ELG 5255: Applied Machine Learning

Assignment 3

Group23_HW3

Part1: Calculations

1(a)

Step 1: we will measure the distance (using Euclidean distance) between the selected two centroids (A2 and A4).

Iteration 1

Points	A2=(5,8)	A4=(1,2)
A1=(2,5)	$\sqrt{(5-2)^2+(8-5)^2}$	$\sqrt{(1-2)^2+(2-5)^2}$
A2=(5,8)	$\sqrt{(5-5)^2+(8-8)^2}$	$\sqrt{(1-5)^2+(2-8)^2}$
A3=(7,5)	$\sqrt{(5-7)^2+(8-5)^2}$	$\sqrt{(1-7)^2+(2-5)^2}$
A4=(1,2)	$\sqrt{(5-1)^2+(8-2)^2}$	$\sqrt{(1-1)^2+(2-2)^2}$
A5=(4,9)	$\sqrt{(5-4)^2+(8-9)^2}$	$\sqrt{(1-4)^2+(2-9)^2}$

Which equal to…

Points	A2=(5,8)	A4=(1,2)
A1=(2,5)	4.242641	3.162278
A2=(5,8)	0	7.211103
A3=(7,5)	3.605551	6.708204
A4=(1,2)	7.211103	0
A5=(4,9)	1.414214	7.615773

Step 2: And based on the minimum distance between each point and the two clusters we will cluster each point to Its cluster.

Points	A2=(5,8) -> cluster 1	A4=(1,2) -> cluster 2	Clusters
A1=(2,5)	4.242641	3.162278	2
A2=(5,8)	0	7.211103	1
A3=(7,5)	3.605551	6.708204	1
A4=(1,2)	7.211103	0	2
A5=(4,9)	1.414214	7.615773	1

Step 3: now we will calculate our new clusters centroids.

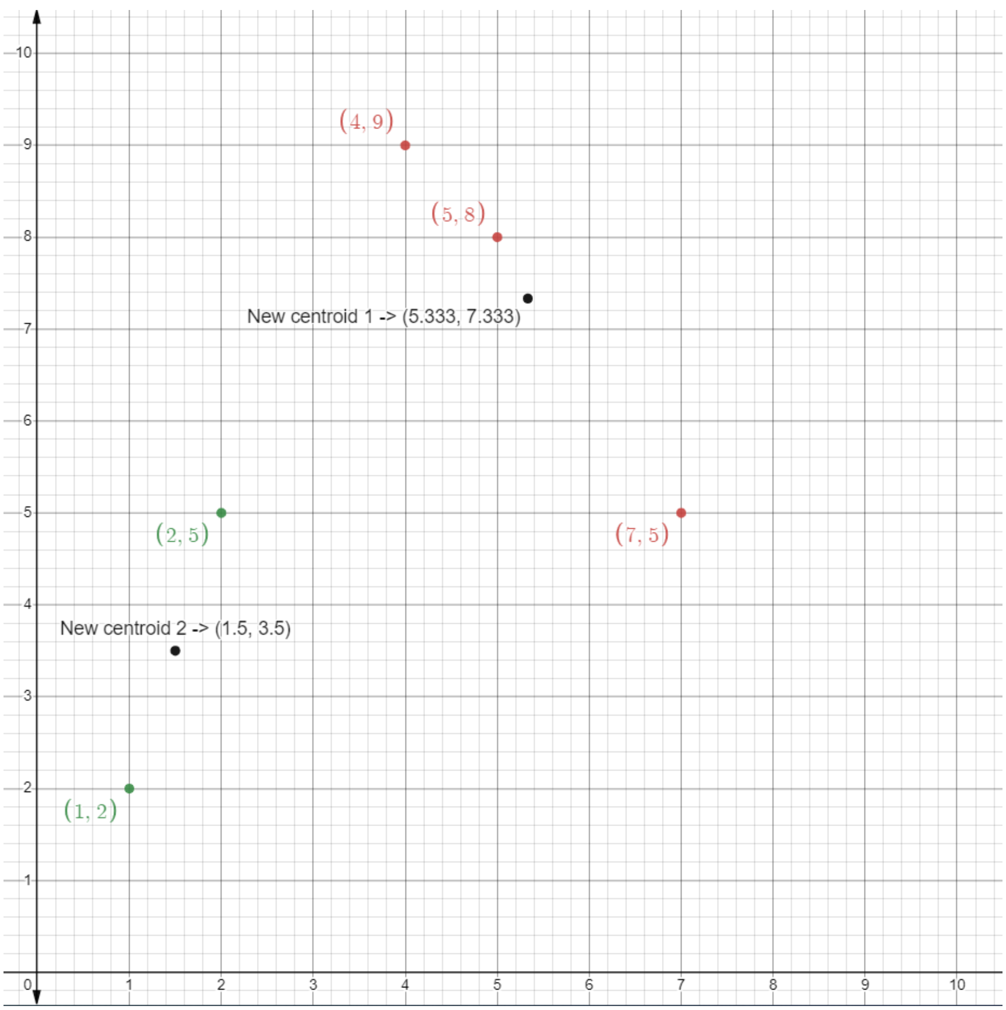
Cluster 1 = A2=(5,8) , A3=(7,5) , A5=(4,9)

New centroid 1 = ((5+7+4)/3,(8+5+9)/3) **-> (5.333, 7.333)**

Cluster 2 = A1=(2,5), A4=(1,2)

New centroid 2 = ((2+1)/2, (5+2)/2) -> (1.5, 3.5)

1(b)



1(c)

Part 1

silhouette score has two parts which is: -

cohesion: and it is referring to the average distance between an instance (sample) and all other data points within the same cluster.

And

Separation: and it is referring to the average distance between an instance (sample) and all other data points in other clusters.

Step 1: now we will calculate **cohesion** score for all the datapoints using Euclidian distance.

Cluster 1 -> A2=(5,8) , A3=(7,5) , A5=(4,9)

Points	**A2=(5,8)**	**A3=(7,5)**	**A5=(4,9)**
A2=(5,8)	0	$\sqrt{(5-7)^2+(8-5)^2}$	$\sqrt{(5-4)^2+(8-9)^2}$
A3=(7,5)	$\sqrt{(5-7)^2+(8-5)^2}$	0	$\sqrt{(7-4)^2+(5-9)^2}$
A5=(4,9)	$\sqrt{(5-4)^2+(8-9)^2}$	$\sqrt{(7-4)^2+(5-9)^2}$	0

Which equal to…

Points	**A2=(5,8)**	**A3=(7,5)**	**A5=(4,9)**
A2=(5,8)	0	3.605551	1.414214
A3=(7,5)	3.605551	0	5
A5=(4,9)	1.414214	5	0

Note* we will not take the zeroes into consideration.

Cohesion score for A2=(5,8) = (3.605551 + 1.414214)/2 = **2.509**

Cohesion score for A3=(7,5) = (3.605551 + 5)/2 = **4.302**

Cohesion score for A5=(4,9) = (1.414214 + 5)/2 = **3.207**

Cluster 2 -> A1=(2,5), A4=(1,2)

Points	A1=(2,5)	A4=(1,2)
A1=(2,5)	0	$\sqrt{(2-1)^2+(5-2)^2}$
A4=(1,2)	$\sqrt{(2-1)^2+(5-2)^2}$	0

Which equal to…

Points	A1=(2,5)	A4=(1,2)
A1=(2,5)	0	3.162278
A4=(1,2)	3.162278	0

Cohesion score for A1=(2,5) = (3.162278)/1 = **3.162278**

Cohesion score for A4=(1,2) = (3.162278)/1 = **3.162278**

Step 2: now we will calculate Separation score for all the datapoints using Euclidian distance.

A1=(2,5) is in cluster 2, we will calculate the distance between this point and all the points in cluster 1 which contain these points -> **A2=(5,8) , A3=(7,5) , A5=(4,9).**

Points	A2=(5,8)	A3=(7,5)	A5=(4,9)
A1=(2,5)	4.242641	5	4.472136

Separation score for A1=(2,5) = (4.242641 + 5 + 4.472136)/3 = **4.571.**

We will do the same for A4=(1,2) which is in cluster 2

Points	A2=(5,8)	A3=(7,5)	A5=(4,9)
A4=(1,2)	7.211103	6.708204	7.615773

Separation score for A4=(1,2) = (7.211103 + 6.708204 + 7.615773)/3 = **7.178**

Now we will do the same process for the other points that **in Cluster 1 -> A2=(5,8) , A3=(7,5) , A5=(4,9)**

Points	**A1=(2,5)**	**A4=(1,2)**
A2=(5,8)	4.242641	7.211103

Separation score for A2=(5,8) = (4.242641 + 7.211103)/2 **= 5.726**

Points	**A1=(2,5)**	**A4=(1,2)**
A3=(7,5)	5	6.708204

Separation score for A3=(7,5) = (5 + 6.708204)/2 **= 5.854**

Points	**A1=(2,5)**	**A4=(1,2)**
A5=(4,9)	4.472136	7.615773

Separation score for A5=(4,9) = (4.472136 + 7.615773)/2 = **6.043**

Step 3: now we will calculate the overall **Silhouette** score.

