Data Mining (CSE 5334)
Assignment 4 (College dataset Report)

Student details:

- Kuldip Rameshbhai Savaliya – 1001832000
- Shivani Manojkumar Panchiwala – 1001982478
- Meghaben Ghanshyambhai Patel – 1002006777

Clustering:

An abstract object is clustered into classes based on their similarities. There are several types of data objects that can be considered one group. During cluster analysis, the data is first divided into groups based on similarity, then the labels are assigned to the groups. Overclassification by clustering has multiple advantages, including the ability to adapt to change and to distinguish between different groups with useful features.

Task 1

Import necessary packages and library.

```
[2]  #Seaborn,numpy,pandas,sklearn,matplotlib only
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     from sklearn.preprocessing import LabelEncoder
     from sklearn.preprocessing import MinMaxScaler
     from sklearn.preprocessing import StandardScaler
     from sklearn.cluster import KMeans
     from sklearn.metrics import confusion_matrix
```

Read datsaet.

```
[3]  df = pd.read_csv("College.csv")
     df.head()
```

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal	S.F.Ratio	perc.alumni
0	Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7440	3300	450	2200	70	78	18.1	12
1	Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12280	6450	750	1500	29	30	12.2	16
2	Adrian College	Yes	1428	1097	336	22	50	1036	99	11250	3750	400	1165	53	66	12.9	30
3	Agnes Scott College	Yes	417	349	137	60	89	510	63	12960	5450	450	875	92	97	7.7	37
4	Alaska Pacific University	Yes	193	146	55	16	44	249	869	7560	4120	800	1500	76	72	11.9	2

Checking for null values.

```
df.isnull().sum()
```

```
Private         0
Apps            0
Accept          0
Enroll          0
Top10perc       0
Top25perc       0
F.Undergrad     0
P.Undergrad     0
Outstate        0
Room.Board      0
Books           0
Personal        0
PhD             0
Terminal        0
S.F.Ratio       0
perc.alumni     0
Expend          0
Grad.Rate       0
dtype: int64
```

After processing dataset

```
df.describe()
```

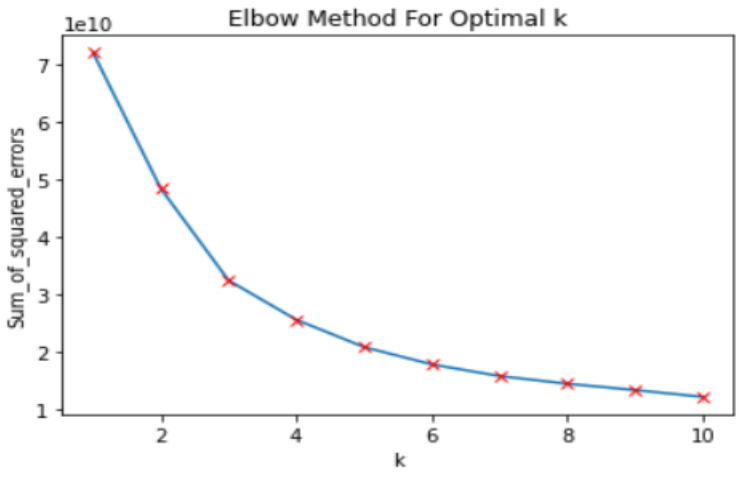
	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000
mean	0.727156	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	855.298584	10440.669241	4357.526384	549.380952	1340.642214	72.660232	79.702703
std	0.445708	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	1522.431887	4023.016484	1096.696416	165.105360	677.071454	16.328155	14.722359
min	0.000000	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.000000	2340.000000	1780.000000	96.000000	250.000000	8.000000	24.000000
25%	0.000000	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	95.000000	7320.000000	3597.000000	470.000000	850.000000	62.000000	71.000000
50%	1.000000	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.000000	9990.000000	4200.000000	500.000000	1200.000000	75.000000	82.000000
75%	1.000000	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	967.000000	12925.000000	5050.000000	600.000000	1700.000000	85.000000	92.000000
max	1.000000	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.000000	21700.000000	8124.000000	2340.000000	6800.000000	103.000000	100.000000

Elbow method for k =1 to 10.

