In [5]:

```
import csv
import numpy as np
import random
from numpy.random import seed
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

#I've converted the .data file to .csv which is read and replaced ? to NaN in the files
COLUMNS_COUNT = 2

with open('water-treatment.data', 'r') as f:
    columns = [next(f).strip() for line in range(COLUMNS_COUNT)]
temp_df = pd.read_csv('water-treatment.data', skiprows=COLUMNS_COUNT, header=None, delimit
er=';', skip_blank_lines=True)
even_df = temp_df.iloc[::2].reset_index(drop=True)
odd_df = temp_df.iloc[1::2].reset_index(drop=True)
df = pd.concat([even_df, odd_df], axis=1)
df.columns = columns
df.to_csv('out.csv', index=False)
text = open("out.csv", "r")
text = ''.join([i for i in text]) \
    .replace("?", "NaN")
x = open("out.csv","w")
x.writelines(text)
x.close()

reader=pd.read_csv('water-treatment.csv',header=None,delimiter=',');
df=pd.DataFrame(reader)
# print('Before Cleaning Up the DataSet\n')
# print(df)

#Calculating the mean of each Column and Replacing "NaN" with the Corresponding mean value
s
for i in range(1,39):
    mean = df.loc[:,i].mean()
#     print('The mean of column :'+str(i))
#     print(mean)
    df.loc[:,i].fillna(mean, inplace=True)

for i in range(1,39):
    for j in range(0,527):
        mean=df.loc[:,i].mean();
        stdevi=df.loc[:,i].std();
        df.loc[j,i]=(df.loc[j,i]-mean)/stdevi;

# print('\n')
# print('After Cleaning Up the DataSet and performing Normalization\n')
# print(df)
```

```
#Dropping the Date Column
# print('\n')
# print('After Dropping\n')
df.drop(df.columns[0], axis=1, inplace=True)
# print(df)

# Implementing K-Means with K as 4
kmeans = KMeans(n_clusters=4, init='k-means++', max_iter=300, n_init=10, random_state=0)
pred_y = kmeans.fit_predict(df)


# Adjusting the Clustering output from 0-3 to 1-4
for i in range(len(pred_y)):
    if pred_y[i]==0:
        pred_y[i]=1
    elif pred_y[i]==1:
        pred_y[i]=2
    elif pred_y[i]==2:
        pred_y[i]=3
    else:
        pred_y[i]=4

# Adjusting the Output to the desired form so that the Clusters get renamed and appear in
 order
l1=[]
l2=[]
cnt=0
for k in pred_y:
    if not k in l1:
        l1.append(k)
        cnt=cnt+1
        l2.append(cnt)
for k in range(len(pred_y)):
    for k1 in range(len(l1)):
        if (pred_y[k]==l1[k1]):
            pred_y[k]=l2[k1]
            break

print('Clustering Output of K-Means Without PCA')
print(pred_y)


# Implementing Principal Component Analysis with 2 components On the Normalized DataFrame
df1 = StandardScaler().fit_transform(df)
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(df1)
principalDf = pd.DataFrame(data = principalComponents
                           , columns = ['principal component 1', 'principal component 2'])
print('PRINCIPAL COMPONENT ANALYSIS\n')
# print('Principal Components\n')
# print(principalDf)

# K-Means Implementation on the PCA applied Dataset
kmeans1 = KMeans(n_clusters=4, init='k-means++', max_iter=300, n_init=10, random_state=0)
pred_y1 = kmeans1.fit_predict(principalDf)
```

```
# Adjusting the Clustering output from 0-3 to 1-4
for i in range(len(pred_y1)):
    if pred_y1[i]==0:
        pred_y1[i]=1
    elif pred_y1[i]==1:
        pred_y1[i]=2
    elif pred_y1[i]==2:
        pred_y1[i]=3
    else:
        pred_y1[i]=4


# Adjusting the Output to the desired form so that the Clusters get renamed and appear in
 order
l1=[]
l2=[]
cnt=0
for k in pred_y1:
    if not k in l1:
        l1.append(k)
        cnt=cnt+1
        l2.append(cnt)
for k in range(len(pred_y1)):
    for k1 in range(len(l1)):
        if (pred_y1[k]==l1[k1]):
            pred_y1[k]=l2[k1]
            break

print('Clustering Output of K-Means With PCA')
print(pred_y1)




x_train,x_test,y_train,y_test=train_test_split(df,pred_y,test_size=0.6,random_state = seed
(50))