Now using the Silhouette equation which is **(b - a)/max(a, b)**

Cohesion score -> a

Separation score -> b

For point A1: -

Cohesion score for A1=(2,5) = (3.162278)/1 = **3.162278**

Separation score for A1=(2,5) = (4.242641 + 5 + 4.472136)/3 = **4.571**

(4.571 - 3.162278)/max(3.162278, 4.571) = 1.408722 / 4.571 **= 0.308**

For point A2: -

Cohesion score for A2=(5,8) = (3.605551 + 1.414214)/2 = **2.509**

Separation score for A2=(5,8) = (4.242641 + 7.211103)/2 = **5.726**

(5.726 - 2.509)/max(2.509, 5.726) = 3.217 / 5.726 **= 0.561**

For point A3: -

Cohesion score for A3=(7,5) = (3.605551 + 5)/2 = **4.302**

Separation score for A3=(7,5) = (5 + 6.708204)/2 = **5.854**

(5.854 - 4.302)/max(4.302, 5.854) = 1.552 / 5.854 = **0.265**

For point A4: -

Cohesion score for A4=(1,2) = (3.162278)/1 = **3.162278**

Separation score for A4=(1,2) = (7.211103 + 6.708204 + 7.615773)/3 = **7.178**

(7.178 - 3.162278)/max(3.162278, 7.178) = 4.015722 / 7.178 = **0.559**

For point A5: -

Cohesion score for A5=(4,9) = (1.414214 + 5)/2 = **3.207**

Separation score for A5=(4,9) = (4.472136 + 7.615773)/2 = **6.043**

(6.043 - 3.207)/max(3.207, 6.043) = 2.836 / 6.043 = **0.469**

Overall Silhouette score = (0.308 + 0.561 + 0.265 + 0.559 + 0.469)/5 **= 0.433**

Part 2

WSS score is a measure of the variability of the observations within each cluster.

$$WSS = \sum (x_i - c_i)^2$$

Step 1: we will measure the distance between each point in the same cluster with the cluster centroid.

The new centroid of cluster 1 is -> **(5.333, 7.333)**

Points of cluster 1 -> **A2=(5,8) , A3=(7,5) , A5=(4,9)**

Points	**(5.333, 7.333)**
A2=(5,8)	$(5-5.333)^2 + (8-7.333)^2$ = 0.555778
A3=(7,5)	$(7-5.333)^2 + (5-7.333)^2$ = 8.221778
A5=(4,9)	$(4-5.333)^2 + (9-7.333)^2$ = 4.555778

WSS score for cluster 1 = (0.555778 + 8.221778 + 4.555778) = **13.333**

The new centroid of cluster 2 is **-> (1.5, 3.5)**

Points of cluster 2 **-> A1=(2,5), A4=(1,2)**

Points	**(1.5, 3.5)**
A1=(2,5)	$(2 - 1.5)^2 + (5 - 3.5)^2$ = 2.5
A4=(1,2)	$(1 - 1.5)^2 + (2 - 3.5)^2$ = 2.5

WSS score for cluster 2 = (2.5 + 2.5) **= 5**

Step 2: we will calculate the WSS overall score.

Overall WSS Score = 13.333 + 5 = **18.333**

Part2: Programming

1(a)

- **First we split our dataset to 75% for training and 25% test**

```
@staticmethod
def Split(X,Y, TestSize , random=0):
    x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=TestSize, random_state=random)
    x_train = x_train.reset_index(drop=True)
    x_test = x_test.reset_index(drop=True)
    y_train = y_train.reset_index(drop=True)
    y_test = y_test.reset_index(drop=True)
    return x_train, x_test, y_train, y_test
```

```
#--------------------------------------------------------------------Main-------------------------

# Create Object from our class
obj = Assignment3()

# Read Dataset
Data = obj.readDataSet('Assignment3_dataset.csv', 'Assignment3_dataset')
X = Data.iloc[:, :8]
Y = Data.iloc[:, [8]]

#----------------------Q1----------------------
x_train, x_test, y_train, y_test = obj.Split(X,Y,0.25,0)
```

x_test	DataFrame	(192, 8)
x_train	DataFrame	(576, 8)
y_test	DataFrame	(192, 1)
y_train	DataFrame	(576, 1)

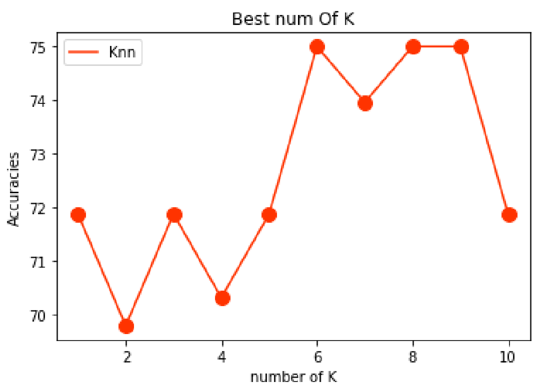
- **Here we implement LR and K-NN**

```python
#----------------------1(a)----------------------
KnnAcc = []
for i in range(1,11):
    KnnReport, Y_pred_Knn, Knn = obj.KNN(x_train,x_test,y_train,y_test,i)
    KnnAcc.append(obj.AccuracyTest(y_test,Y_pred_Knn))

# Plot the best Accuracies based number of neighbors
obj.Plot([1,2,3,4,5,6,7,8,9,10],KnnAcc,'Knn','#FF3300', 'o' , 100 , 'number of K' , 'Accuracies' , 'Best num Of K')

# Knn and Log
KnnReport, Y_pred_Knn, Knn = obj.KNN(x_train,x_test,y_train,y_test,6)
LogReport, Y_pred_Log, Log = obj.Logistic(x_train,x_test,y_train,y_test)
```

- **Regarding the plot the best number of K is 6**



- **Providing accuracy of LR and K-NN**

Text editor - LogReport

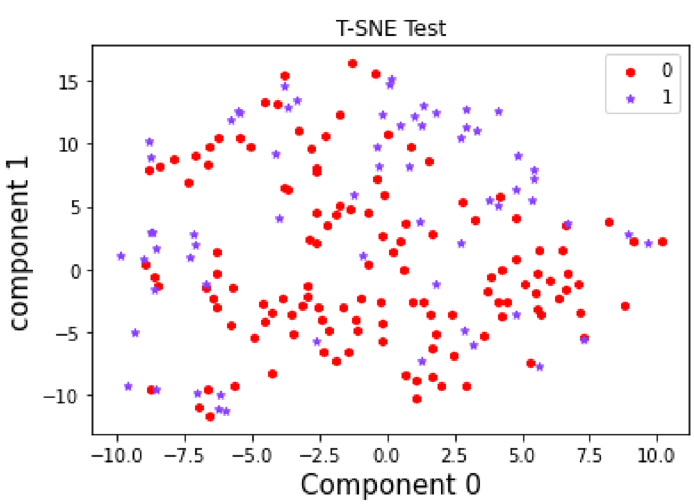
	precision	recall	f1-score	support
0	0.79	0.87	0.83	127
1	0.69	0.55	0.62	65
accuracy			0.77	192
macro avg	0.74	0.71	0.72	192
weighted avg	0.76	0.77	0.76	192

