# Importing respective libraries and checking Data

import numpy as np  
import pandas as pd   
import seaborn as sns   
import matplotlib.pyplot as plt  
%matplotlib inline  
from sklearn.decomposition import PCA  
from sklearn.preprocessing import StandardScaler  
from sklearn.preprocessing import Normalizer  
from sklearn.pipeline import make\_pipeline  
from sklearn.pipeline import Pipeline

data = pd.read\_table('dailysportsdata.txt', delimiter=',', header=0)

data.head()

Activity

T-x\_acc (1)

T-y\_acc (1)

T-z\_acc (1)

T-x\_gryo (1)

T-y\_gryo (1)

T-z\_gryo (1)

T-x\_mag (1)

T-y\_mag (1)

T-z\_mag (1)

...

RL-z\_mag (125)

LL-x\_acc (125)

LL-y\_acc (125)

LL-z\_acc (125)

LL-x\_gryo (125)

LL-y\_gryo (125)

LL-z\_gryo (125)

LL-x\_mag (125)

LL-y\_mag (125)

LL-z\_mag (125)

0

A1

8.1305

1.0349

5.4217

-0.009461

0.001915

-0.003424

-0.78712

-0.069654

0.15730

...

-0.036874

-2.8154

-9.0600

2.6025

-0.003904

-0.006729

-0.009789

0.73897

0.30275

-0.056262

1

A1

7.9665

1.1684

5.6755

-0.005730

0.026995

-0.009029

-0.79062

-0.071635

0.13429

...

-0.038551

-2.8233

-9.0757

2.6337

-0.006769

-0.006575

-0.004326

0.74027

0.30192

-0.057155

2

A1

7.8917

1.1390

5.6980

0.014180

0.028722

-0.009079

-0.79531

-0.069460

0.12447

...

-0.040145

-2.8091

-9.0846

2.6295

-0.000714

-0.002681

0.004770

0.74072

0.30101

-0.057301

3

A1

7.9366

1.1536

5.6318

0.003242

0.029965

0.009111

-0.79292

-0.070358

0.13194

...

-0.041109

-2.8844

-9.0849

2.6298

-0.010604

-0.002827

-0.004194

0.74150

0.30305

-0.055743

4

A1

7.8913

1.1972

5.9082

-0.044333

-0.067467

-0.004235

-0.79592

-0.073174

0.12086

...

-0.039495

-2.8249

-9.1083

2.6322

0.013583

0.013670

0.007613

0.74007

0.30324

-0.055548

5 rows × 5626 columns

CLASS = data.Activity   
x = data  
list = ['Activity']  
x = data.drop(list,axis = 1 )  
data2=x

data2.shape

(9120, 5625)

# Standardization

scaler = StandardScaler()

scaler.fit\_transform(data2)

array([[ 0.04391049, 0.69440013, 0.70868137, ..., 0.76503261,  
 0.01769077, -0.34162785],  
 [ 0.01516927, 0.7447215 , 0.77789485, ..., 0.76840313,  
 0.01551985, -0.34416899],  
 [ 0.00206047, 0.73363949, 0.7840308 , ..., 0.76956985,  
 0.01313969, -0.34458445],  
 ...,  
 [ 0.3902948 , 0.46881719, -0.76377687, ..., 0.81706826,  
 0.60548656, 0.02427635],  
 [ 1.08569219, -0.45123074, -0.68856106, ..., 0.75577664,  
 -0.04905843, 1.09561572],  
 [ 0.07577119, 0.66723414, 0.17984454, ..., 0.06839784,  
 0.36668553, 1.36964904]])

data2\_scaled = scaler.fit\_transform(data2)

col = data2\_scaled[:,0]

np.mean(col)

3.739698609263685e-17

np.var(col)

0.9999999999999998

col = data2\_scaled[:,1]

np.mean(col)

-8.570142646229279e-18

np.var(col)

1.0000000000000002

# Feature reduction and PCA

pca=PCA(n\_components=500)  
pca.fit(data2\_scaled)

PCA(n\_components=500)

data\_pca = pca.fit\_transform(data2\_scaled)

data\_pca.shape

(9120, 500)

plt.bar(range(pca.n\_components\_), pca.explained\_variance\_)  
plt.xlabel('Principle Component')  
plt.ylabel('Variance')  
plt.xlim([0,400])  
plt.ylim([0,100])  
plt.show()

png

png

plt.bar(range(pca.n\_components\_), pca.explained\_variance\_)  
plt.xlabel('Principle Component')  
plt.ylabel('Variance')  
plt.xlim([0,50])  
plt.ylim([0,100])  
plt.show()

png

png

model = PCA()  
model.fit(data2\_scaled)

PCA()

plt.plot(np.cumsum(model.explained\_variance\_ratio\_))  
plt.xlim(0,400,1)  
plt.xlabel('Number of Principal Components')  
plt.ylabel('Cumulative Explained Variance')

Text(0, 0.5, 'Cumulative Explained Variance')

png

png

np.cumsum(model.explained\_variance\_ratio\_)[400]

