

Guided Project Report

Exploratory Factor Analysis

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement: Build a machine learning model using EFA for dimensionality reduction

Prerequisites

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

Python 3.8 or higher versions 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-external-command/>. 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 3.8 then run below commands in command prompt/terminal to install these packages `pip install -U scikit-learn` `pip install numpy` `pip install scipy` if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages `conda install -c scikit-learn` `conda install -c anaconda numpy` `conda install -c anaconda scipy`

Dataset used

The data source is airline satisfaction datasets(14 columns which contribute to passenger ratings) dataset provided in the kaggle <https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction>. The dataset is divided into 80 percent training data and 20 percent a test data. We focused mainly on 14 columns to understand what factors are highly

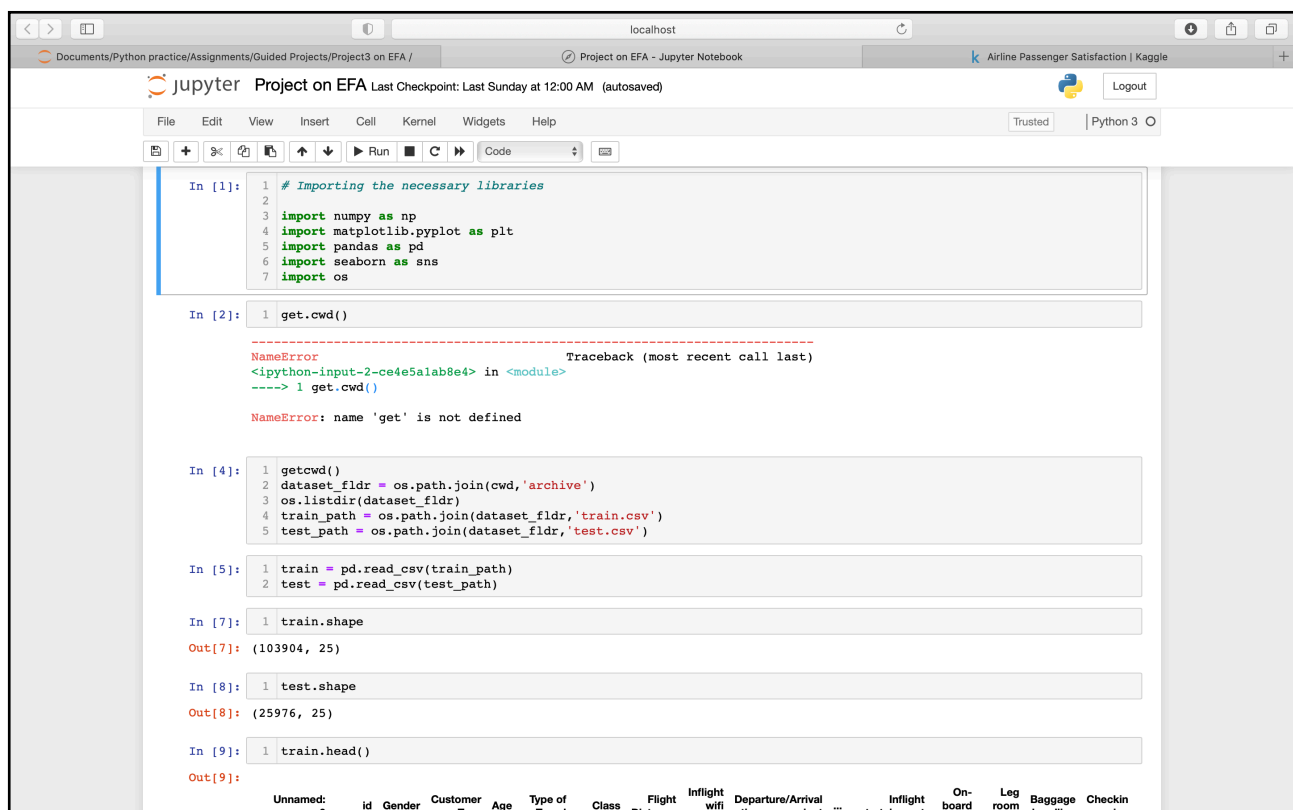
correlated to a satisfied or dissatisfied passenger. Train data is having more than one lakh records and test data is around 25 thousand records.