Text editor - KnnReport

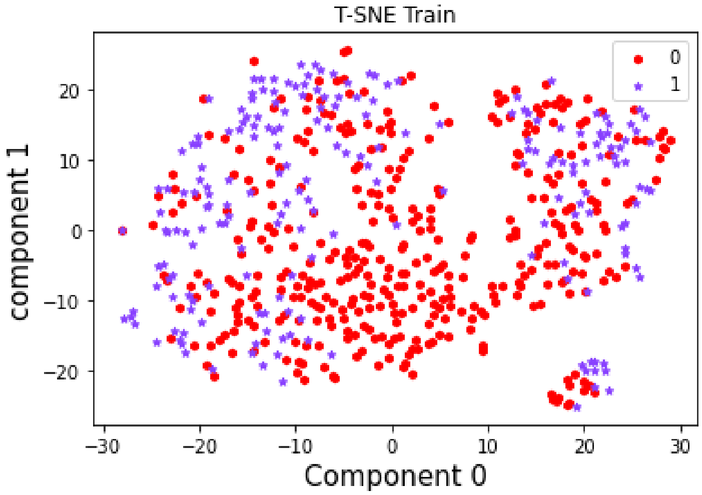
	precision	recall	f1-score	support
0	0.77	0.88	0.82	127
1	0.68	0.49	0.57	65
accuracy			0.75	192
macro avg	0.73	0.69	0.70	192
weighted avg	0.74	0.75	0.74	192

1(b)

- **Providing TSNE for testing set**

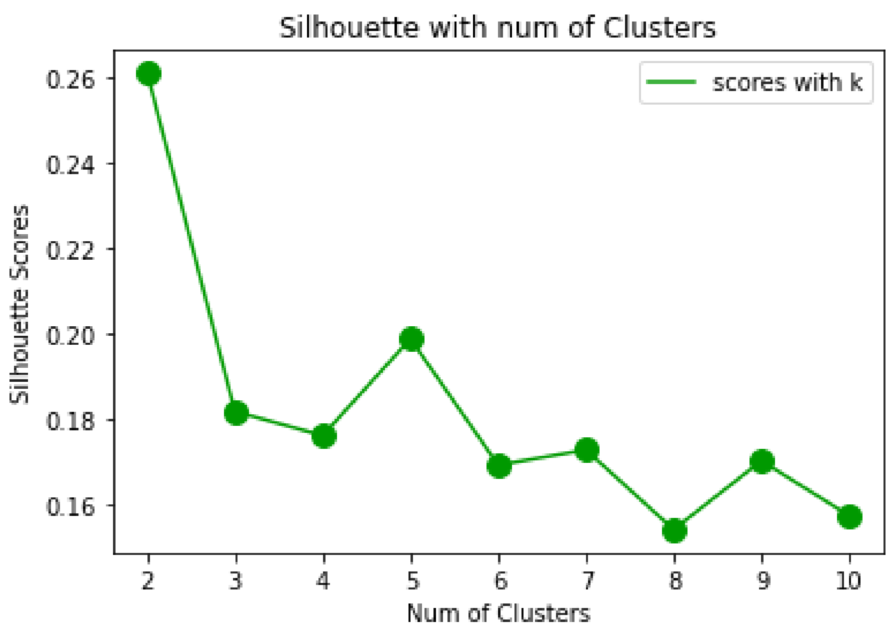


- **Providing TSNE for Training set**



2(a)

- **Using silhouette score to find the best number if K by plotting silhouette score with number of clusters**

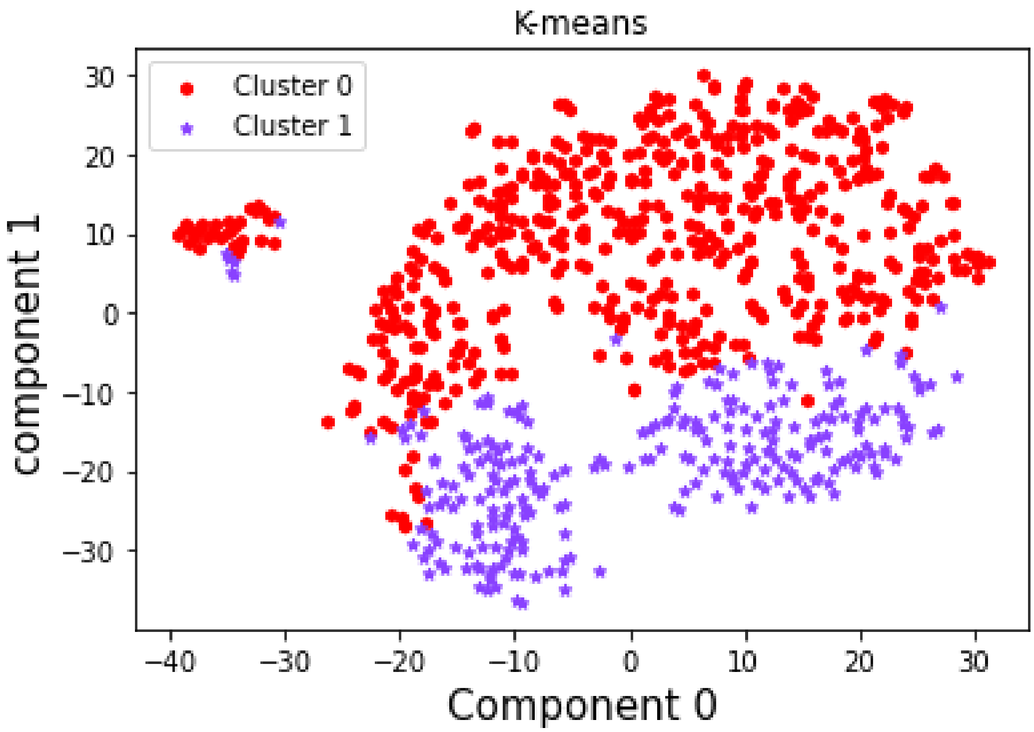


2(b)

The best number of k is = 2, because it's having the highest silhouette score which is = 0.261146

2(c)

- **After we choose k= 2 we are plotting the clustered data with it**

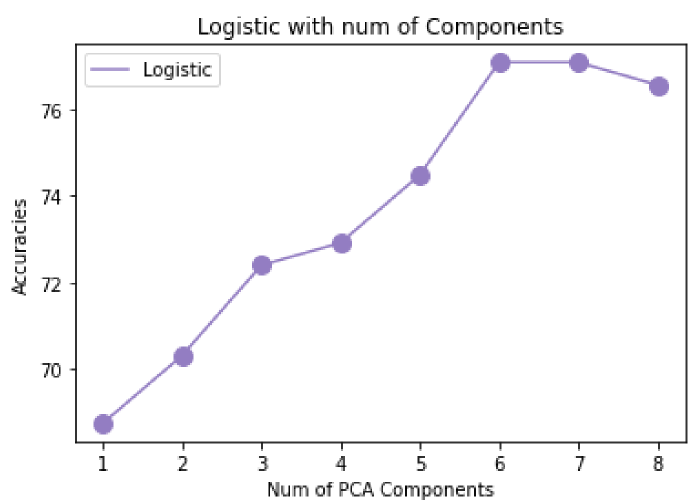


3(a)

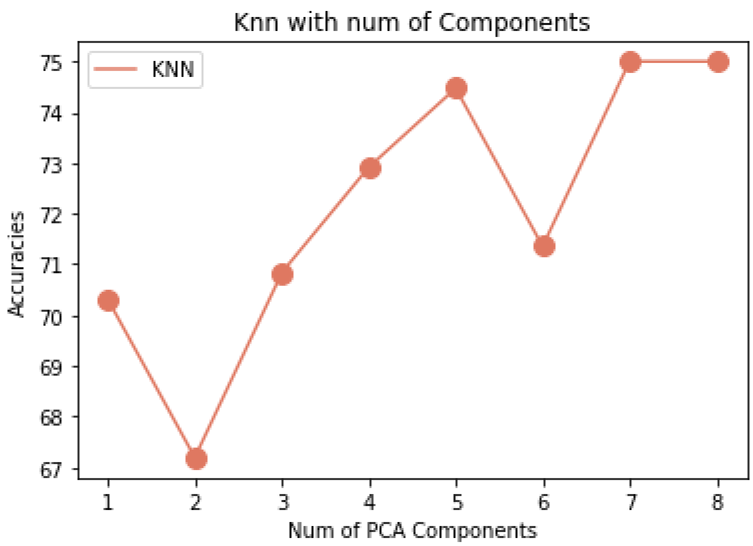
- **7 components were the highest accuracy for both KNN and logistic regression**

3(b)

