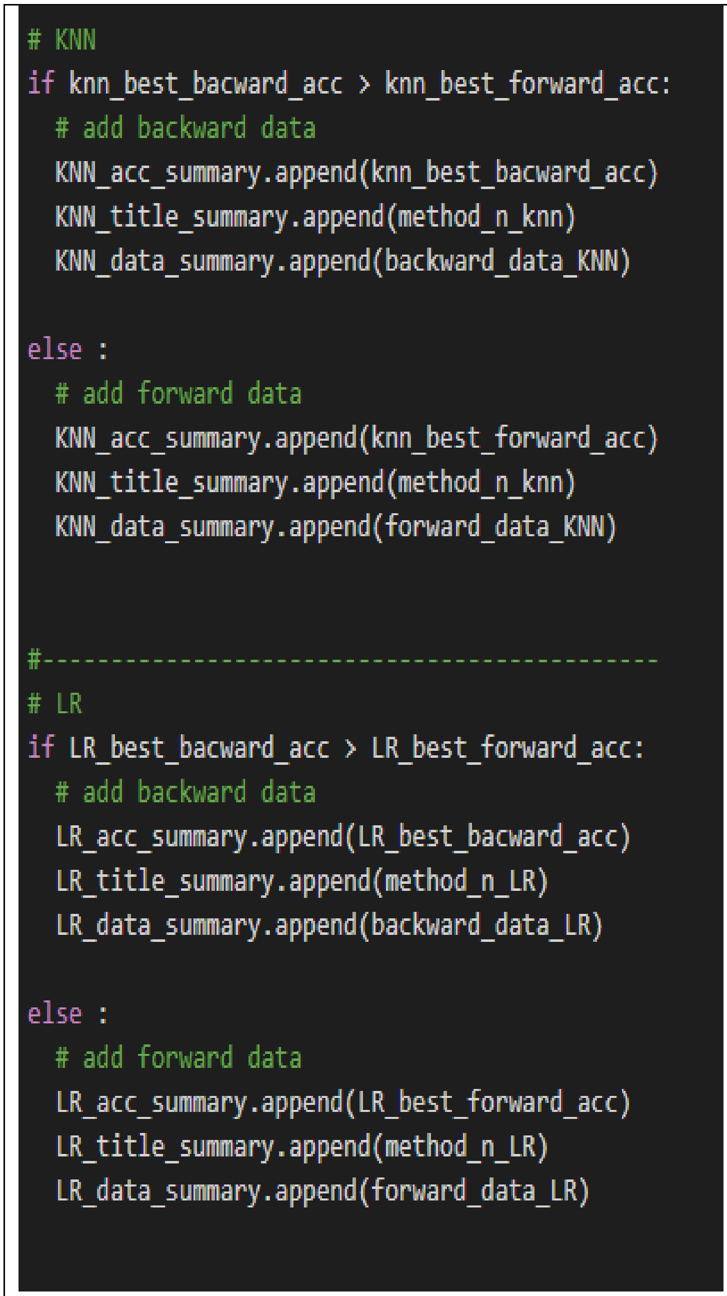
```
for KNN model     :Accuracy
BaseLine          :76.04166666666666%
PCA               :76.04166666666666%
informtion_gaing  :76.04166666666666%
ANOVA             :77.60416666666666%
Chi-square        :77.08333333333334%
```



(b) Wrapper Methods (Forward or Backward Feature Elimination, Recursive Feature Elimination etc.). Plot the number of features versus accuracy graph with the improved baseline performance as shown in Q3, using only the method that gives you the best test accuracy.



```
# backwoard
knn_best_bacward_acc , method_n_knn ,backward_data_KNN = find_max_warrper_acc(KNN_accs_backward,"wrapper_backward",KNN_features_backward,X)
LR_best_bacward_acc , method_n_LR , backward_data_LR = find_max_warrper_acc(LR_accs_backward,"wrapper_backward",LR_features_backward,X)

# forward
knn_best_forward_acc , method_n_knn , forward_data_KNN = find_max_warrper_acc(KNN_accs_forward,"wrapper_forward",KNN_features_forward,X)
LR_best_forward_acc , method_n_LR , forward_data_LR = find_max_warrper_acc(LR_accs_forward,"wrapper_forward",LR_features_forward,X)
```



```
# KNN
if knn_best_bacward_acc > knn_best_forward_acc:
  # add backward data
  KNN_acc_summary.append(knn_best_bacward_acc)
  KNN_title_summary.append(method_n_knn)
  KNN_data_summary.append(backward_data_KNN)

else :
  # add forward data
  KNN_acc_summary.append(knn_best_forward_acc)
  KNN_title_summary.append(method_n_knn)
  KNN_data_summary.append(forward_data_KNN)


#---------------------------------------------
# LR
if LR_best_bacward_acc > LR_best_forward_acc:
  # add backward data
  LR_acc_summary.append(LR_best_bacward_acc)
  LR_title_summary.append(method_n_LR)
  LR_data_summary.append(backward_data_LR)

else :
  # add forward data
  LR_acc_summary.append(LR_best_forward_acc)
  LR_title_summary.append(method_n_LR)
  LR_data_summary.append(forward_data_LR)
```

Comment on Wrapper methods:

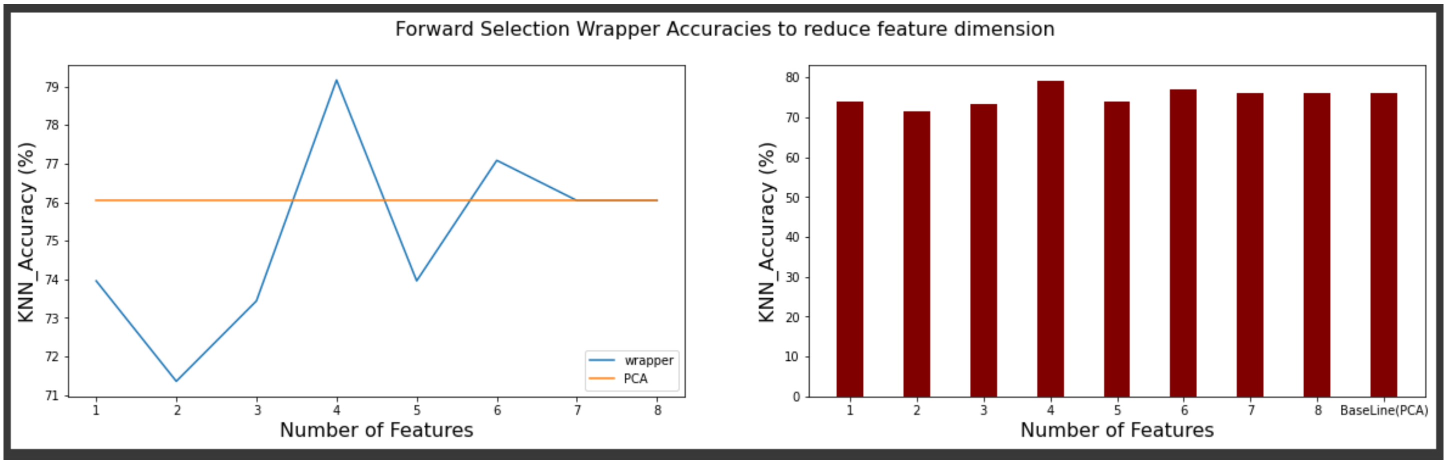
We applied dimensionality reduction with wrapper methods (forward selection and backward elimination) on both LR and KNN classifiers.

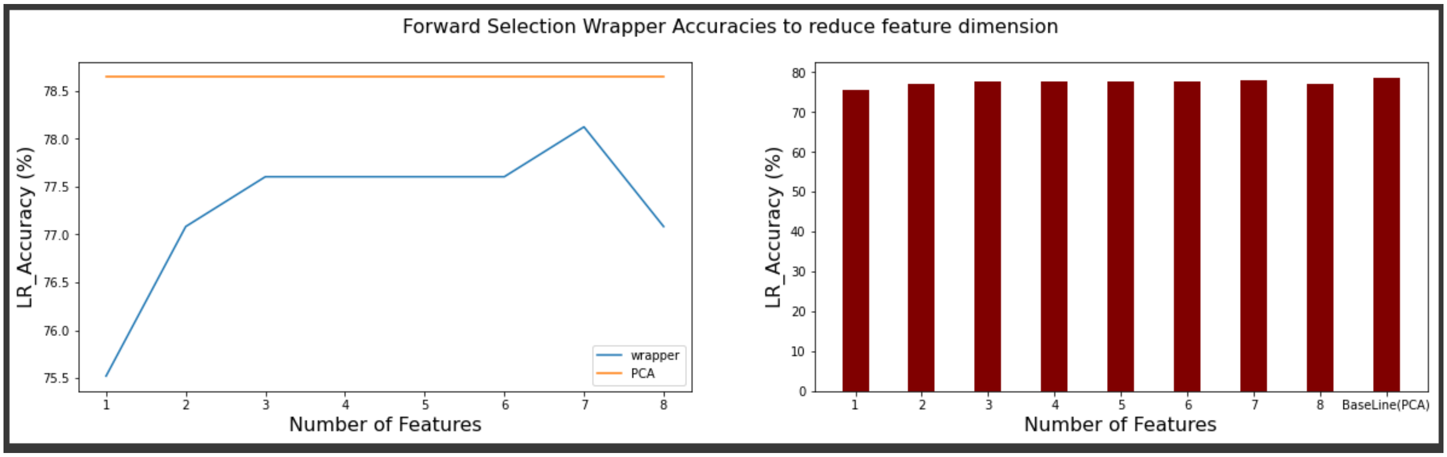
For each classifier, we built our condition to keep track of the accuracy, title, and data for the best method, whoever the best was, we handled its data to use later in sections 5,6, and 7.