The Elbow Method is one of the most popular methods to determine the optimal value of k.

```
plt.ylabel('Sum_of_squared_errors')
plt.title('Elbow Method For Optimal k')
plt.show()

##########################begin code for Task 1-a
```

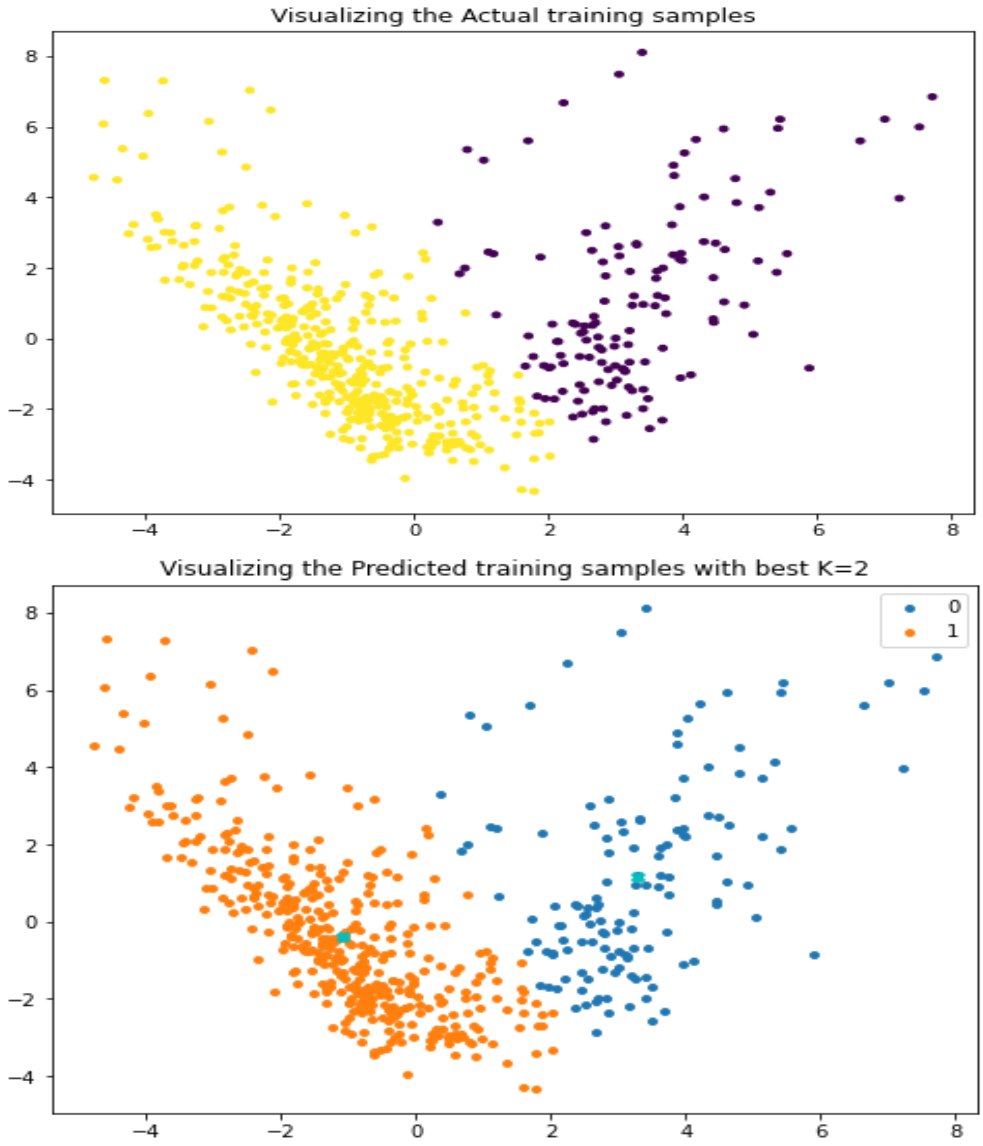


In the Elbow method we are varying the number of clusters K and for each value of K sum ofsquared errors between each point and centroid in a cluster. Here in the above plot, we can see that as the number of clusters increases the SSE starts decreasing.
Here we took the Optimal value of K as 3 because before 3 variation changes rapidly but after 3 it slows down leading to an elbow formation in the curve.