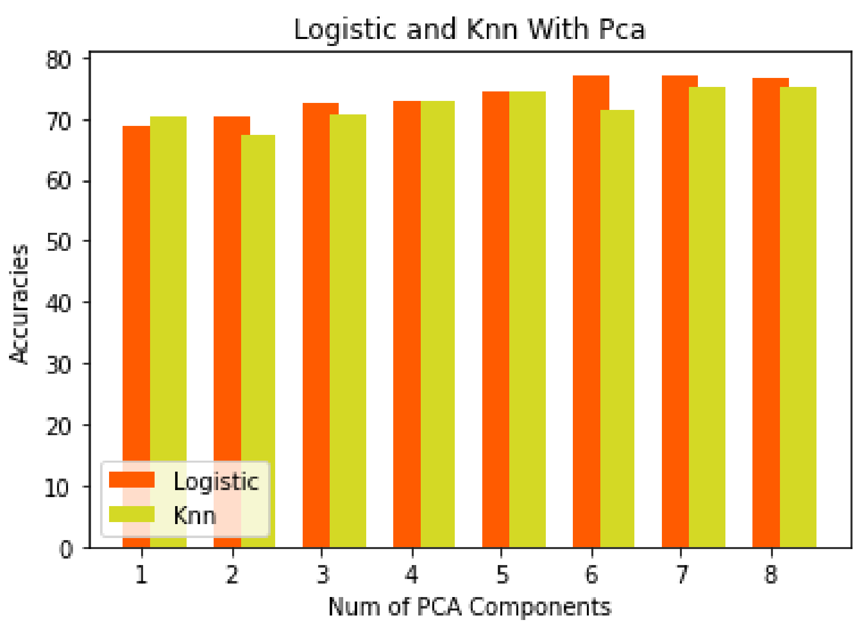
- **Graph for LR of number of PCA component and accuracies**



- **Graph for KNN of number of PCA component and accuracies**

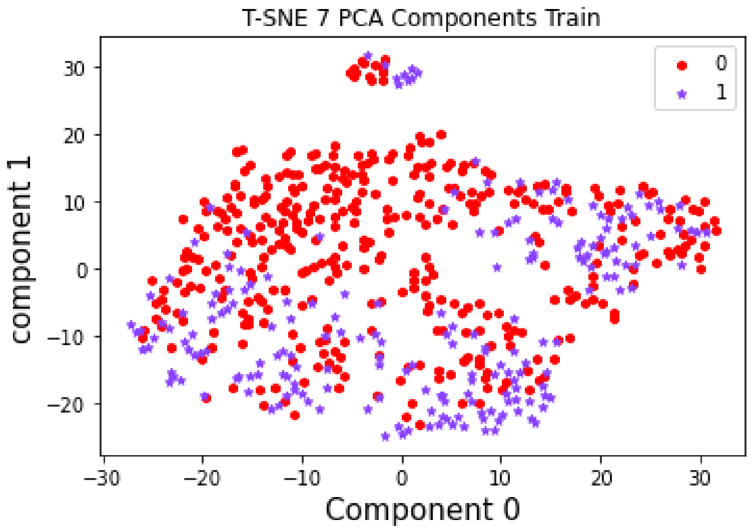


- **For both KNN and LR**

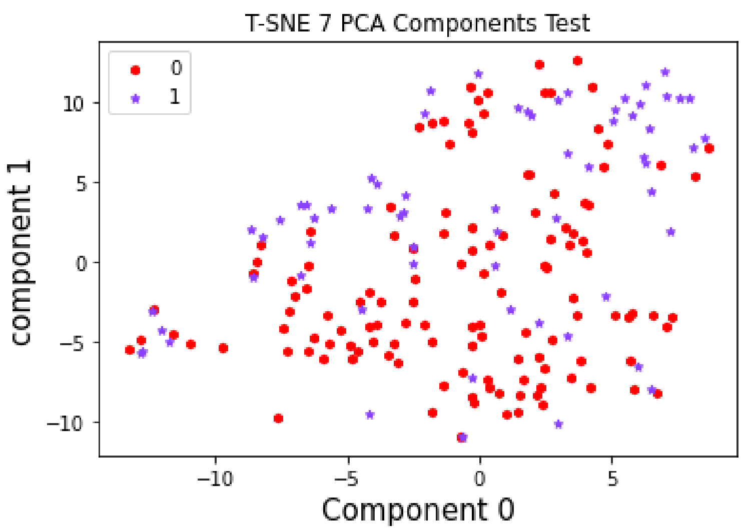


3(c)

- **Providing 2D TSNE for training set**

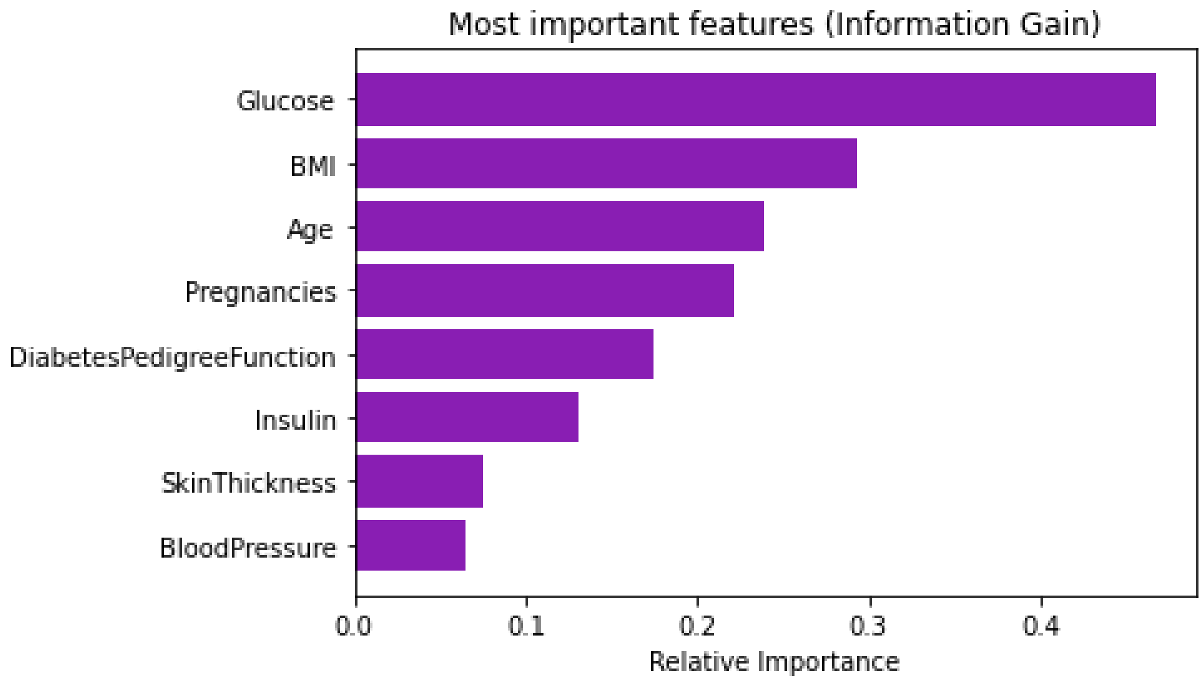


- **Providing 2D TSNE for testing set**

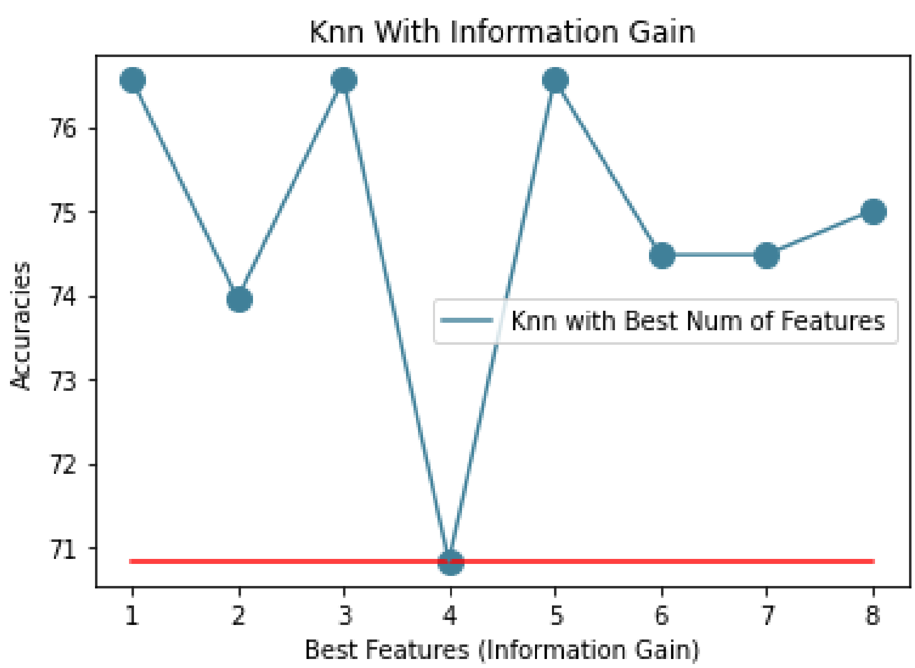


4(a)