print(x_train)

from sklearn.linear_model import LogisticRegression

# instantiate the model (using the default parameters)
logreg = LogisticRegression()

# fit the model with data
logreg.fit(x_train,y_train)

#
y_pred_1=logreg.predict(x_test)
print(y_test.shape,y_pred_1.shape)
acc1=accuracy_score(y_test, y_pred_1)
print('Accuracy of K-Means')
print(acc1)


x1_train,x1_test,y1_train,y1_test=train_test_split(principalDf,pred_y1,test_size=0.6,rando
m_state = seed(50))
```

```
print(x1_train)


# instantiate the model (using the default parameters)
logreg1 = LogisticRegression()

# fit the model with data
logreg1.fit(x1_train,y1_train)

#
y_pred_2=logreg1.predict(x1_test)
print(y1_test.shape,y_pred_2.shape)
acc2=accuracy_score(y1_test, y_pred_2)
print('Accuracy of K-Means with PCA')
print(acc2)
```

```
Clustering Output of K-Means Without PCA
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
 1 2 2 1 2 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 2
 2 2 2 2 2 1 1 1 1 2 2 1 1 1 1 1 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 3 3 3 3 3 4 4 4 4]
PRINCIPAL COMPONENT ANALYSIS

Clustering Output of K-Means With PCA
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
 2 2 2 1 2 1 2 2 1 1 2 2 2 2 2 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 2 1 2 2 2 2 2
 2 2 2 2 2 1 1 1 1 2 2 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 3 3 3 3 3 4 4 4 4]
            1         2         3         4         5          6         7  \
200   0.590763 -0.423138  0.714049  1.008833  0.292013   1.680037  0.618711
352   1.400013 -0.190434  1.466344  1.855094  1.606130   0.336461  1.770435
237   0.780591 -0.383726  0.920541  1.474868  0.581768   0.254449  1.262709
398   2.111671 -0.028521  1.720476  1.752335  1.870799  10.031188  0.255011
377   1.652888  0.379027  1.649311  1.131598  0.839033   0.622823  1.362012
420   2.186539  2.678968  1.905348  1.875776  1.928250   2.421378  1.381358
344   1.060596  2.145240  1.384347  1.794678  1.341711   0.956791  1.588089
205   1.248719  0.381630  0.757701 -0.636734 -0.415493   0.691982 -0.411081
149   0.234865 -0.270699  0.472307  0.046730  0.512421   0.154988  0.445428
21    1.232463 -0.383702  0.127330 -0.352451  0.108834  -0.132550  0.881118
54    0.233283 -0.046460  0.331127  0.232899  1.477659   0.128200  0.735460
245   0.846820 -0.421894  0.977470  0.994263  0.962478  -0.007417  1.193418
485   3.323863 -0.108912  3.242973  3.928109  3.871371   6.105776  2.665868
252   1.253853  0.066732  1.058455  1.418055  2.122703   0.966213  0.617208
490   3.743444 -0.299344  3.375641  4.000520  4.537703   4.907776  3.316742
169   0.511896 -0.052735  0.591991  0.655740  0.994602   2.207036 -0.216315
362   2.097964 -0.183000  1.627275  1.230548  0.901667   0.301851  2.031621
187  -0.437104  0.777727  0.674308  0.278430  0.633920   2.678738 -0.283110
28   -0.540658  0.284394  0.112487 -0.255365 -0.068751  -0.332838  0.596743
239   0.780147 -0.077626  1.010005  0.686432  0.546092   0.194284  1.186346
241   0.928437  0.047806  1.044977  0.780462  0.564595   0.400927  1.534317
```

```
176   1.114679  0.030657  0.612794  0.167937 -0.168613   1.339998  0.637930
495   4.673696  1.360567  3.586214  5.013939  4.181641   4.449305  3.984402
331   1.213541  0.680707  1.482656  1.669625  1.483849   0.765364  1.586044
324   1.091904  0.388089  1.239995  3.027606  0.951835   0.559556  1.190813
407   1.448280  0.645470  1.770556  3.463661  2.125569   2.402735  2.397075
431   2.057505  1.350249  1.994110  1.815614  3.072080   1.966455  2.460311
336   1.187680  2.272963  1.425591  1.255809  1.185370   0.490929  1.798154
365   1.440864  0.045862  1.649448  1.898872  1.738716   0.630358  1.810256
450   2.825899  0.576642  2.296573  2.769058  1.522377   1.584871  2.568984
..         ...       ...       ...       ...       ...        ...       ...