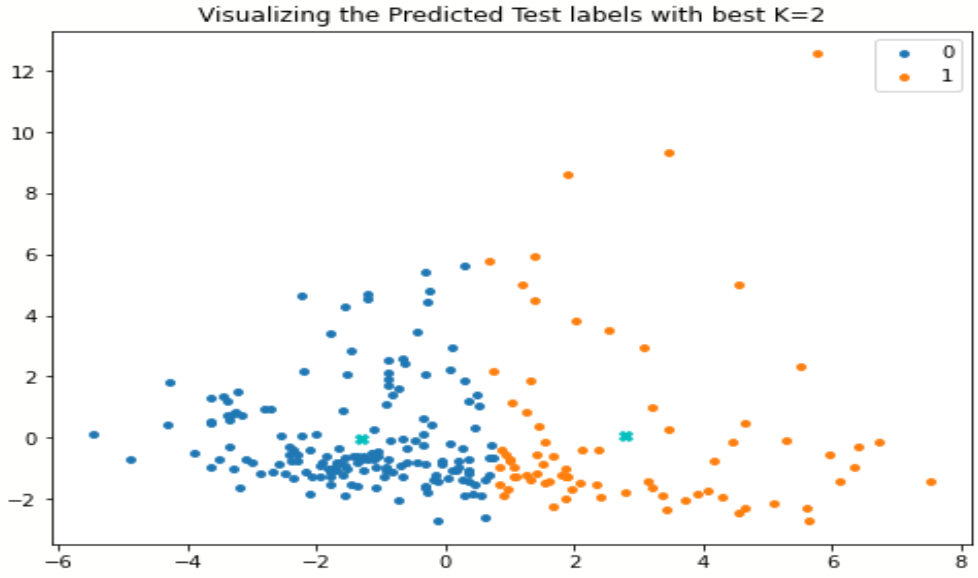
Visualization for K-Means Clustering

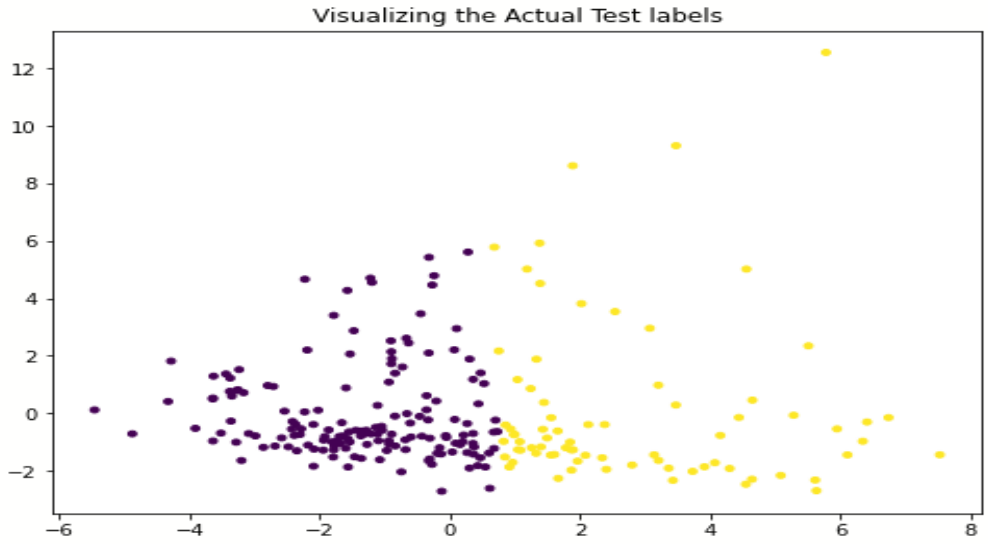
For k =2

Visualizing the Predicted training samples Vs Actual training samples



Visualizing the Predicted Test labels Vs Actual Test labels





The above graph is the scatter plot of predicted labels versus actual labels for testing data whe re I have run Kmeans on the whole dataset and assigned the actual labels dividing the data in three clusters for k=2. Moreover, I have then predicted the labels. Black Cross are the centroi ds of the clusters respectively. The model divides the data into three clusters for k=3 which is displayed in the first subplot of graph.