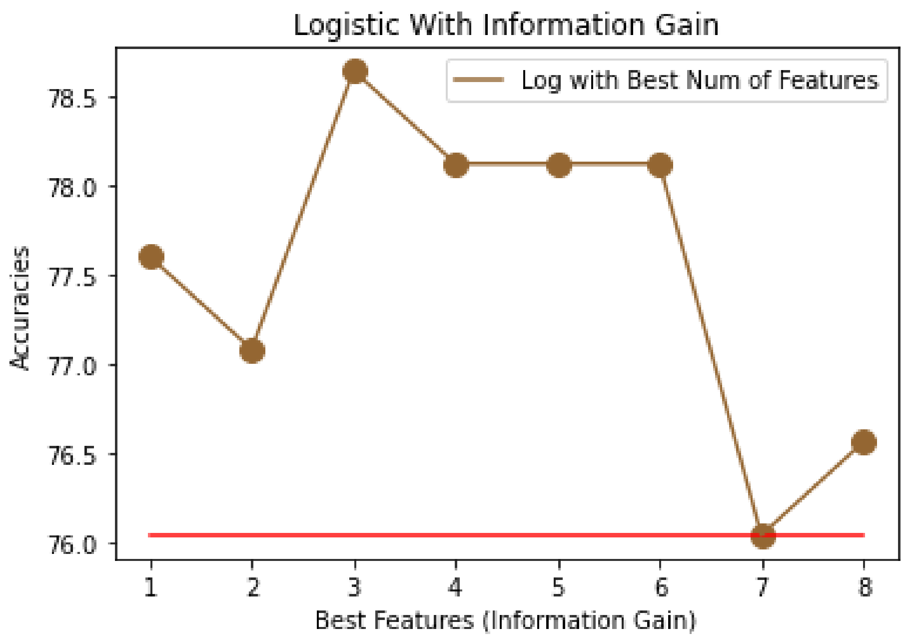
- **Identify input features having high correlation with target variable.**



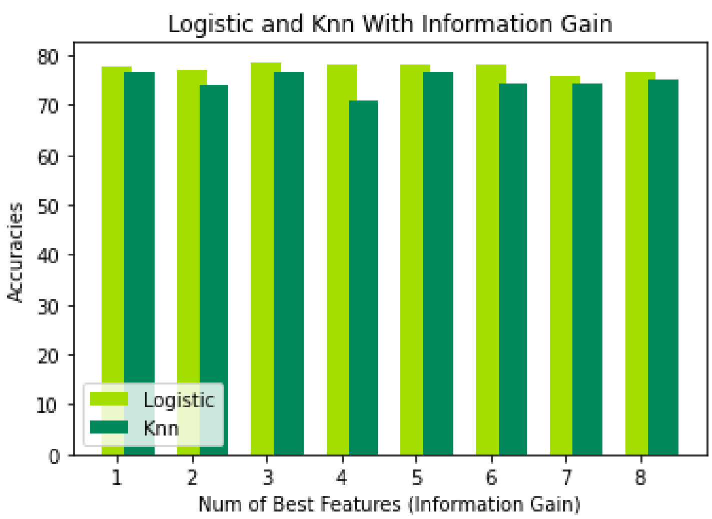
- **Apply Filter Method Information Gain with KNN and the best number of features 5**



- **Apply Filter Method Information Gain with LR and the best feature is 3**

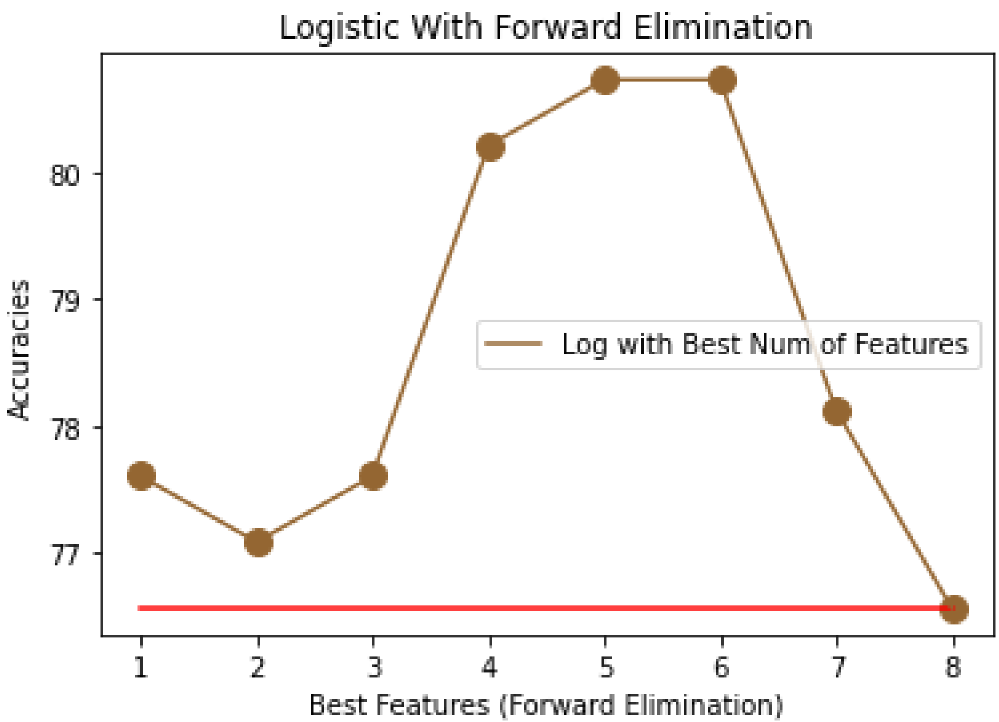


- **For both KNN and LR**

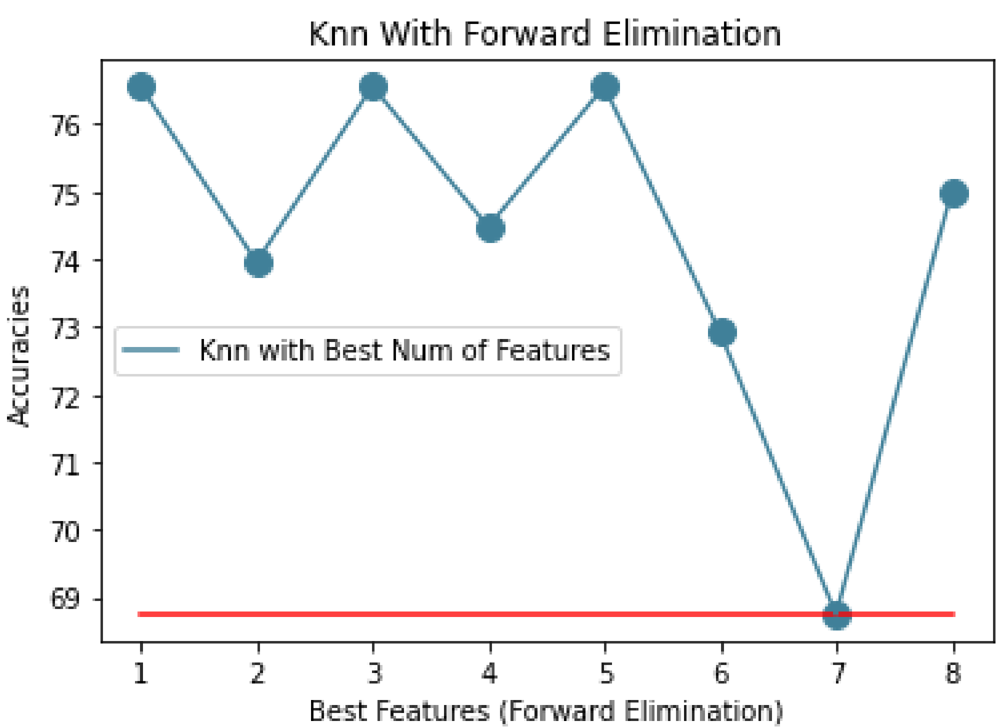


4(b)