0.8614645192936279

plt.plot(np.cumsum(model.explained\_variance\_ratio\_))  
plt.xlim(0,45,1)  
plt.xlabel('Number of Principal Components')  
plt.ylabel('Cumulative Explained Variance')

Text(0, 0.5, 'Cumulative Explained Variance')

png

png

np.cumsum(model.explained\_variance\_ratio\_)[45]

0.6201554857440478

FC=45  
pipeline = Pipeline([('pca', PCA(n\_components=FC))])  
transformed=pipeline.fit\_transform(data2\_scaled)

tran=transformed.tolist()

for i in range(0,len(tran)):  
 for j in range(0,FC):  
 tran[i][j]=str(tran[i][j])   
len(tran)

9120

list0=[]  
for i in range(1,FC+1):  
 list0.append("Attribute "+str(i))   
list0=",".join(list0)

file = open('dailysportsdataPCA1.txt', 'w')  
file.write("%s\n" % list0)  
count=0  
for item in tran:  
 item=",".join(item)  
 file.write("%s\n" % item)  
 count=count+1

data1 = pd.read\_table('dailysportsdataPCA1.txt', delimiter=',', header=0)

data1.info

<bound method DataFrame.info of Attribute 1 Attribute 2 Attribute 3 Attribute 4 Attribute 5 \  
0 -1.637071 -0.838735 3.347988 -9.458428 -7.743542   
1 -1.749408 -0.970375 3.377247 -9.282271 -7.725263   
2 -1.748035 -1.016446 3.374029 -9.110532 -7.734951   
3 -1.700473 -1.116532 3.374351 -9.011856 -7.627430   
4 -1.590936 -1.198873 3.333404 -8.908969 -7.557411   
... ... ... ... ... ...   
9115 -14.293856 -14.116724 0.828153 -0.751334 0.542264   
9116 -2.732382 -17.674730 -5.787976 7.303718 4.846983   
9117 -6.944206 6.739651 -26.469235 2.396547 -9.336720   
9118 -18.624201 -22.287414 13.653796 8.434307 -8.786071   
9119 -14.740642 -17.646131 12.134152 12.154559 -7.454618   
  
 Attribute 6 Attribute 7 Attribute 8 Attribute 9 Attribute 10 ... \  
0 -23.807102 -9.581168 -7.642276 -11.361543 -0.266466 ...   
1 -23.857871 -9.762795 -7.540563 -11.318303 -0.268322 ...   
2 -23.876501 -9.873329 -7.561801 -11.326528 -0.282362 ...   
3 -23.929857 -9.868562 -7.549670 -11.321522 -0.296492 ...   
4 -23.990856 -9.868686 -7.565691 -11.325635 -0.277031 ...   
... ... ... ... ... ... ...   
9115 15.952670 -0.965580 4.486552 -11.194345 3.239676 ...   
9116 10.606923 3.092321 2.177551 -2.438917 6.407283 ...   
9117 -2.572455 10.498769 30.389892 -14.748274 6.116827 ...   
9118 4.507573 8.573041 -4.581754 -3.421447 -3.207566 ...   
9119 6.225593 10.534479 -7.282178 -2.400744 -10.717747 ...   
  
 Attribute 36 Attribute 37 Attribute 38 Attribute 39 Attribute 40 \  
0 2.073364 0.974592 0.791543 -0.700301 -0.404438   
1 1.984538 0.980023 1.040050 -0.856222 -0.342692   
2 2.000784 1.218666 0.824946 -0.764731 -0.326847   
3 2.038874 1.079182 0.837559 -0.729533 -0.353082   
4 2.000701 1.081808 0.925057 -0.786908 -0.320531   
... ... ... ... ... ...   
9115 9.203627 14.396985 -9.561305 10.104632 -15.151837   
9116 11.361846 2.968851 -9.171966 7.517901 -4.520726   
9117 1.598336 -7.671363 -7.473155 16.088170 18.020874   
9118 -2.551254 -9.069357 16.391776 -21.867964 -0.773144   
9119 -0.260988 -4.717039 13.047815 -17.054062 -4.162224   
  
 Attribute 41 Attribute 42 Attribute 43 Attribute 44 Attribute 45   
0 -0.751582 -0.741852 -0.034034 0.985440 0.534058   
1 -0.634753 -0.762791 0.138708 0.865257 0.542783   
2 -0.702696 -0.805013 0.099017 0.865078 0.519723   
3 -0.731957 -0.783896 0.075949 0.857791 0.500859   
4 -0.655008 -0.781665 0.067867 0.879226 0.516790   
... ... ... ... ... ...   
9115 -9.684862 9.899371 -1.396864 -5.095063 -10.930188   
9116 0.763019 -5.394976 0.142910 -0.913324 3.723007   
9117 -14.690267 -7.987767 2.643529 10.971603 2.095462   
9118 13.575252 -7.642678 0.263808 -2.206267 -1.478670   
9119 13.097214 -7.291438 5.170682 -7.891294 -8.677844   
  
[9120 rows x 45 columns]>