Confusion Matrix

```
confus_matx = confusion_matrix(c_labels,c_index)
print("Confusion Matrix for Testing data with k=2")
print(confus_matx)

####################end code for Task 1-b-4
```

```
Confusion Matrix for Testing data with k=2
[[159   0]
 [  0  75]]
```

Task 2

Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering is also known as the bottom-up approach. We begin by separating each object into its own group. In the process, close objects or groups are constantly merged. In this way, it continues until all groups are merged into one or until the termination condition is met.

F-1 score for complete and average linkage

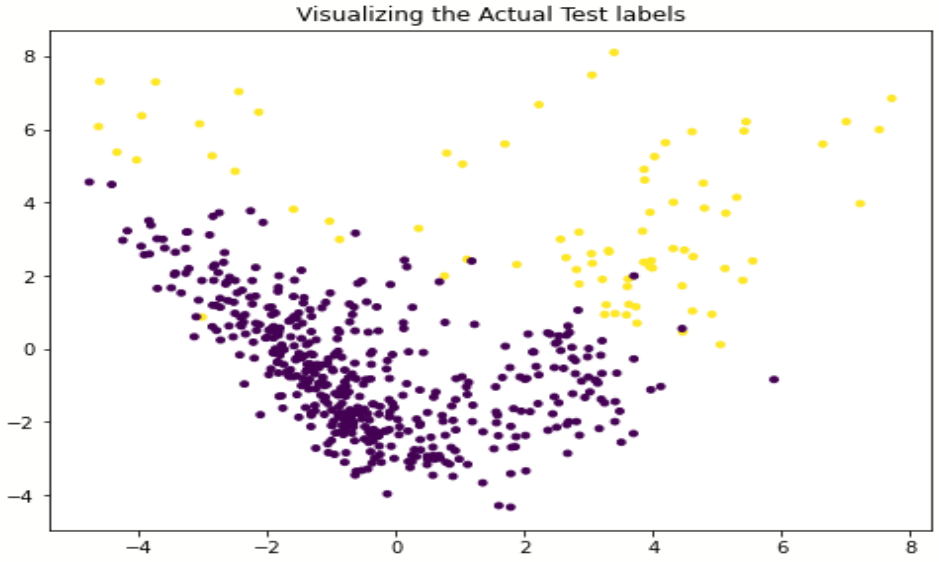
```
print("F1-score for complete linkage + cosine\t\t", f1_cos_complete)
print("F1-score for complete linkage + euclidean\t", f1_euclid_complete)
print("F1-score for complete linkage + manhattan\t", f1_manhat_complete)
print("F1-score for average linkage + cosine\t\t", f1_cos_avg)
print("F1-score for average linkage + euclidean\t", f1_euclid_avg)
print("F1-score for average linkage + manhattan\t", f1_manhat_avg)

####################end code for Task 2-a
```

```
F1-score for complete linkage + cosine          0.787042660495231
F1-score for complete linkage + euclidean       0.4768799600052604
F1-score for complete linkage + manhattan       0.42649856560471083
F1-score for average linkage + cosine           0.8678536429352478
F1-score for average linkage + euclidean        0.8799131990608571
F1-score for average linkage + manhattan        0.9575117931021825
```

Visualization for Hierarchical Agglomerative Clustering

Visualize using the best model Vs Actual Test labels



Compare K-Means Clustering and Hierarchical Agglomerative Clustering

The above graph is the scatter plot of predicted labels by kmean and Agglomerative clustering versus actual labels for traning data. The actual labels are divided in two clusters. Moreover, I have then predicted the labels. Black Cross are the centroids of the clusters respectively. The model divides the data into two clusters for kmeans no. of clusters 2 &Cosine Complete method which is displayed in the first two subplot of graph.

Confusion matrix for Kmeans and Agglomerative.

```
Confusion matrix:

K-means:

[[668   0]
 [  0 109]]

Agglomerative:

[[385 283]
 [100   9]]

Precision:

Kmeans: 1.0     Agglomerative: 0.5070785070785071
K-means Clustering method has higher precision value than Agglomerative Clustering


F1 Score

Kmeans: 1.0     Agglomerative: 0.5804357516188087

Recall

Kmeans: 1.0     Agglomerative: 0.5070785070785071
K-means Clustering method has higher recall value than Agglomerative Clustering
```