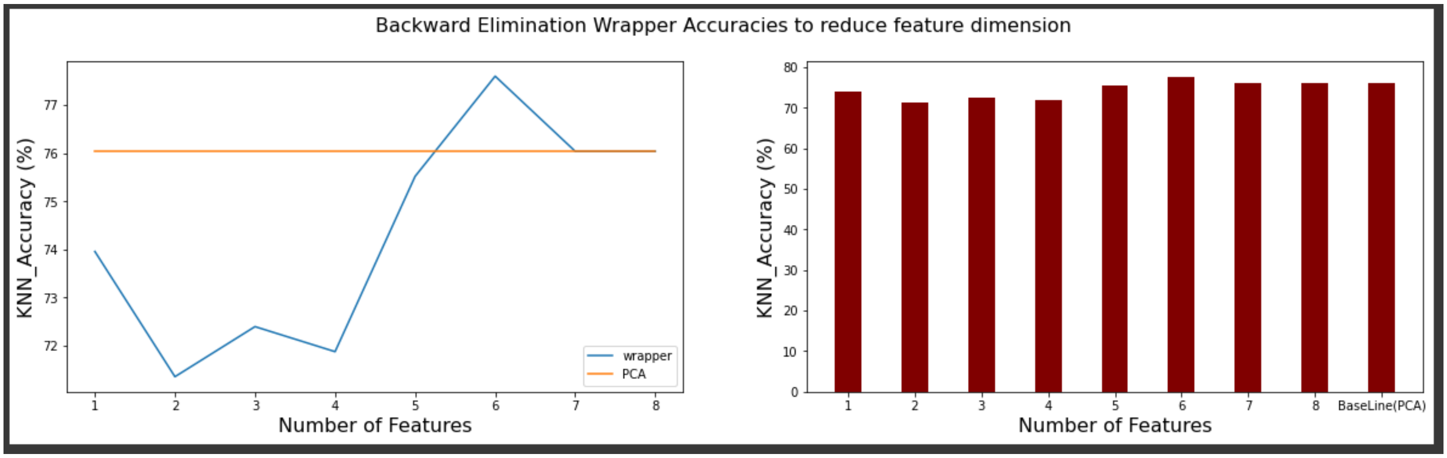
Note that the forward selection method with KNN classifiers was the best in accuracy but the LR classifier forward selection and backward elimination accuracies were the same.

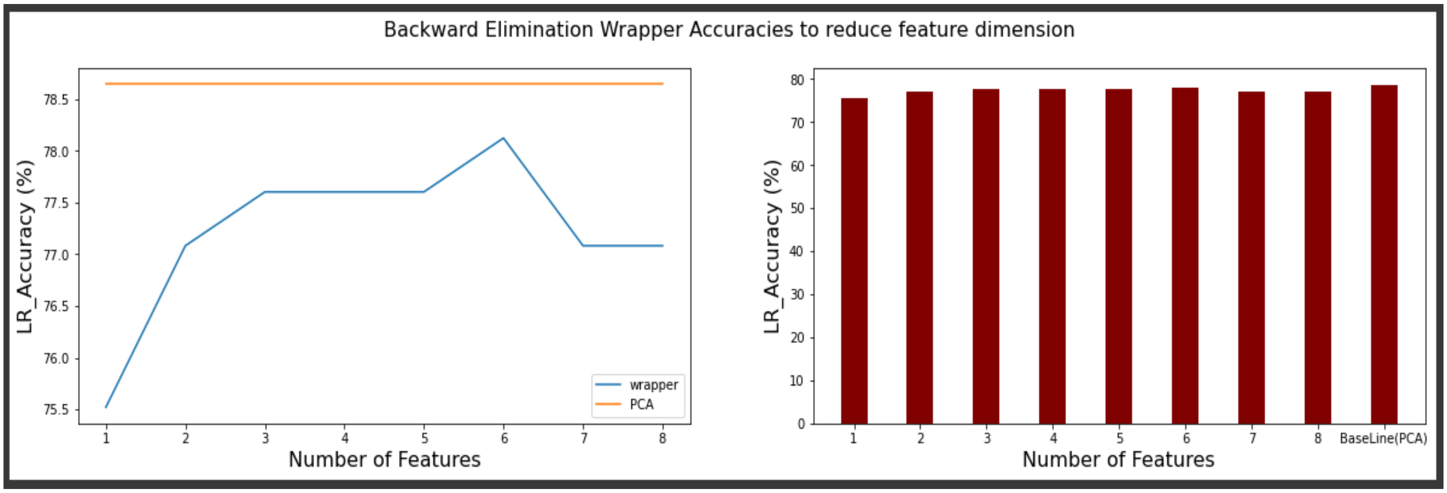
For the next page:

We compared the results from this stage with the improved Baseline which is (PCA), we noticed that PCA with LR was better.

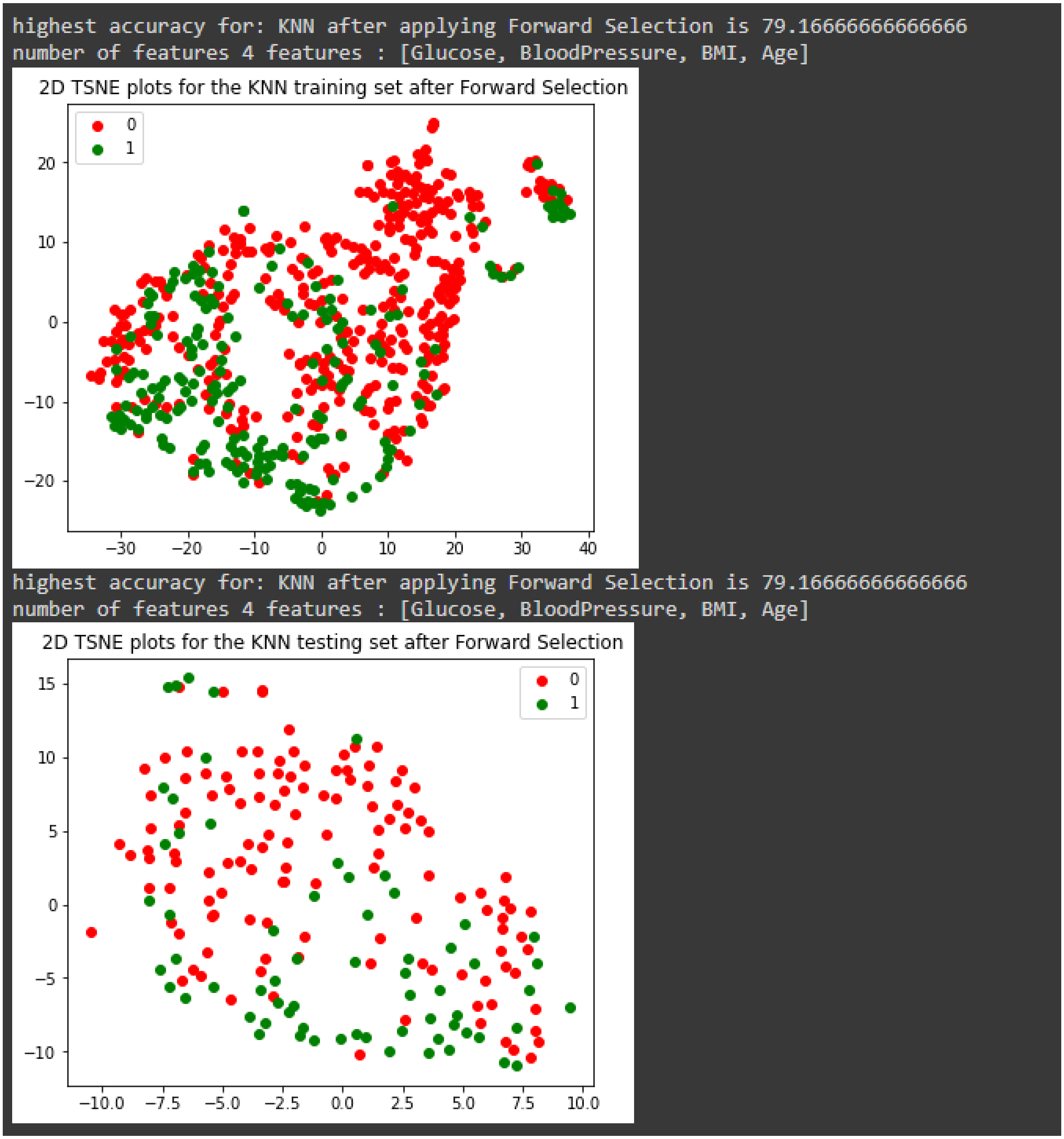

Forward Selection Wrapper Accuracies to reduce feature dimension


Forward Selection Wrapper Accuracies to reduce feature dimension


Backward Elimination Wrapper Accuracies to reduce feature dimension


Backward Elimination Wrapper Accuracies to reduce feature dimension

(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).

