CAPSTONE PROJECT REPORT

Name - Paleti Samuel Yashaswi
Course - Machine learning and AI
Duration - 10 months
Question -3

Exploratory Factor Analysis

Factor analysis is a useful technique to find latent factors that can potentially describe multiple attributes, which is sometimes very useful for dimensionality reduction. Use the Airline Passenger Satisfaction dataset to perform factor analysis. (Use only the columns that represent the ratings given by the passengers, only 14 columns). Choose the best features possible that helps in dimensionality reduction, without much loss in information. .

Data Set Used - Dataset which is used is download from given link (Airplane passenger review dataset)

Method used is Exploratory Factor Analysis

Importing the Data from the location
Selecting the Features which are customer ratings

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv ("C:\\Users\\sampa\\OneDrive\\Desktop\\project3\\test.csv")



df1=df.iloc[:,9:23]
print(df1.head())
x=df1.values
x_mean=np.mean(x,axis=0)
x_n=x-np.matrix(x_mean)
x_n=x_n.T
print(x_n.shape)

c1=np.cov(x_n)
c2=np.corrcoef(x_n)
ax=sns.heatmap(c2,cmap='Blues')
```

Computing eigen vectors and values

```
eig_val,eig_vec=np.linalg.eig(c1)
eig_sorted=np.sort(eig_val)[::-1]
arg_sort=np.argsort(eig_val)[::-1]

eig_vec_ls=[]
eig_val_ls=[]
imp_vec= arg_sort[:2]
for i in imp_vec:
    eig_vec_ls.append(eig_vec[:,i])
    eig_val_ls.append(eig_val[i])
print(eig_vec_ls)
print(eig_val_ls)
```

```
[array([-9.95541243e-06, -4.05623633e-05,  2.75807511e-04, -9.27028708e-04,
       -7.64759529e-04, -1.00357593e-03, -9.01610175e-04, -8.94214056e-04,
        5.42701294e-04, 5.85781336e-05, -6.76848721e-04, -1.66929744e-03,
       -5.81218214e-04,  9.99995989e-01]), array([0.06371542, 0.1123214 , 0.02345764, 0.35901916, 0.2816334 ,
       0.39362684, 0.46860302, 0.25513111, 0.21281894, 0.21666156,
       0.1708363 , 0.21961118, 0.40655392, 0.00218294])]
[1400.5041098886813, 6.222626676088535]
```

Calculating the Lamda and Vector V

```
eig_val_arr= np.array(eig_val_ls)
lamda_1=np.diag(eig_val_arr)
print(lamda_1)
eig_vec_mat=np.matrix(eig_vec_ls).T
V=eig_vec_mat@np.sqrt(lamda_1)
print(V)
```

```
[[1400.50410989    0.        ]
 [   0.            6.22262668]]
[[-3.72564483e-04 1.58939351e-01]
 [-1.51797788e-03 2.80187910e-01]
 [ 1.03216299e-02 5.85155391e-02]
 [-3.46924824e-02 8.95580221e-01]
 [-2.86198327e-02 7.02539957e-01]
 [-3.75571328e-02 9.81909759e-01]
 [-3.37412368e-02 1.16893928e+00]
 [-3.34644496e-02 6.36429484e-01]
 [ 2.03096786e-02  5.30880970e-01]
 [ 2.19218764e-03  5.40466445e-01]
 [-2.53299193e-02 4.26154454e-01]
 [-6.24706350e-02 5.47824342e-01]
 [-2.17511092e-02  1.01415661e+00]
 [ 3.74231596e+01 5.44538650e-03]]
```

Computing the S matrix

```
var_ls=[]
x_var = np.var(x_n,axis=1)
x_var=np.ravel(x_var)
print(x_var.shape)
print(x_var)
for i in range(V.shape[0]):
    s=np.sum(np.square(np.ravel(V[i,:])))
    sig_2=x_var[i]-s
    var_ls.append(sig_2)

var_ls=np.array(var_ls)
S=np.diag(var_ls)
print(S.shape)
print(S)
```

Plotting the points after reducing into 2 features

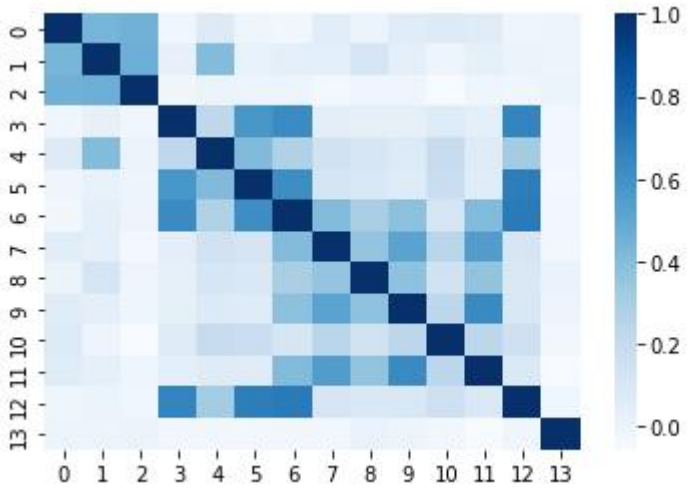
```
c1_inv=np.linalg.inv(c1)
W=V.T@c1_inv
print(W.shape)
print(W)
```

```
(2, 14)
[[-2.66021703e-07 -1.08387963e-06 7.36993902e-06 -2.47714249e-05]
  -2.04353793e-05 -2.68168672e-05 -2.40922084e-05 -2.38945743e-05
   1.45016915e-05  1.56528469e-06 -1.80862870e-05 -4.46058205e-05
  -1.55309143e-05 2.67212066e-02]
 [ 2.55421640e-02  4.50272730e-02  9.40367181e-03  1.43923180e-01
   1.12900869e-01 1.57796669e-01 1.87853031e-01 1.02276662e-01
   8.53146103e-02 8.68550330e-02 6.84846571e-02 8.80374753e-02
   1.62978861e-01 8.75094520e-04]]
```

```
z=W@x_n
z1=z.T
plt.scatter(np.ravel(z1[:,0]),np.ravel(z1[:,1]))
plt.show
```

OUTPUT-

Heatmap of the taken data



Scatter plot of data after reducing into 2d feature

```
: <function matplotlib.pyplot.show(close=None, block=None)>
```