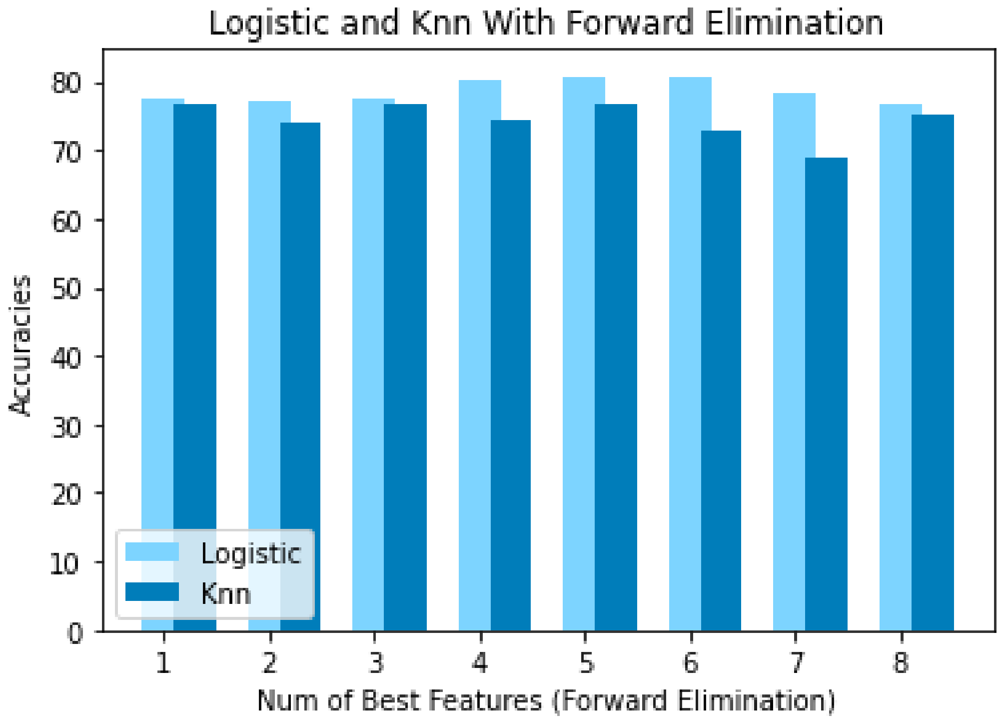
- **Apply Wrapper Method forward elimination with LR and the best feature is 5**



- **Apply Wrapper Method forward elimination with KNN and the best feature is 5**

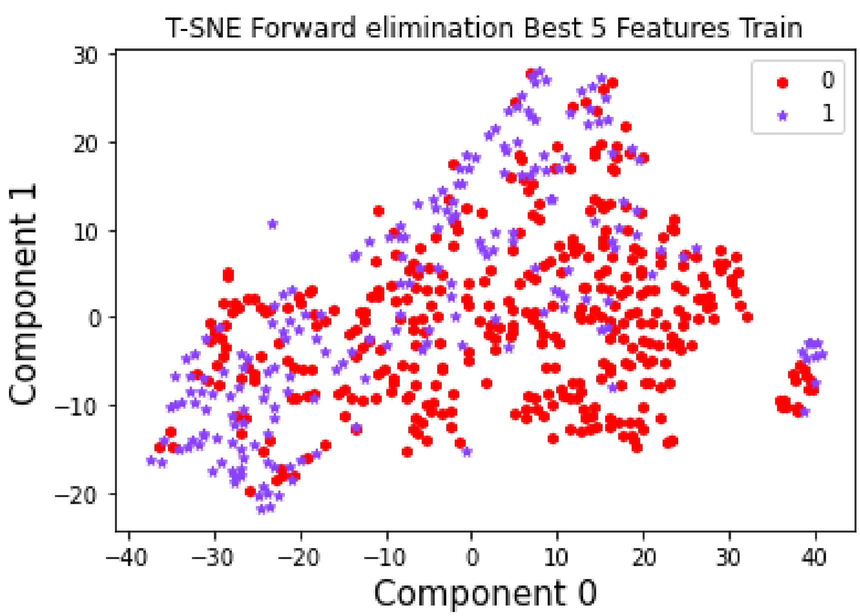


- **For both LR and KNN**

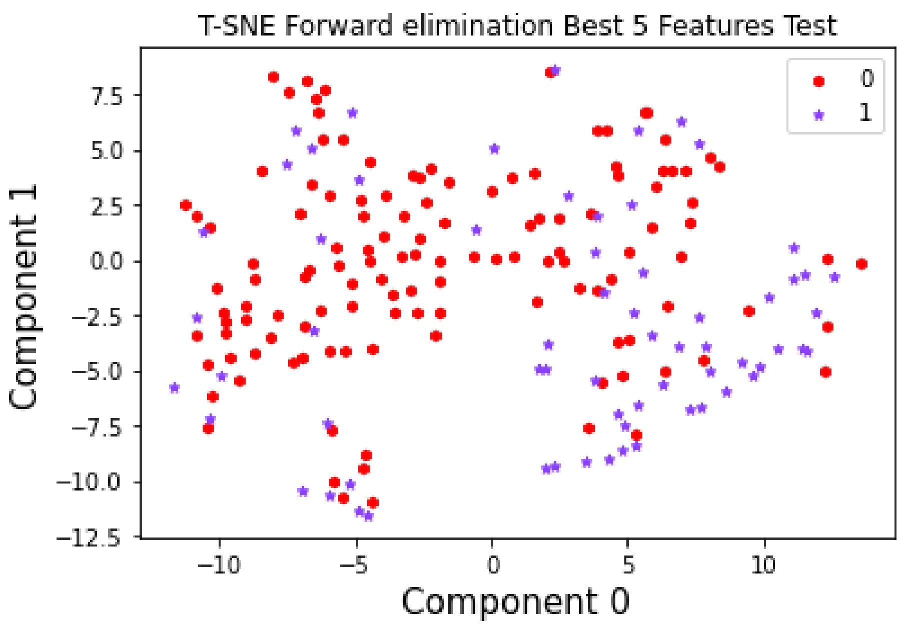


4(c)

- **Provide 2D TSNE for training set for filter method " forward elimination"**

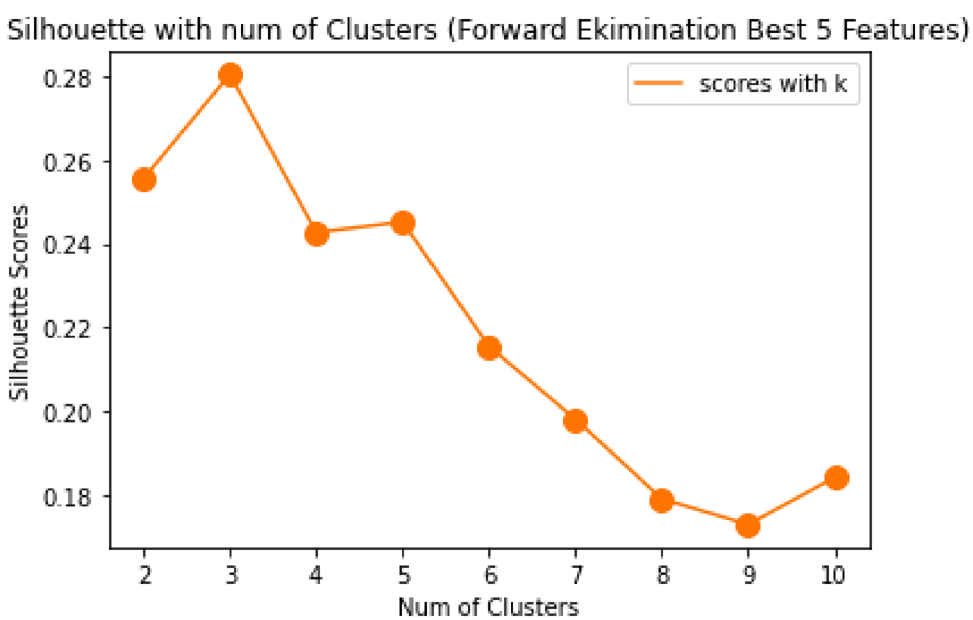


- **Provide 2D TSNE for testing set for filter method forward elimination**



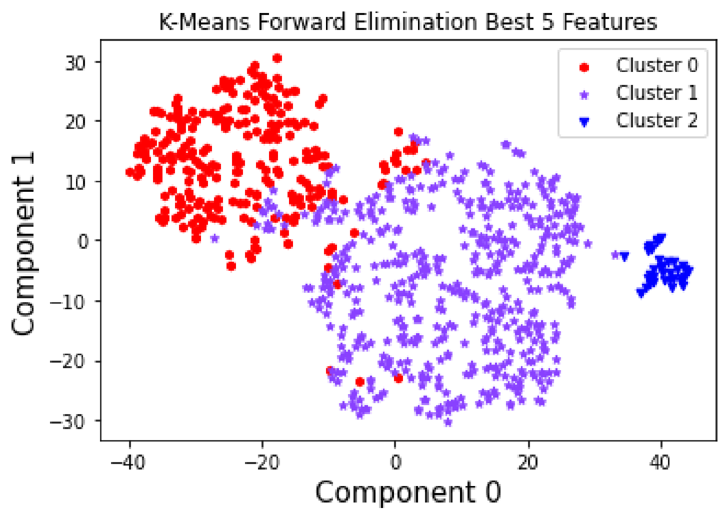
5(a)