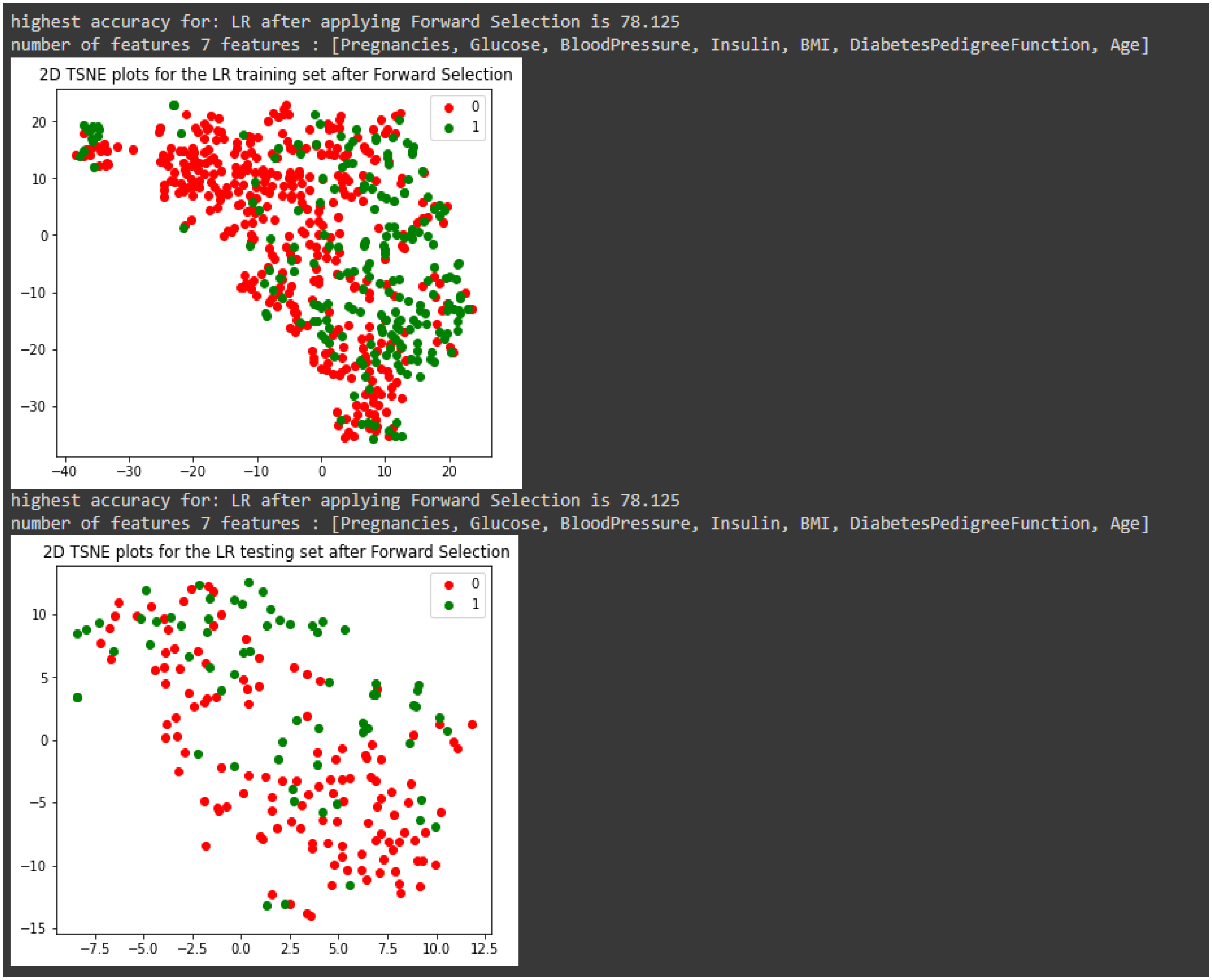


```
highest accuracy for: KNN after applying Forward Selection is 79.16666666666666
number of features 4 features : [Glucose, BloodPressure, BMI, Age]
```

2D TSNE plots for the KNN training set after Forward Selection

```
highest accuracy for: KNN after applying Forward Selection is 79.16666666666666
number of features 4 features : [Glucose, BloodPressure, BMI, Age]
```

2D TSNE plots for the KNN testing set after Forward Selection

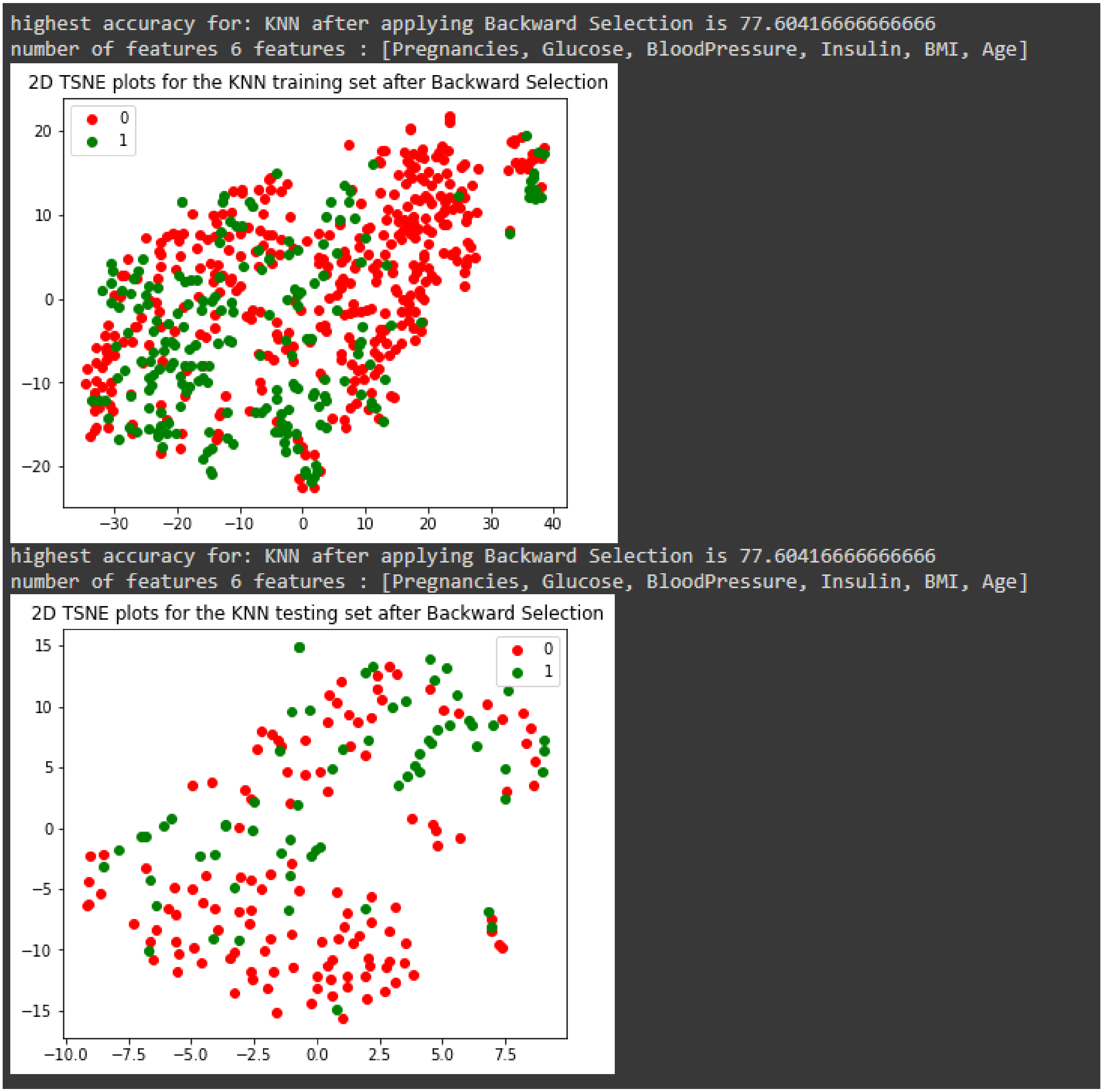


```
highest accuracy for: LR after applying Forward Selection is 78.125
number of features 7 features : [Pregnancies, Glucose, BloodPressure, Insulin, BMI, DiabetesPedigreeFunction, Age]
```

2D TSNE plots for the LR training set after Forward Selection
● 0
● 1
20
10
0
−10
−20
−30
−40
−30
−20
−10
0
10
20

```
highest accuracy for: LR after applying Forward Selection is 78.125
number of features 7 features : [Pregnancies, Glucose, BloodPressure, Insulin, BMI, DiabetesPedigreeFunction, Age]
```

2D TSNE plots for the LR testing set after Forward Selection
● 0
● 1
10
5
0
−5
−10
−15
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5