320   1.830642  0.620724  1.278242  0.164247  0.050886   0.509717  0.334891
425   1.614096  0.529895  1.960845  0.563962  0.610387   1.641015  1.089480
106   0.572391 -0.140477  0.593472  1.002941 -0.104972  -0.000912  0.207660
341   1.043546  2.898011  1.397281  1.491188  1.550029   0.881657  1.703613
49    0.610073 -0.645352  0.223776 -0.630452 -0.136382  -0.479881 -0.044012
287   1.975932  0.117697  1.260384  1.107921  0.263412   0.438070  0.969707
299   1.824621 -0.269591  1.205253  1.097363  0.567340   0.648442  0.995148
404   1.878787 -0.013465  1.844090  3.582471  2.765613   1.813977  1.963494
134   0.818741 -0.362746  0.670045  0.288915  0.592372   0.099438  0.593870
218   0.325261 -0.405971  0.771200  0.458046  0.155576   0.079081  0.834152
154   0.413256 -0.339343  0.605735  0.455254  0.640438   0.093605  0.591032
91    0.254457  1.483865  0.380106  0.873321  0.588648   0.308546  0.476446
348   1.217714 -0.059457  1.499129  1.493912  1.038530   0.381324  1.782549
163   0.578048  0.317702  0.690130  0.537257  0.505359   0.719045  0.018720
462   2.660281  0.671826  2.513365  4.713649  3.773788   4.260175  2.866774
452   2.881550 -0.036298  2.287790  2.308542  2.501319   1.895502  2.703236
367   1.579226  0.047587  1.664955  3.332874  1.154654   0.487034  1.826511
71    0.019363  0.540536  0.273064  0.940964  1.309603   1.946153 -0.308414
250   1.338435  2.663701  0.995970  0.258902  1.108490   0.709026  0.442608
95    0.587908  0.400099  0.461686  0.678011  0.104833  -0.093637  0.761111
258   1.465955  1.261248  1.082060  0.431185  0.239373   0.185104  0.877329
133   0.415836  0.442169  0.575119  0.740715  2.049244   1.156561 -0.150348
278   0.991900  1.907814  1.055815  0.903896  0.989762   0.568774  1.217188
522  10.314482 -0.260073  5.314858  5.827934  6.801317   9.494955  8.658073
229   0.446986  0.096386  0.944892  1.439223  1.366804   0.399281  1.168394
70    0.558059 -0.011970  0.308016  0.162594  0.375761  -0.081756  0.338200
132   0.543798 -0.109031  0.602935 -0.493877  0.112885  -0.303026  1.077143
289   1.663318  0.526455  1.242216  1.083272  0.484876   0.441789  0.830830
109   0.756729 -0.248332  0.536264  0.464978  0.196191   0.084363  0.518309
480   3.367822  0.626779  2.986142  3.111462  3.338931   2.690030  3.233659

             8         9        10  ...        29        30        31  \
200   0.181934  1.487496  0.678557  ...  1.590789  0.767834  1.563994
352   0.207541  0.973508  1.512081  ...  1.005961  1.846716  0.519833
237   0.255720  0.811225  0.938397  ...  1.107790  0.370612  0.688950
398  12.330188  2.561965  1.706572  ...  2.606051  1.866646  2.958773
377   0.100806  1.806033  1.635237  ...  1.838024  0.764556  1.158444
420   2.157219  1.570842  2.001030  ...  1.335825  2.978069  1.910323
344   1.026321  1.809527  1.400734  ...  1.645292  1.737098  1.671343
205  -0.488447  0.641911  0.804779  ...  0.639627  0.765439  1.120902
149   0.219731  0.132817  0.464752  ...  0.227886  1.000355  0.962413
21   -0.146614  0.927555  0.049446  ...  0.583474 -0.609619 -0.367444