- **Plotting the silhouette scores with num of cluster for k-means with forward elimination (5 features)**



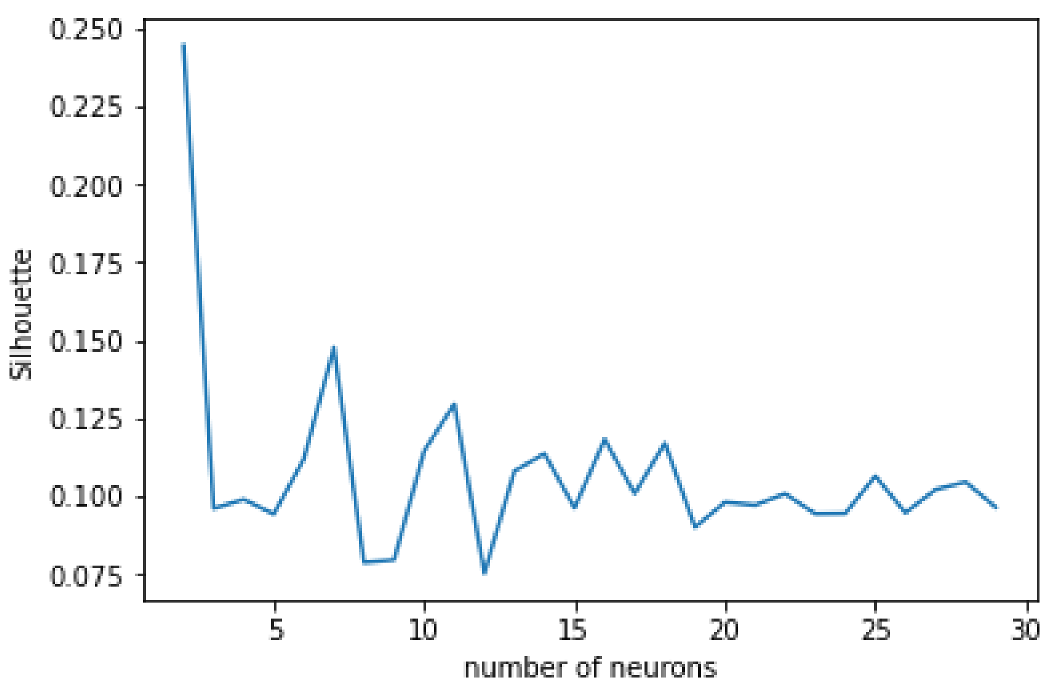
5(b)

- **The optimal number of clusters is 3 with best 5 features using forward elimination**



6(a)

We have plotted the silhouette_score vs the Num of clusters.



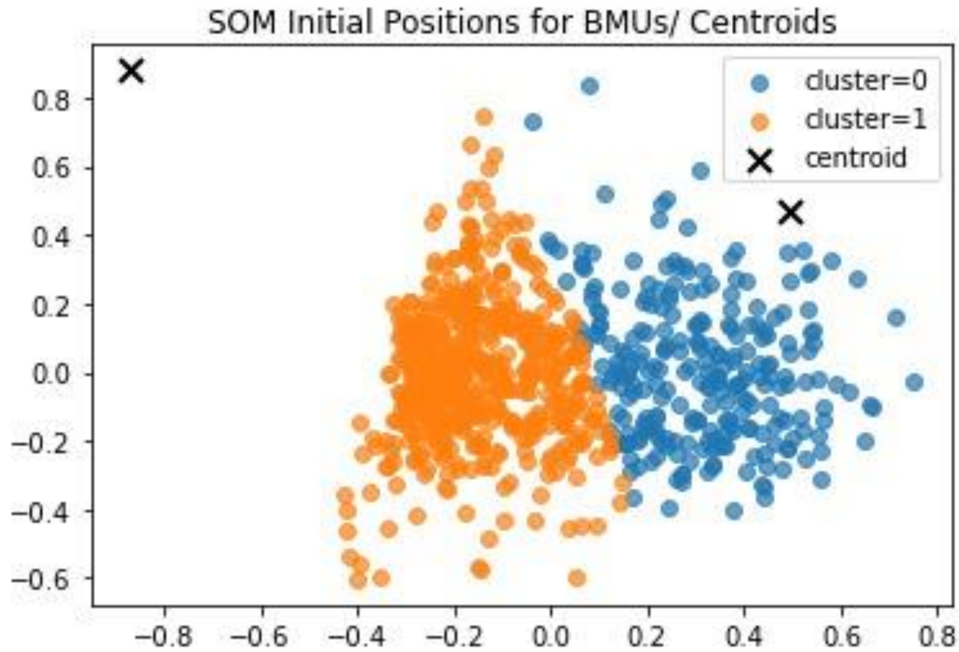
6(b)

We found that the number of 2 clusters is the optimal number of clusters with SOM

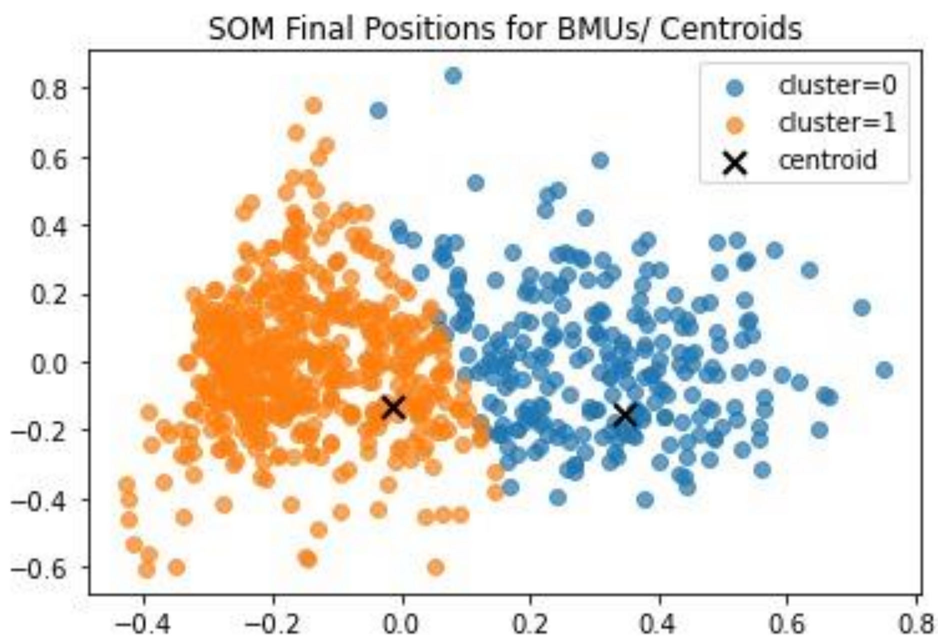
Based on silhouette_score which is = 0.24439547434879225

6(c)

We have plotted the Initial position of neurons.



After that we plotted the final position of neurons.



7(a)

We have tried more than 4500 combinations of epsilon and min_samples Values.