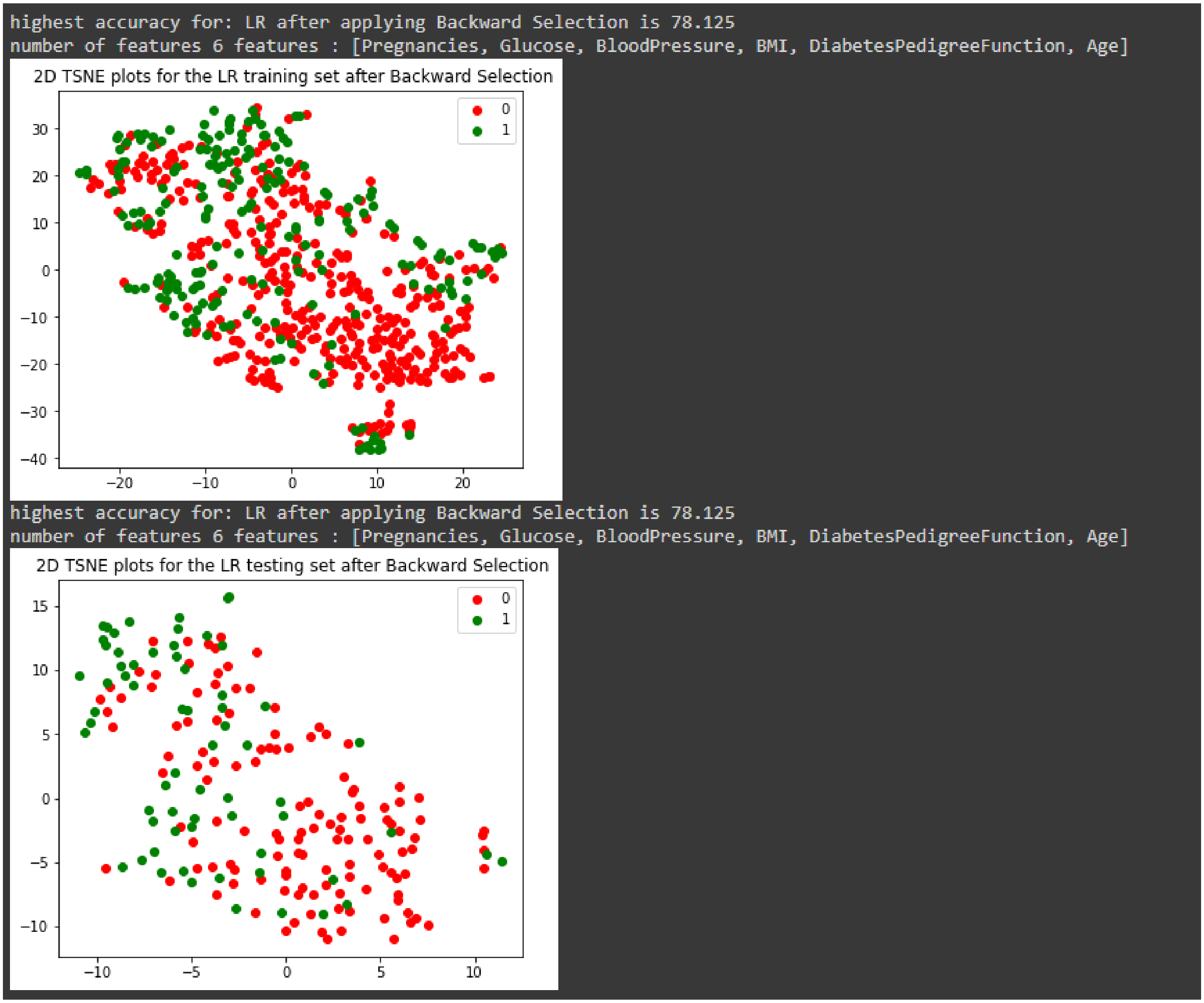


```
highest accuracy for: KNN after applying Backward Selection is 77.60416666666666
number of features 6 features : [Pregnancies, Glucose, BloodPressure, Insulin, BMI, Age]
```

2D TSNE plots for the KNN training set after Backward Selection
0
1
20
10
0
−10
−20
−30
−20
−10
0
10
20
30
40

```
highest accuracy for: KNN after applying Backward Selection is 77.60416666666666
number of features 6 features : [Pregnancies, Glucose, BloodPressure, Insulin, BMI, Age]
```

2D TSNE plots for the KNN testing set after Backward Selection
0
1
15
10
5
0
−5
−10
−15
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

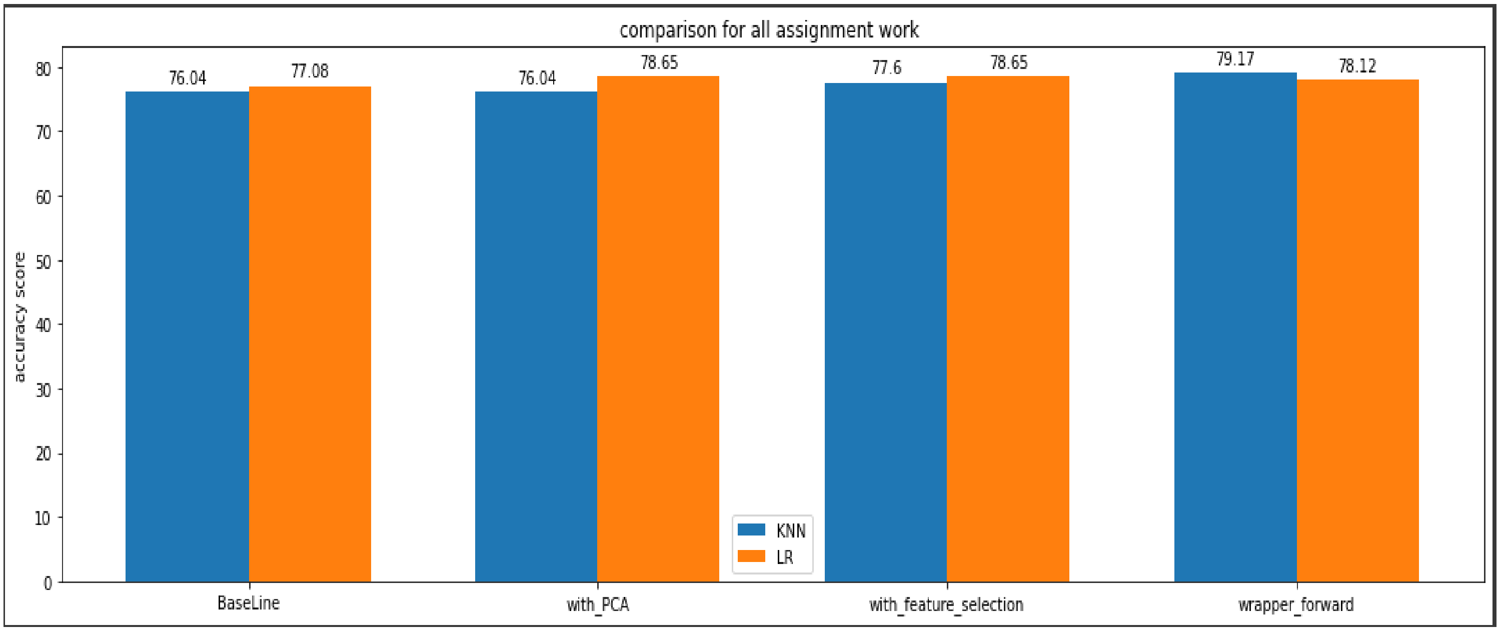


```
highest accuracy for: LR after applying Backward Selection is 78.125
number of features 6 features : [Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, Age]
```

2D TSNE plots for the LR training set after Backward Selection
0
1
30
20
10
0
−10
−20
−30
−40
−20
−10
0
10
20

```
highest accuracy for: LR after applying Backward Selection is 78.125
number of features 6 features : [Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, Age]
```