Method used for detection

EFA

Feature subset dataset -> Zero centring dataset -> correlation among features -> based on variance and Eigen values -> dimensionality reduction transformation

Importing the libraries and capturing images:



The screenshot shows a Jupyter Notebook titled "Project on EFA" with a last checkpoint from Sunday at 12:00 AM. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and other functions. The code is written in Python 3.

```
In [1]: 1 # Importing the necessary libraries
        2
        3 import numpy as np
        4 import matplotlib.pyplot as plt
        5 import pandas as pd
        6 import seaborn as sns
        7 import os

In [2]: 1 getcwd()

-----
NameError                                Traceback (most recent call last)
<ipython-input-2-ce4e5a1ab8e4> in <module>
----> 1 getcwd()

NameError: name 'get' is not defined

In [4]: 1 getcwd()
        2 dataset_fldr = os.path.join(cwd, 'archive')
        3 os.listdir(dataset_fldr)
        4 train_path = os.path.join(dataset_fldr, 'train.csv')
        5 test_path = os.path.join(dataset_fldr, 'test.csv')

In [5]: 1 train = pd.read_csv(train_path)
        2 test = pd.read_csv(test_path)

In [7]: 1 train.shape
Out[7]: (103904, 25)

In [8]: 1 test.shape
Out[8]: (25976, 25)

In [9]: 1 train.head()
Out[9]:
```

Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi	Departure/Arrival time convenient	...	Inflight entertainment	On-board	Leg room	Baggage handling	Checkin service
0															

Importing the train and test data and feature subset

The screenshot shows a Jupyter Notebook titled "Project on EFA" with a last checkpoint at 12:00 AM. The notebook is running on a local host. The first cell, labeled "In [9]:", contains the code `train.head()`. The output, labeled "Out[9]:", displays the first five rows of the training data as a table with 25 columns. The columns include "Unnamed: 0", "id", "Gender", "Customer Type", "Age", "Type of Travel", "Class", "Flight Distance", "Inflight wifi service", "Departure/Arrival time convenient", "Inflight entertainment", "On-board service", "Leg room service", "Baggage handling", and "Checkin service". The second cell, labeled "In [10]:", contains code for feature subset selection, including comments and operations to split the data into training and testing sets. The output, labeled "Out[10]:", shows the shapes of the resulting datasets and the first five rows of the training subset as a table with 14 columns, including "Inflight wifi service", "Departure/Arrival time convenient", "Ease of Online booking", "Gate location", "Food and drink", "Online boarding", "Seat comfort", "Inflight entertainment", "Inflight service", "On-board service", "Leg room service", "Baggage handling", "Checkin service", and "Cleanliness".

```
In [9]: 1 train.head()
```

Out[9]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	4
1	1	5047	Male	dissloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	1
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4	3	4	4
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2	5	3	1
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...	3	3	4	4	3

5 rows x 25 columns

```
In [10]: 1 # Feature subset selection mainly concerned to the airline satisfaction
2 X_train = train.iloc[:,8:-3]
3 X_test = test.iloc[:,8:-3]
4 print(X_train.shape)
5 print(X_test.shape)
6 X_train.head()
```

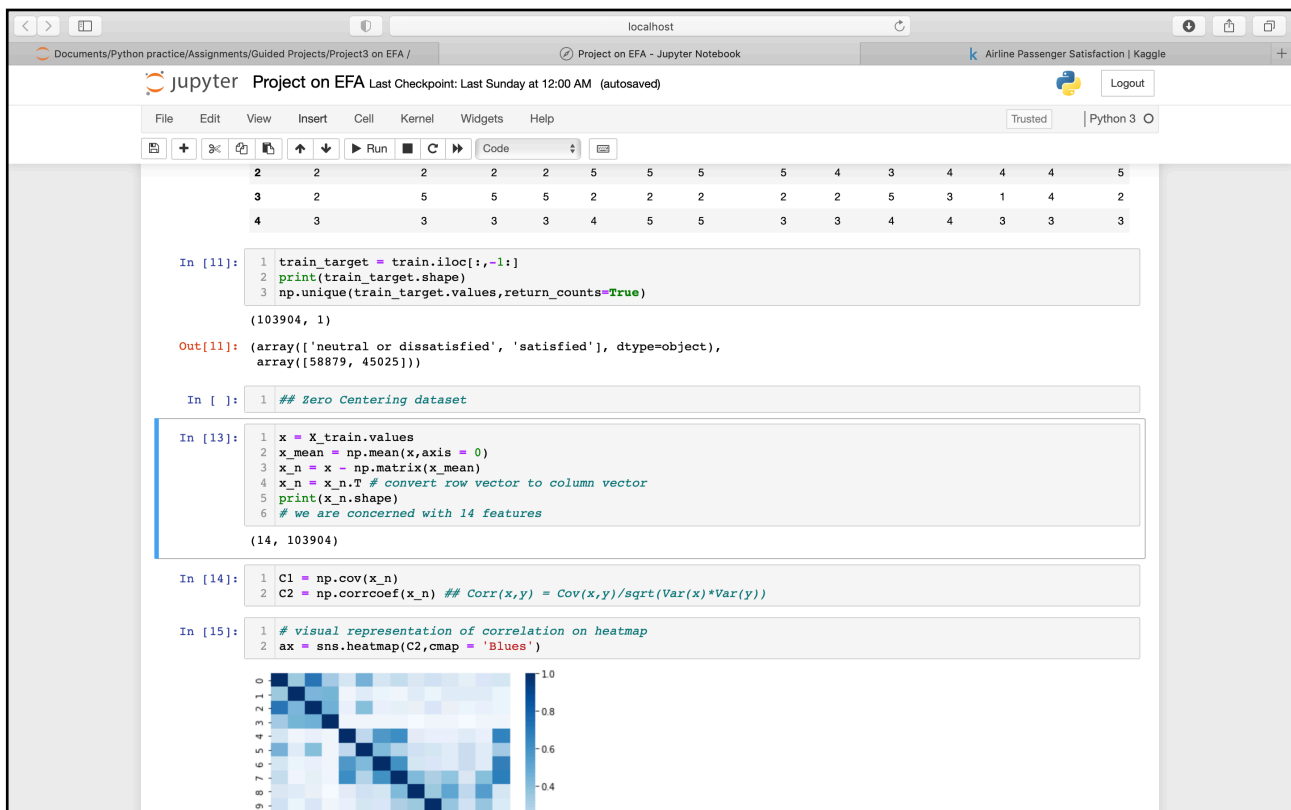
(103904, 14)
(25976, 14)