First, we have created these lists and after that we have tried all possible combinations.

epslist = np.array([0.3, 0.4, 0.5, 0.6,0.7])

minPoint = np.array([2,3,4,5,6,7,8,9,10,11,12,13,14,15])

5*14 = 70 different combinations

```python
#----------------------7(a)----------------------
#find DBSCAN optimal eps and minpoints
epslist = np.array([0.3, 0.4, 0.5, 0.6,0.7])
minPoint = np.array([2,3,4,5,6,7,8,9,10,11,12,13,14,15])

comb_array = np.array(np.meshgrid(epslist, minPoint)).T.reshape(-1, 2)

silhouetteDB = []
numberofcluster = []
epsls =[]
misls = []

for i in range(0,69):
    for j in range(0,69):
        model = DBSCAN(eps=comb_array[i][0], min_samples=comb_array[j][1])
        predLabels = model.fit_predict(XForward.to_numpy())
        if len(np.unique(predLabels)) == 1:
            continue
        else:
            epsls.append(comb_array[i][0])
            misls.append(comb_array[j][1])
            numberofcluster.append(len(np.unique(predLabels)))
            silhouetteDB.append(silhouette_score(XForward.to_numpy(), predLabels, metric='euclidean'))

EpsandMin = pd.concat([pd.DataFrame(silhouetteDB), pd.DataFrame(epsls), pd.DataFrame(misls), pd.DataFrame(numberofclust
EpsandMin.columns = ['silhouette_score', 'Eps', 'Min', 'NumofClusters']
```

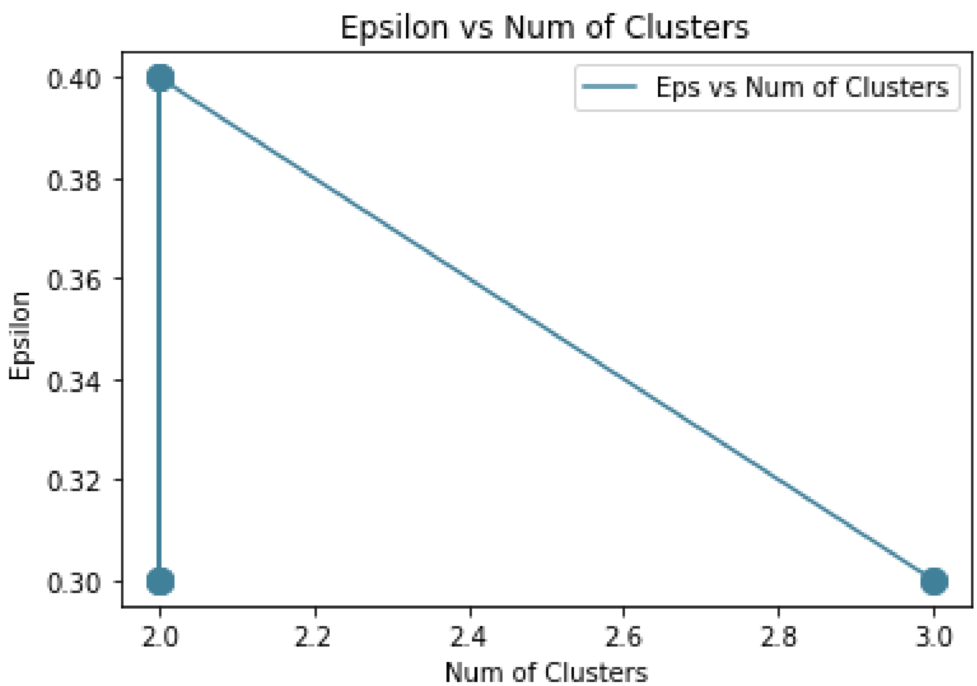
After that we have got something like this.

EpsandMin - DataFrame

Index	silhouette_score	Eps	Min	NumofClusters
978	0.4994	0.4000	14.0000	2.0000
979	0.4994	0.4000	15.0000	2.0000
992	0.4994	0.4000	14.0000	2.0000
993	0.4994	0.4000	15.0000	2.0000
1006	0.4994	0.4000	14.0000	2.0000
1007	0.4994	0.4000	15.0000	2.0000
1020	0.4994	0.4000	14.0000	2.0000
1021	0.4994	0.4000	15.0000	2.0000
1034	0.4994	0.4000	14.0000	2.0000
1047	0.4994	0.4000	14.0000	2.0000
1048	0.4994	0.4000	15.0000	2.0000
1061	0.4994	0.4000	14.0000	2.0000
1062	0.4994	0.4000	15.0000	2.0000

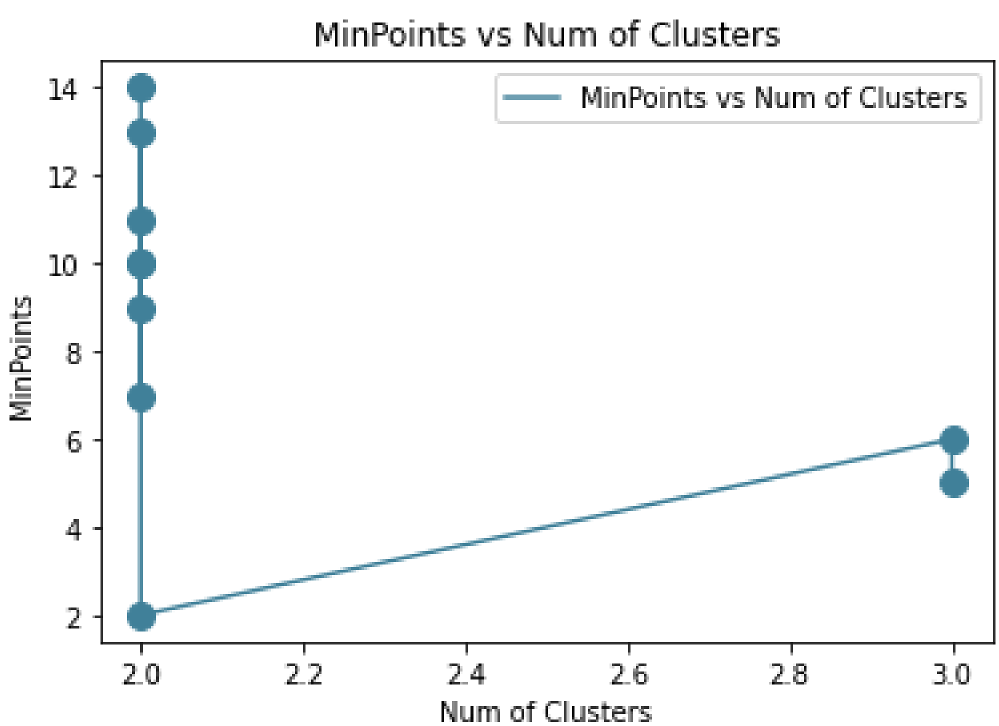
We have taken the parameters and the number of clusters with the highest silhouette.

And we have plotted the Epsilon vs the number of clusters.



7(b)

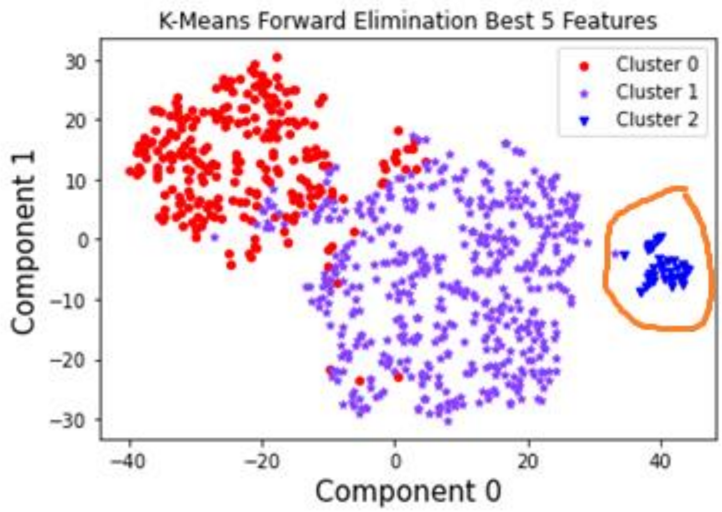
MinPoint vs the number of clusters.



8(a)

Before applying feature selection technique which is forward elimination with 5 features, we have found that the best number of k was 2.

And after applying forward elimination with 5 features, we have found that the best number of k was 3 and we think that make sense because as shown in the figure below, we think that it's makes more sense to clusters these data points into 3 clusters instead of 2.



8(b)

We have applied T-SNE technique for 3 different dataframes, the first one was the normal dataframe, and the second one was after applying the PCA technique on the dataframe, and the third on after applying the Forward elimination on the dataframe.