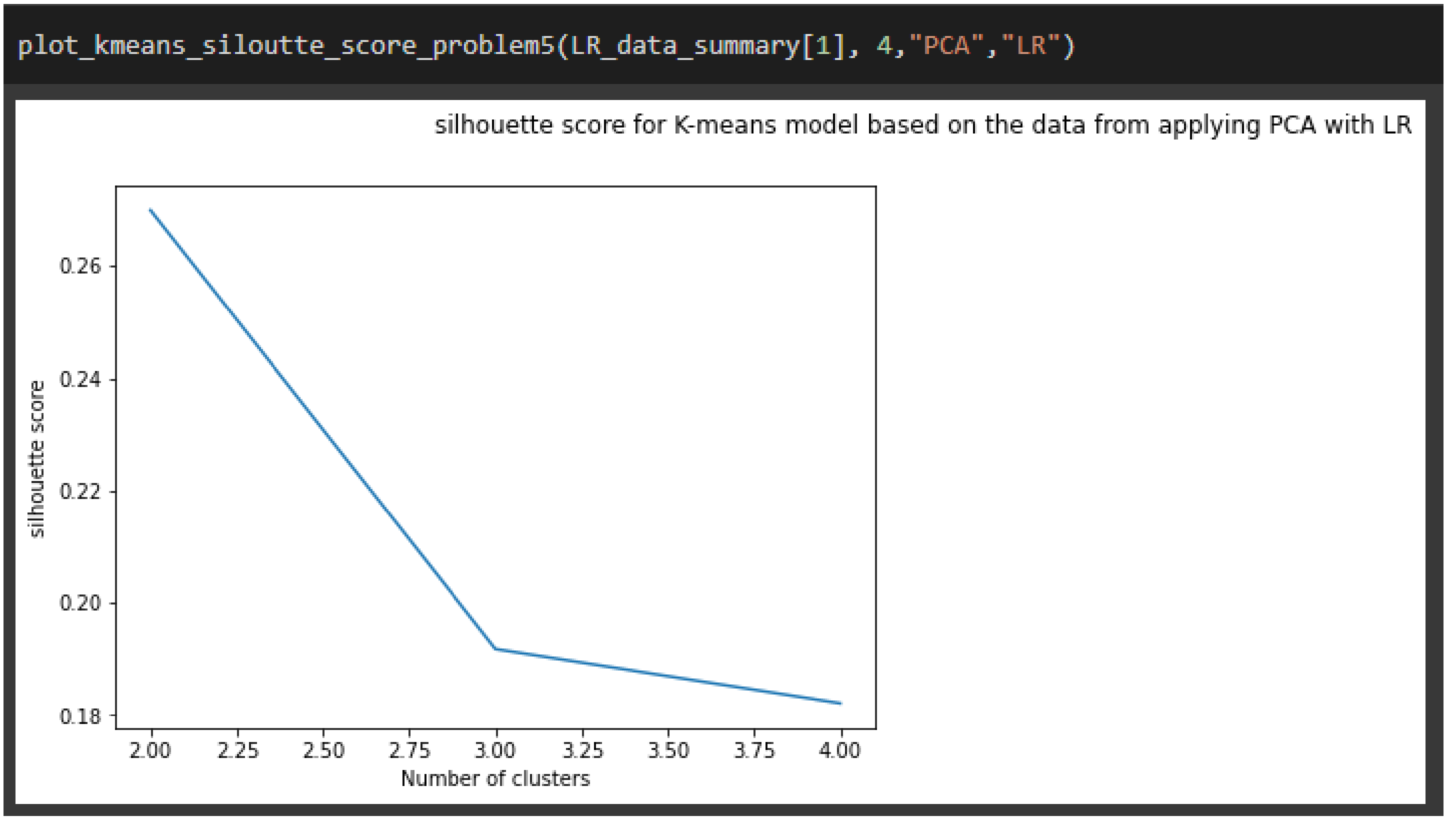
2D TSNE plots for the LR testing set after Backward Selection
0
1
15
10
5
0
−5
−10
−10
−5
0
5
10



5. Choose the best number of clusters for k-means clustering algorithm on the processed data, using the best features from Q3 or best number of dimensionalities Q4.

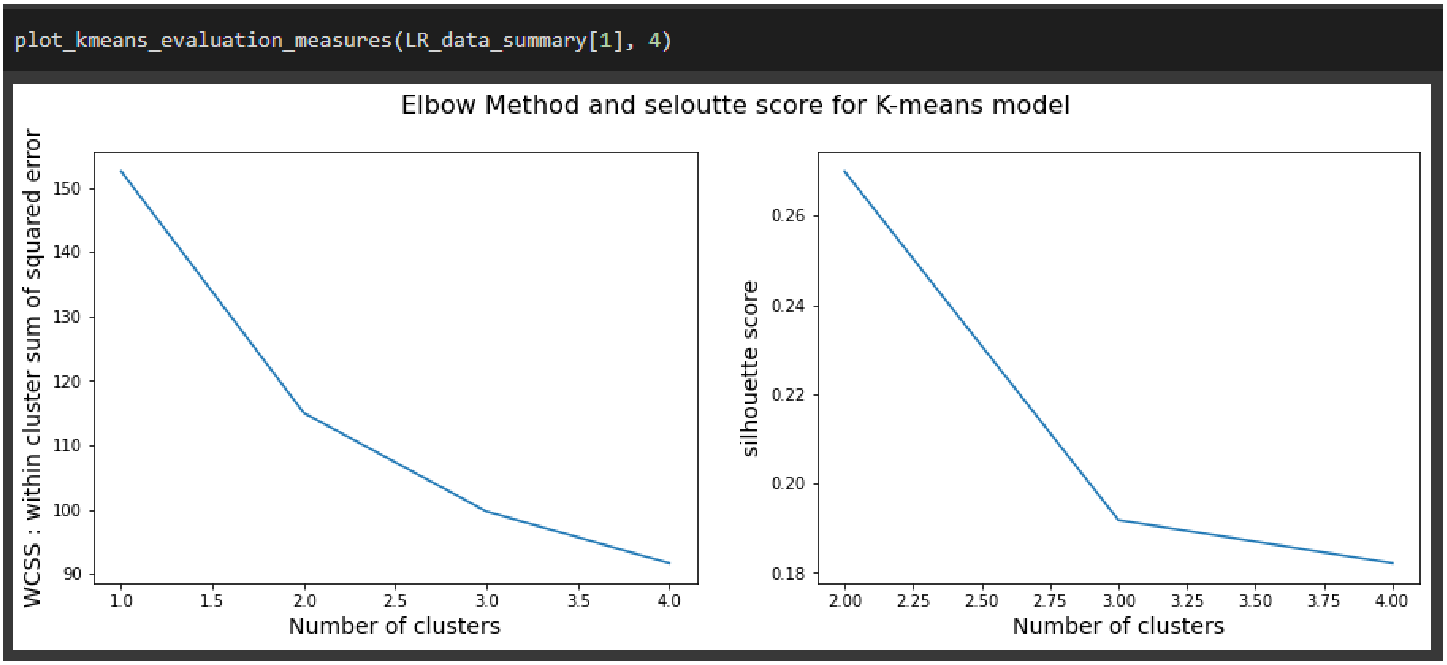
We used the improved data with PCA (LR_**data_summary** [1]) because it gave us the best accuracy with LR model

(a) Plot the silhouette score vs the number of clusters



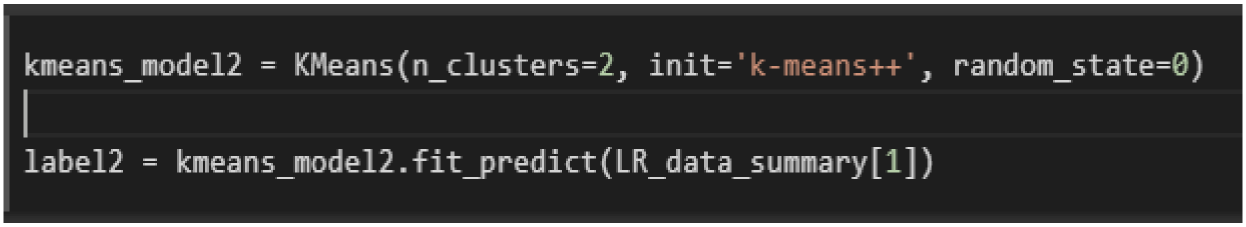
```
plot_kmeans_siloutte_score_problem5(LR_data_summary[1], 4,"PCA","LR")
```

(b) Determine the optimal number of clusters for k-means



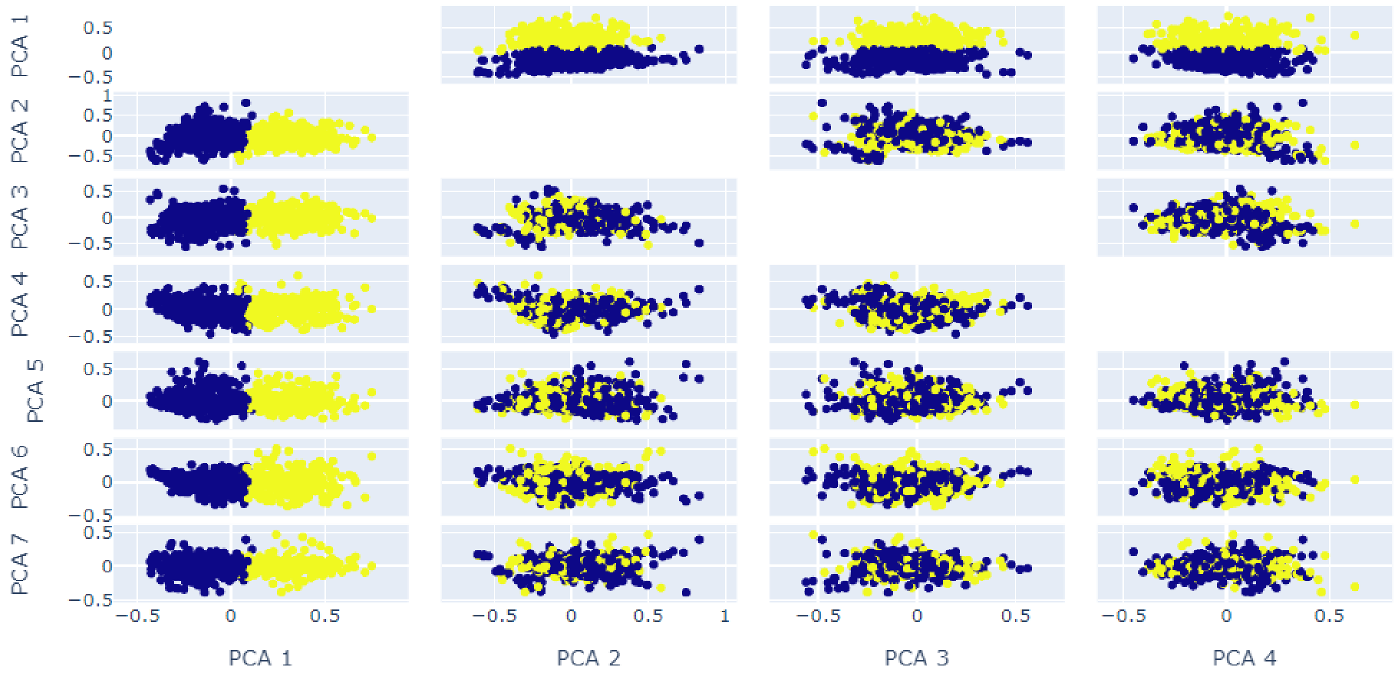
```
plot_kmeans_evaluation_measures(LR_data_summary[1], 4)
```