54    1.292595  0.386866  0.364722  ... -0.160873  0.182634  0.454624
245   0.757502  1.204052  0.995180  ...  1.220365  1.586539  0.518175
485   3.655517  3.636097  3.184426  ...  3.703161  4.610370  3.970541
252   0.613656  1.018458  1.048151  ...  0.807398  1.883955  0.838422
490   2.449473  5.004741  3.366454  ...  4.744580  3.427344  4.473590
```

```
169  1.022678  2.039420  0.555836  ...  2.359183  0.594820  1.573879
362  0.741028  1.916689  1.612405  ...  2.095637  0.946988  0.244036
187  1.291828  0.778931  0.666526  ...  0.397059  2.392133  1.621229
28  -0.439973 -0.691592  0.100824  ... -0.635665  0.333488 -0.054203
239  0.515995  0.957041  1.000200  ...  1.101657  1.320133  0.697384
241  0.586783  0.693237  1.034940  ...  1.020021  1.526993  1.018219
176  0.288031 -0.235310  0.605211  ... -0.118090  0.854469  1.428808
495  2.524738  5.174742  3.632303  ...  4.503750  2.718852  4.443561
331  0.833467  1.448902  1.439559  ...  1.575205  0.977008  1.347622
324  0.816655  0.542440  1.198971  ...  0.548114  2.568639  1.217048
407  1.284342  1.929292  1.791312  ...  1.875081  2.329956  1.524692
431  1.826308  2.022871  1.981464  ...  2.048492  2.825447  2.539493
336  0.661654  1.443596  1.440749  ...  1.667377 -0.110821  1.354467
365  1.261169  1.403279  1.665636  ...  1.350893  1.494736  1.198747
450  0.473737  3.013844  2.284428  ...  3.216090  1.895223  2.177687
..        ...       ...       ...  ...       ...       ...       ...
320 -0.302975  0.712540  1.293732  ...  0.681241  0.483373  1.250665
425  0.036232  0.987651  1.948313  ...  1.224310  1.788055  2.534653
106 -0.394132  0.626612  0.551807  ...  0.217094  0.536380  0.905274
341  1.016972  0.945843  1.383686  ...  1.084422  1.217556  1.445004
49   0.107765 -0.595778  0.169801  ... -0.779102  0.168389  0.447469
287  0.373148  1.301896  1.219744  ...  1.289073  1.488912  0.607888
299  0.556203  1.268169  1.191857  ...  1.360194  1.035636  1.084717
404  1.801763  1.663953  1.863442  ...  1.557173  2.421345  1.956644
134 -0.145831  0.455012  0.630621  ...  0.311890  0.421893  0.563405
218 -0.431486  1.349944  0.763359  ...  1.360665  1.223581  0.733252
154  0.231630  1.484557  0.597853  ...  1.382091  0.142308  0.760396
91   0.366552  0.275476  0.406696  ...  0.373946  0.642233  0.939748
348  0.199535  1.281730  1.485047  ...  1.349124  1.714919  0.360795
163  0.088565  0.867785  0.681819  ...  0.957363  0.253679  0.977439
462  3.563962  3.705145  2.542994  ...  3.700323  3.348492  3.266220
452  1.063251  2.676667  2.275461  ...  3.156297  1.540790  2.299021
367  1.099948  1.256257  1.681448  ...  1.349820  1.051843  0.865591
71   1.082371 -0.404230  0.264554  ... -0.187894  1.470585  1.331470
250  0.639416  1.054918  0.959037  ...  0.986261  0.115571  1.032293
95  -0.088985  0.366663  0.453487  ...  0.015559  0.541544  0.564796
258 -0.338782  1.171430  1.099213  ...  1.201271  1.185044  1.028771
133  0.522235  0.202639  0.566673  ...  0.244670  0.598751  1.132885
