Capstone Project Report

Name: Sumit Patel

Course: AI and ML (Batch – AUG 2020)

Duration: 10 months

Exploratory Factor Analysis

Problem Statement:

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.

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic. Second and easier option is to download anaconda and use its anaconda prompt to run the commands.

To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages:

pip install numpy

pip install matplotlib

pip install pandas

pip install seaborn

If you have chosen to install anaconda then run below commands in anaconda prompt to install these packages:

conda install -c anaconda numpy

conda install -c anaconda matplotlib

conda install -c anaconda pandas

conda install -c anaconda seaborn

Dataset used:

Dataset Link: https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction

Importing the libraries and loading dataset.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

Analyzing the data and removing unwanted columns

```
train.columns
```

```
Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction'],
      dtype='object')
```

```
unwanted_columns = ['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction']
for col in unwanted_columns:
        train.pop(col)
        test.pop(col)
train.head()
```

Zero centering the data

```
x = np.array(train)
x_mean = np.mean(x,axis=0)
x_n = x - np.matrix(x_mean)
x_n = x_n.T ## Converts row vectors to column vectors
print(x_n.shape)
```

```
(14, 103904)
```

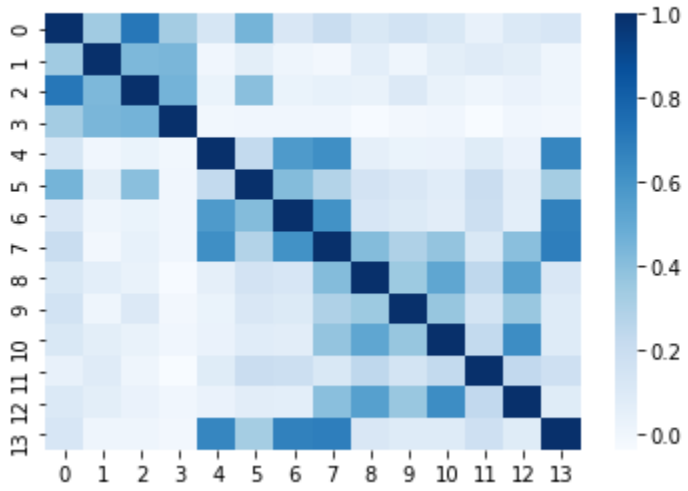
```
# Finding the covariance matrix and coefficients
C1 = np.cov(x_n)
eig_val,eig_vec = np.linalg.eig(C1)

C2 = np.corrcoef(x_n)## Corr(x,y) = Cov(x,y)/sqrt(Var(x)*Var(y))
```

```
C1.shape, C2.shape
```

```
((14, 14), (14, 14))
```

```
ax = sns.heatmap(C2,cmap='Blues')
```



Extract the eigen vectors and eigen values

```
eig_sorted = np.sort(eig_val)[::-1]
arg_sort = np.argsort(eig_val)[::-1]

eig_vec_ls = []
eig_val_ls = []
imp_vec = arg_sort[:5] # Choose the number of imp vecs form the heat map
for i in imp_vec:
    e_1 = eig_vec[:,i]
    lambda_1 = eig_val[i]
    eig_vec_ls.append(e_1)
    eig_val_ls.append(lambda_1)
print(np.matrix(eig_vec_ls).shape)
print(np.matrix(eig_val_ls).shape)
```

```
(5, 14)
(1, 5)
```

Estimate V (The Fator loading Matrix)

```
eig_val_arr  = np.array(eig_val_ls)
lambda_1 = np.diag(eig_val_arr) # Diagonal array with the Eigen values
eig_vec_mat = np.matrix(eig_vec_ls).T
V = eig_vec_mat@np.sqrt(lambda_1)
print(lambda_1)
print('---'*20)
print(V)
print('---'*20)
print(V.shape)
```

Compute $\sigma 2i, i$=0,1,2,...13 and estimate the Source (S)

```
x_var = np.var(x_n,axis=1)
print(x_var.shape)
x_var = np.ravel(x_var)
print(x_var.shape)
```

```
(14, 1)
(14,)
```

Dimensionality reduction transformation

```
x_n.shape
```

```
(14, 103904)
```

```
C1_inv = np.linalg.inv(C1)
W = V.T@C1_inv # Weight Matrix
print(W.shape)
print(W)
```

```
(5, 14)
[[-0.10580392 -0.06063097 -0.08453211 -0.03571301 -0.1262144  -0.12137862
  -0.13922513 -0.16629195 -0.09503427 -0.07963447 -0.07888021 -0.06365334
  -0.0793249  -0.14230927]
 [-0.18451462 -0.2319447  -0.2435102  -0.17897573  0.09580334 -0.04581331
   0.09430532  0.10427343  0.03144506  0.01118984  0.02123786  0.02152568
   0.0233128   0.10390693]
 [-0.02864458  0.00445306 -0.04052911 -0.03457271 -0.17240676 -0.07868107
  -0.14585269 -0.00203799  0.23625016  0.1964674   0.22882555  0.08858214
   0.23304814 -0.14682961]
 [-0.20943694  0.39706671 -0.16921011  0.244008    0.13441971 -0.39803833
   0.03089906  0.09707286  0.01519059 -0.10610572  0.04676241 -0.01416551
   0.05582059  0.10096707]
 [-0.12670416  0.21661685 -0.09534508 -0.12503402 -0.09192071  0.1952928
   0.08914799 -0.19542992  0.00228309 -0.18885173 -0.0383107   0.63328028
  -0.04180564  0.01146245]]
```

```
# Finding the latent factors z.
z = W@x_n
z1 = z.T
print(z.T.shape)
```

```
(103904, 5)
```