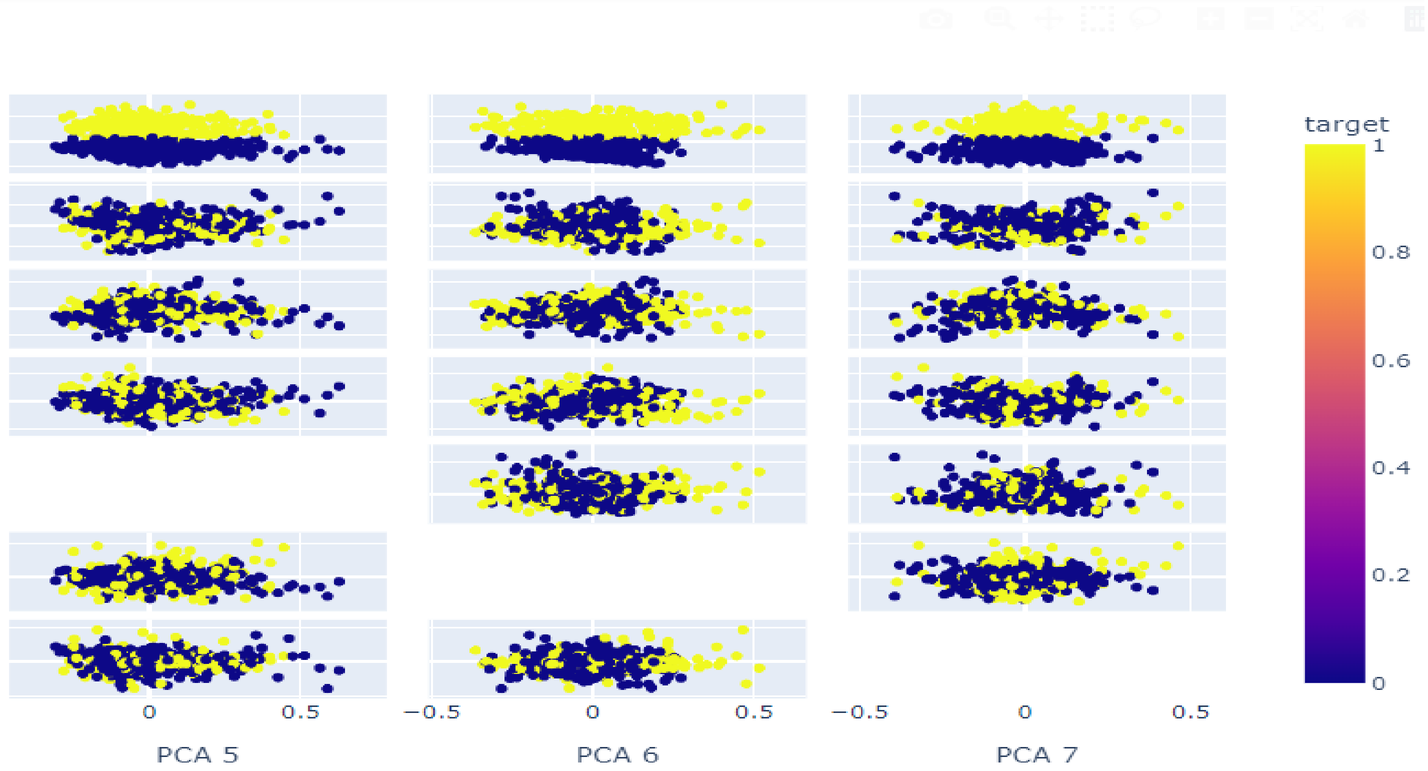
We built K-means again with the improved data to compare the results with the original data before applying PCA, because without performing this step we can’t answer the question 8 which is to comment on the result here and the results from Q1.



```
kmeans_model2 = KMeans(n_clusters=2, init='k-means++', random_state=0)

label2 = kmeans_model2.fit_predict(LR_data_summary[1])
```

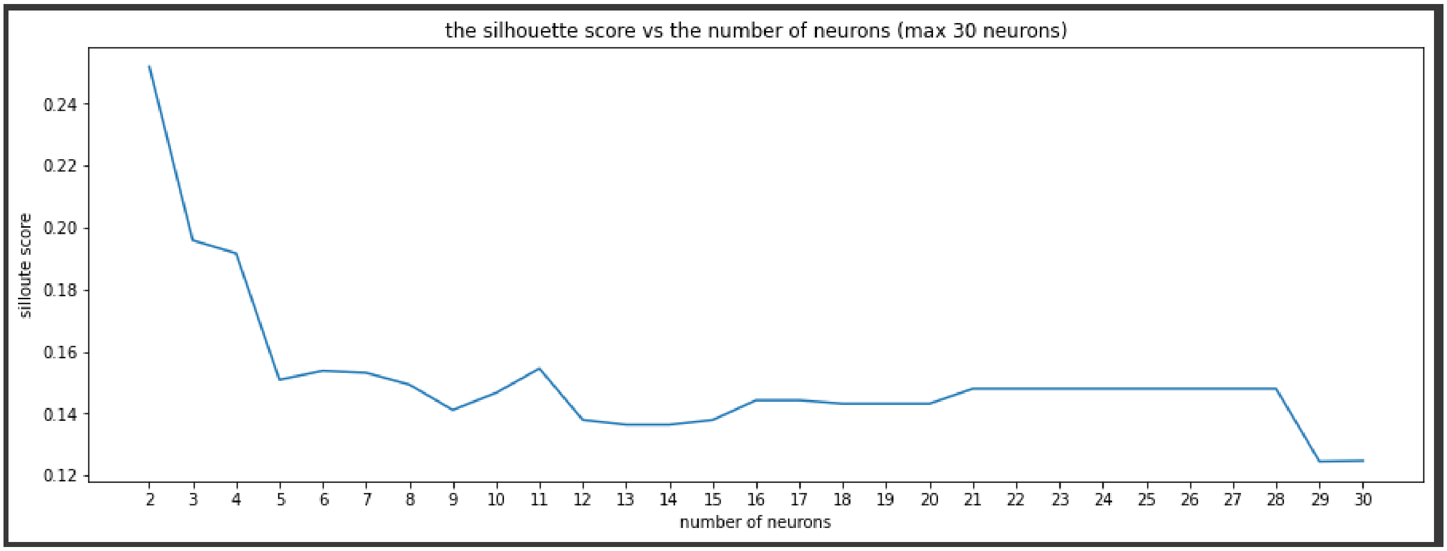
Visualize the principal components




target
1
0.8
0.6
0.4
0.2
0
0
0.5
−0.5
0
0.5
−0.5
0
0.5
PCA 5
PCA 6
PCA 7

6. Choose the best number of neurons for SOM algorithm, using the best features from Q3 or best number of dimensionalities Q4. You might find it easier if you use the MiniSom library.