278  0.679024  2.004100  1.044503  ...  2.022767  1.631695  1.539661
522  1.087743  8.575488  5.297481  ...  8.522675  9.380209  9.639611
229  0.620171  1.231441  0.936004  ...  1.380238  1.007389  0.692204
70   0.163280  0.048938  0.299305  ... -0.034757 -1.555194 -0.646540
132  0.016780 -0.123348  0.594339  ... -0.121295 -0.215735  0.162503
289  0.376286  1.778051  1.228904  ...  1.620510  0.649807  0.138905
109  1.149756  0.377540  0.527652  ...  0.405869  0.521483  0.926764
480  2.273880  2.940092  3.027037  ...  3.120192  3.031194  2.889353

           32        33         34        35        36         37         38
200  0.943700  0.766121   0.574541  0.790060  0.655664   0.842037   0.764185
352  1.146698  1.551989   1.377219  1.528453  1.453054   1.404887   1.412855
237  0.831691  1.044688   1.078634  0.992600  0.932606   0.954433   0.908912
398  1.897716  1.743673   1.745460  1.788695  1.887258   1.988935   1.768685
377  1.516506  1.631146   1.770686  1.547958  1.401482   1.614113   1.598709
420  1.856821  2.131973   2.232611  2.053202  2.180008   2.136667   1.987821
344  1.356576  1.440030   1.364396  1.346502  1.249260   1.345729   1.382488
205  0.779560  0.861020   0.128184  0.798700  0.343038   0.903608   0.809102
149  0.794707  0.755596   0.585543  0.716581  0.747905   0.728199   0.639787
```

```
21   0.229237  0.274591   0.651733  0.114083  0.430882   0.140853   0.227209
54   0.505434  0.035875  -0.080848  0.317087  0.499214   0.157875   0.353163
245  0.694731  1.090938   1.301257  1.027594  1.134487   0.806419   0.939158
485  3.534748  3.460580   3.385617  3.552638  3.538027   3.650700   3.419993
252  1.055269  0.840510   0.905170  0.945260  0.922442   0.998360   0.964361
490  3.775417  3.712411   3.924678  3.473465  3.873502   3.704503   3.607916
169  0.845772  0.543942   0.375067  0.656166  0.643136   0.756958   0.673174
362  0.861951  1.590098   1.199851  1.468385  1.005470   1.163651   1.493277
187  0.889222  0.583136   0.588685  0.808616  0.931848   0.923241   0.752778
28   0.454903  0.287432   0.495251  0.171496  0.254582  -0.066158   0.266552
239  0.989099  1.016702   0.990239  0.932588  0.811153   0.921504   0.922001
241  1.043189  0.893087   1.259248  0.887972  1.090500   0.903991   0.929067
176  0.796150  0.725515   0.822252  0.745994  0.811446   0.856597   0.709218
495  4.019445  3.251793   3.235398  3.513729  3.539384   4.051784   3.918250
331  1.276451  1.183196   1.161566  1.231822  1.187354   1.269487   1.291304
324  1.243088  1.324240   0.624641  1.425922  1.298091   1.271435   1.256335
407  1.819419  2.026262   1.763352  2.013125  1.774311   1.992709   1.854495
431  2.046373  2.200504   0.218225  2.252598  1.842033   2.280673   2.110198
336  1.350635  1.336834   1.248206  1.271648  1.224881   1.224641   1.337371
365  1.449912  1.689114   1.646980  1.591965  1.552781   1.545665   1.513371
450  2.416670  2.526318   1.885612  2.517839  1.959549   2.277256   2.435534
..        ...       ...        ...       ...       ...        ...        ...
