Guided Projects: Feature Engineering

Exploratory Factor Analysis (EDA)

Name	**Lakshmi Thirunavukkarasu**
Course	**AI and ML (Batch 5)**
Problem Statement	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

Software requirements perquisites

1. Anaconda
2. Python 3.8
3. Python Packages
    - NumPy
    - Pandas
    - Seaborn
    - Matplotlib

Steps

1. Download the test and train dataset from Kaggle https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction and saved it where the scripts are saved.

2. Combine train and Test Dataset and construct a data frame.

Generate the dataframe from the excel file

```
[173]: # Importing the dataset
df_train = pd.read_csv("train.csv") ## As the dataset is in excel format
df_test = pd.read_csv("test.csv") ## As the dataset is in excel format
df = pd.concat([df_train,df_test])
df.head(5)
# We have a total of 99 datapoints and 14 features
```

3. Remove the columns that doesn't represent ratings from the dataset.

```
In [175]: df.drop(columns = ['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age',
                            'Type of Travel', 'Class', 'Flight Distance', 'Departure Delay in Minutes',
                            'Arrival Delay in Minutes','satisfaction'], axis=1,inplace=True)
```

4. Zero center the input data set.

Zero Centering the Data

Find $\tilde{x}_n$ from x_n

```
In [5]: x = df.values
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, 99)
```

5. Generate Covariance Matrix

```
n [6]: C1 = np.cov(x_n)
C2 = np.corrcoef(x_n)## Corr(x,y) = Cov(x,y)/sqrt(Var(x)*Var(y))
```

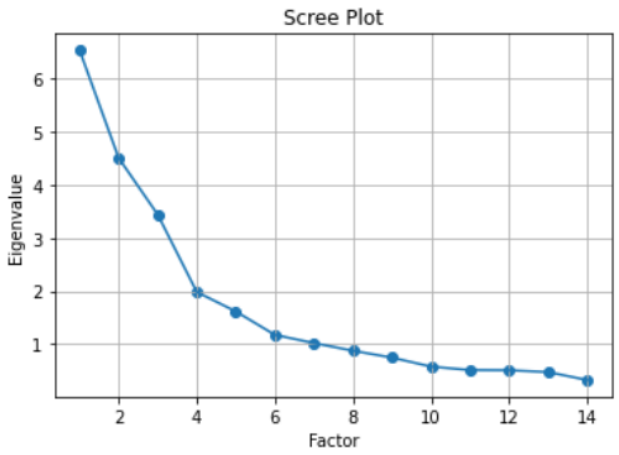
6. Extract eigen vectors and eigen values for the covariance matrix and sort the eigen values in descending order.

```
In [10]: eig_val,eig_vec = np.linalg.eig(C1)
         eig_sorted = np.sort(eig_val)[::-1]
         arg_sort = np.argsort(eig_val)[::-1]
         print("eigen values", eig_sorted)
```

7. Generate the Scree plot to the identify the number of latest factors

Plot Scree Plot to identify the factors

```
In [184]: xvals = range(1, df.shape[1]+1)
          plt.scatter(xvals, eig_sorted)
          plt.plot(xvals,eig_sorted)
          plt.title('Scree Plot')
          plt.xlabel('Factor')
          plt.ylabel('Eigenvalue')
          plt.grid()
```



8. Generate factor loading matrix for 8 latent factors.

Build the Vector based on the latent factors (=8)

```
In [12]: eig_vec_ls = []
         eig_val_ls = []
         imp_vec = arg_sort[:8]
         for i in imp_vec:
             eig_vec_ls.append(eig_vec[:,i])
             eig_val_ls.append(eig_val[i])
```

Estimate V

```
In [13]: eig_val_arr  = np.array(eig_val_ls)
         lambda_1 = np.diag(eig_val_arr)
         eig_vec_mat = np.matrix(eig_vec_ls).T
         V = eig_vec_mat@np.sqrt(lambda_1)
         print(V.shape)

         Factor_Loading = pd.DataFrame(np.matrix(V),index = df.columns)
         Factor_Loading
```

```
(14, 8)
```

9. Generate S(additional source)

Estimate S(additional source)

In [14]:
```
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)
```

9. Perform dimensionality reduction

Dimensionality reduction transformation

In [189]:
```
C1_inv = np.linalg.inv(C1)
W = V.T@C1_inv
print(W.shape)
print(W)
```