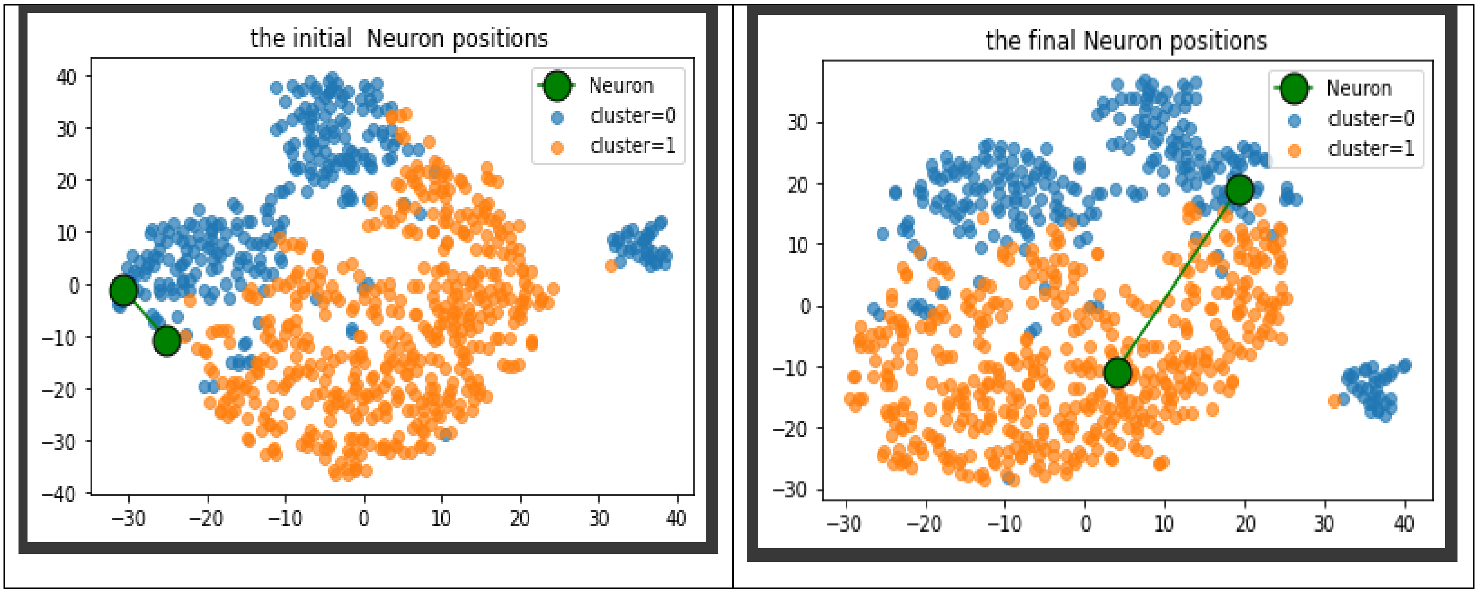
(a) Plot the silhouette score vs the number of neurons (max 30 neurons)



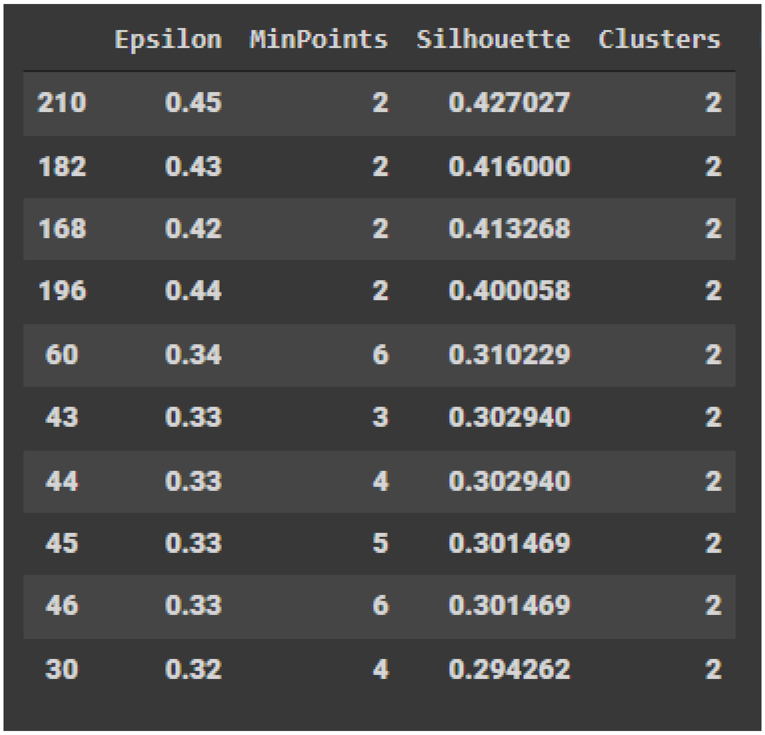
(b) Determine the optimal number of neurons for SOM

based on the curve of (number_Of_neurons vs silloute_score) the highest point on the curve refers to the optimal number of neurons which is equal to 2 in this case.

(c) Plot the initial and final Neuron positions

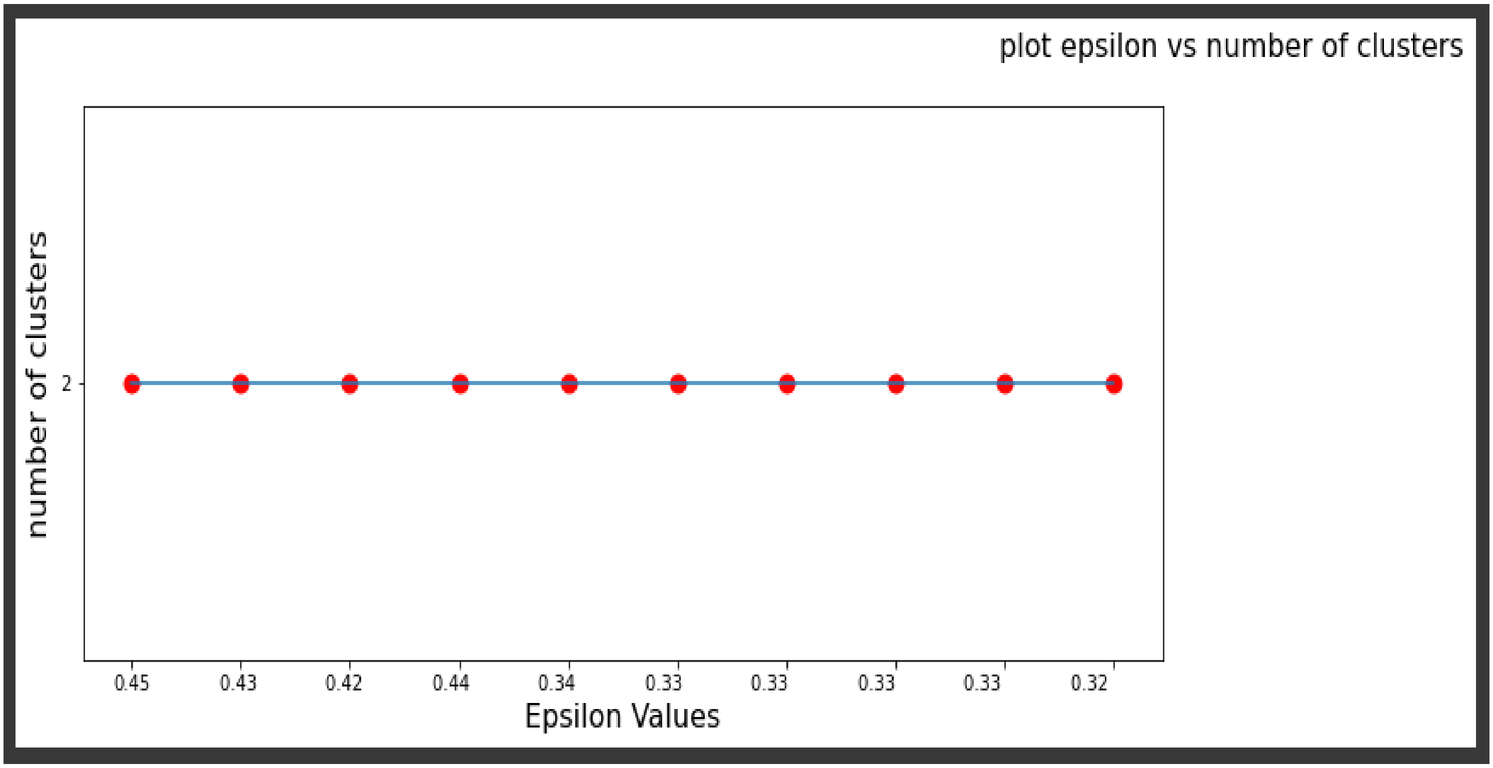


7. Tune the epsilon (0.3-0.7) and minpoints (2-15) values to obtain the same number of clusters in Q6 by using DBSCAN. Provide two separate plots; where you show only the best 10 combinations of epsilon and minpoints that brings you closer to the desired cluster number with the highest silhouette score.