320  1.260787  1.136114   1.182739  1.099090  1.194438   1.271367   1.255130
425  1.950768  2.135210   2.053645  1.841158  1.997733   2.124412   2.020009
106  0.599051  0.511326   0.510047  0.490408  0.312518   0.221730   0.497650
341  1.458030  1.475575   2.030090  1.397113  1.780256   1.426047   1.365310
49   0.409408  0.302686   0.040878  0.212304  0.036889   0.227205   0.322174
287  1.063757  1.197500   1.227319  1.205673  1.185471   1.075131   1.098779
299  1.216026  1.142754   1.095721  1.138816  1.054378   1.063683   1.152191
404  1.807420  1.903817   2.044470  1.919192  2.058095   1.813902   1.824945
134  0.532858  0.429030   0.780926  0.445888  0.640440   0.526533   0.583094
218  0.753714  0.654516   0.883500  0.727917  0.653902   0.724581   0.816180
154  0.660576  0.754574   0.678307  0.696408  0.684882   0.650730   0.654531
91   0.630882  0.423659  -0.105674  0.458648  0.305743   0.533418   0.447827
348  1.310243  1.540810   1.322098  1.467031  1.359158   1.324210   1.397435
163  0.804355  0.657895   0.823146  0.675911  0.738257   0.737035   0.681246
462  2.811911  2.823123   3.182993  2.766614  2.996332   2.793539   2.686641
452  2.583797  2.563043   2.349274  2.469807  2.141866   2.276101   2.464886
367  1.343723  1.663319   1.825777  1.641376  1.593115   1.581668   1.522604
71   0.492026  0.103430   0.374481  0.453404  0.696494   0.547587   0.393377
250  0.945355  0.645693   0.791143  0.692423  0.983605   0.970732   0.942695
95   0.585786  0.453764   0.326619  0.448587  0.305231   0.419671   0.486320
258  1.006760  1.106081   1.146734  1.064195  1.013915   1.040991   0.976375
133  0.682660  0.599111   0.597377  0.588260  0.815415   0.698122   0.587120
278  1.189123  1.315657   0.981588  1.190170  1.027467   1.076988   1.060065
522  9.898832  8.884289  10.856066  9.260702  9.878169  10.775296  10.119783
229  0.874966  0.905599   0.825848  0.869482  0.852292   0.819917   0.887600
70   0.319484  0.483727   0.125769  0.324202 -0.055439   0.365472   0.410847
132  0.541707  0.512204   0.533059  0.419238  0.530505   0.596321   0.593483
289  0.921071  1.158531   1.012392  1.165125  1.031951   1.059010   1.101096
109  0.733554  0.352850   0.204456  0.390326  0.277135   0.294337   0.526487
480  3.068058  2.483615   3.722654  3.327820  3.229108   3.265797   3.217998

[210 rows x 38 columns]
(317,) (317,)
Accuracy of K-Means
0.9085173501577287
```

```
     principal component 1  principal component 2
200              -1.987848              -0.717049
352              -0.538563              -0.867165
237              -1.888722              -0.716493
398               3.934969              12.782758
377              -0.123516              -0.841084
420               1.499599               0.923613
344               0.554047               1.341072
205              -3.479328              -1.078665
149              -3.533482              -0.892024
21               -4.038778              -1.453991
54               -3.266765               0.438525
245              -1.647838              -0.203628
485               6.766421               2.095481
252              -1.388085              -0.285998
490               8.576293               2.096788
169              -1.839171               0.490285
362               0.007156              -0.159884
187              -2.398015               1.522220
28               -4.510180              -1.114859
239              -1.795579              -0.794625
241              -1.639878              -0.438064
176              -3.234522              -0.694267
495               8.979959               1.182692
331              -0.180363               0.059778
324              -1.079475              -0.067051
407               1.600190               0.010572
431               1.937542               0.892235
336              -0.397423               0.162493
365               0.192289               0.056228
450               2.806411              -1.063748
..                     ...                    ...
320              -1.936370              -0.946310
425               0.445597              -0.399898
106              -3.064910              -1.009880
341              -0.332341               0.492464
49               -4.913427              -1.049846
287              -1.517393              -0.651309
299              -1.244258              -0.299383
404               1.819039               0.593564
134              -3.041046              -0.867104
218              -2.272288              -0.105506
154              -2.808659              -0.919068
91               -3.135435               0.605232
348              -0.487324              -0.968356
163              -2.924589              -0.611728
462               5.686925               2.812578
452               3.236613              -0.379374
367               0.210154              -0.210586
71               -3.578414               0.688623
250              -2.097674               0.105015
95               -3.539813              -0.946661
258              -2.016863              -1.094202
133              -2.806577               0.385260
278              -0.716675               0.343939
522              21.325591              -2.950235
229              -1.346707              -0.186457
```

```
70               -3.984971            -0.825699
132              -3.690078            -1.040832
289              -1.503117            -0.747202
109              -3.003904             0.514880
480               5.365421             0.646668

[210 rows x 2 columns]
(317,) (317,)
Accuracy of K-Means with PCA
0.9463722397476341
```