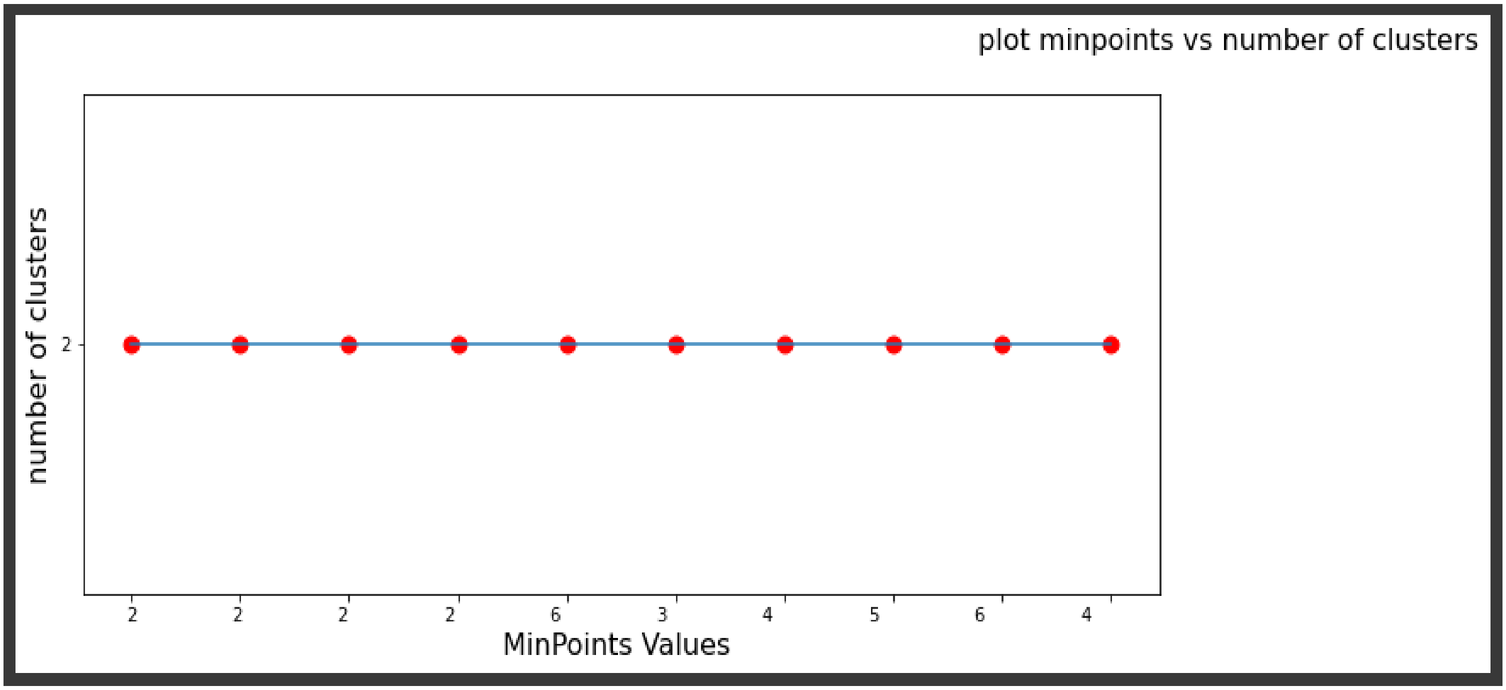


	Epsilon	MinPoints	Silhouette	Clusters
210	0.45	2	0.427027	2
182	0.43	2	0.416000	2
168	0.42	2	0.413268	2
196	0.44	2	0.400058	2
60	0.34	6	0.310229	2
43	0.33	3	0.302940	2
44	0.33	4	0.302940	2
45	0.33	5	0.301469	2
46	0.33	6	0.301469	2
30	0.32	4	0.294262	2

(a) First plot is epsilon vs number of clusters.



(b) Second plot is minpoints vs number of clusters.

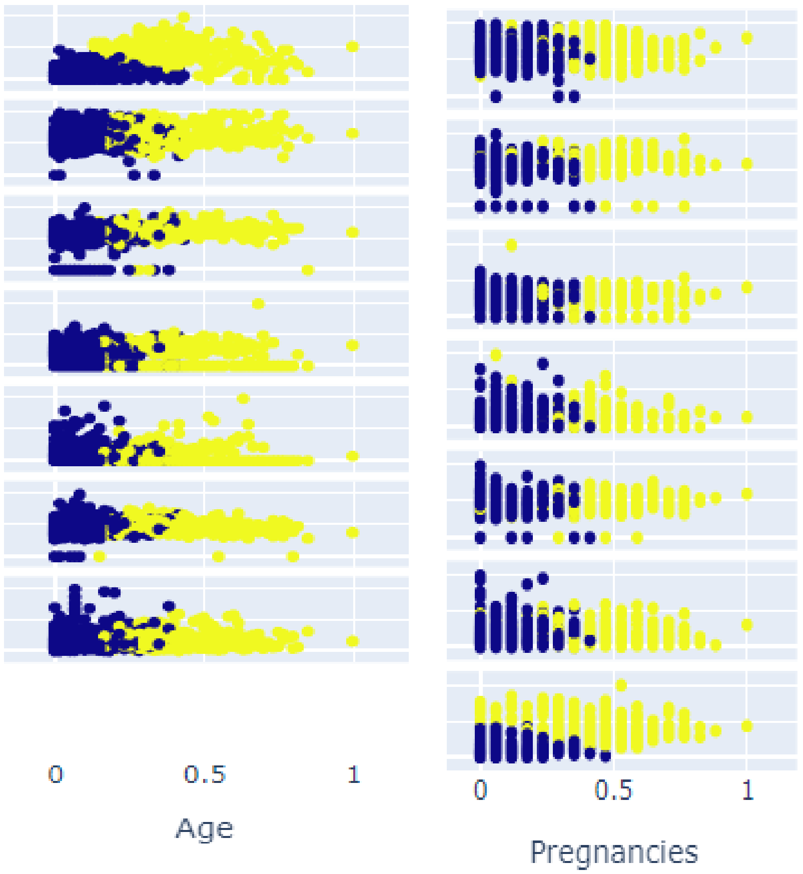


8. In your report, make sure to include a conclusion section where you comment on the followings:

(a) The results of K-means from Q2 and Q5 (before and after applying DR methods and FS).

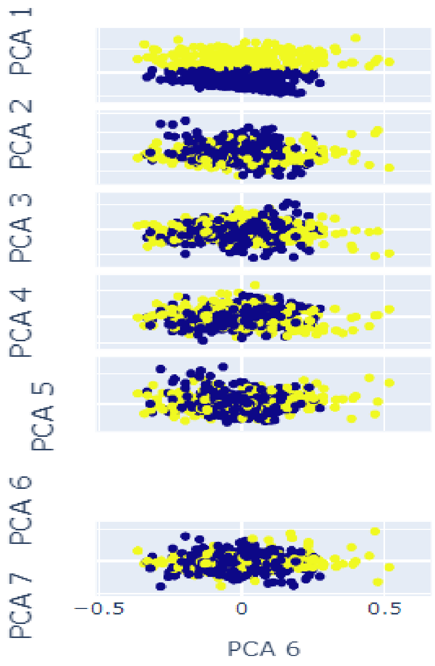
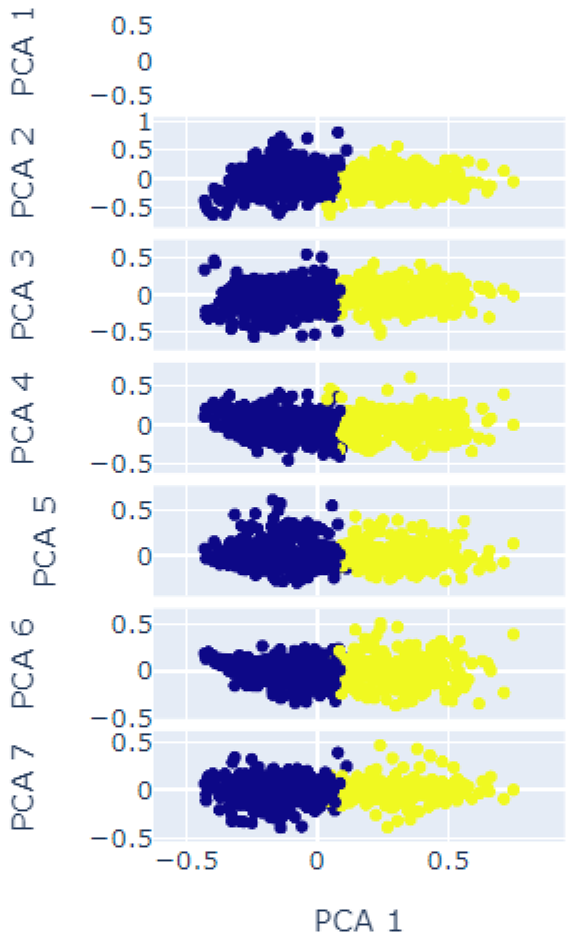
In Q2

We used px.scatter.matrix to visualize all the original dimensions, the dimensions here are our original features because we have not performed any dimensionality reduction technique yet. Notice that some of the combinations of two features is able to represent our two cluster in the data like:

	The **age** feature can represent our clusters, whatever the second feature that we will use with it to represent the data. Also, the **Pregnancies** with any other feature is good to represent the clusters in our dataset. But **AGE** still better. But there are some features we cannot use to represent the clusters in our dataset like **BMI** feature.

In Q5

we apply PCA the same dataset, and retrieve **7** components because we have already passed this number to the function while constructing the PCA. We use the same px.scatter_matrix trace to display our results, but this time our features are the resulting *principal components*, ordered by how much variance they are able to explain.

	The subplot between PC3 and PC6 is clearly unable to separate each class
	the subplot between PC1 and PC2 shows a clear separation between classes in our dataset.

(b) The results of TSNE plots from Q1, Q3 and Q4.

the original data in Q1 was hardly separable because the data variation within the class was more significant than the variation between classes.

but it got smoother with the improved data from PCA in Q3.

with feature selection, the data dimensionality is also reduced but unlike the feature extraction technique, the number of features is reduced too.

the result of applying T-SNE with the selected feature in our case has a no bigger effect on the visualization results.

References:

[1] https://www.kaggle.com/datasets/mathchi/diabetes-data-set

[2] https://plotly.com/python/pca-visualization/