Out[10]:

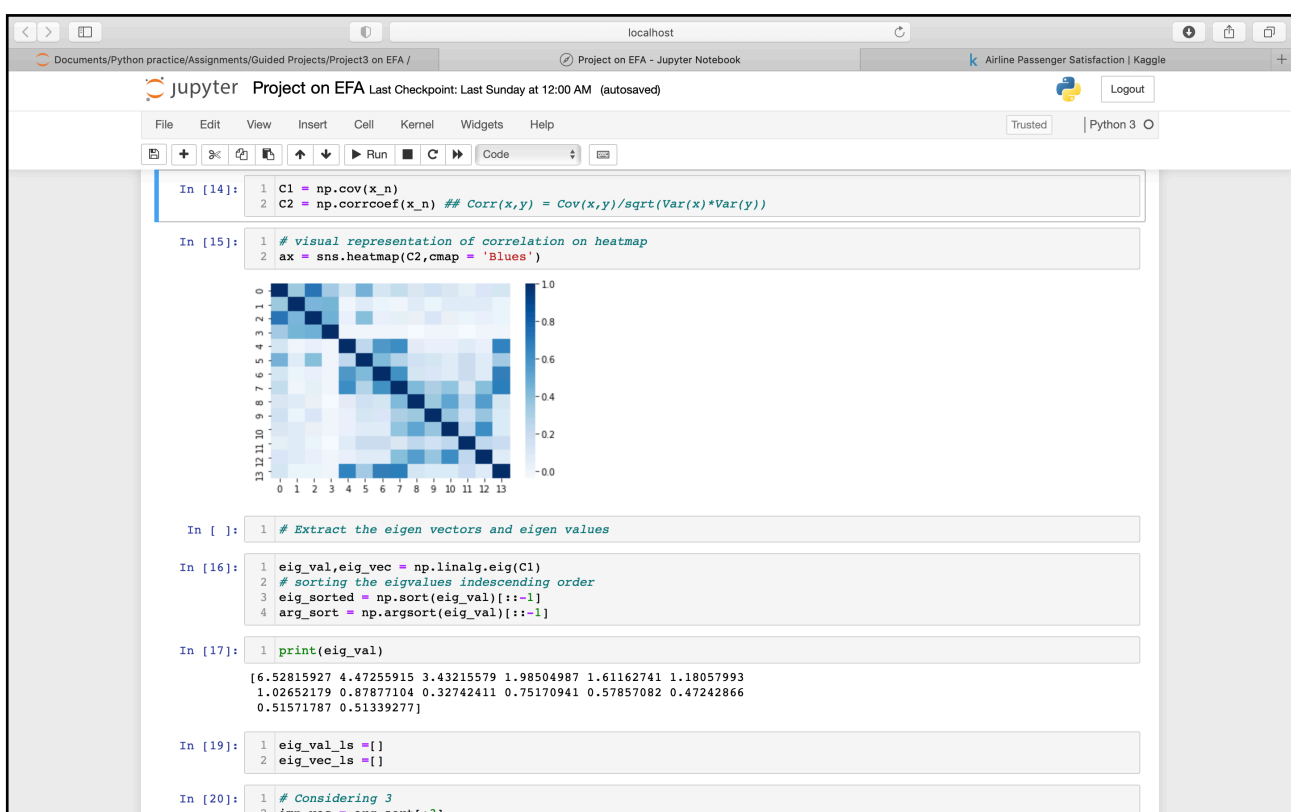
	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	Inflight service	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanliness
0	3	4	3	1	5	3	5	5	4	3	4	4	5	5	
1	3	2	3	3	1	3	1	1	1	5	3	1	4	1	
2	2	2	2	2	5	5	5	5	4	3	4	4	4	5	
3	2	5	5	5	2	2	2	2	2	5	3	1	4	2	
4	3	3	3	3	4	5	5	3	3	4	4	3	3	3	

Splitting the training and testing data into 80 :20.

Zero Centring data

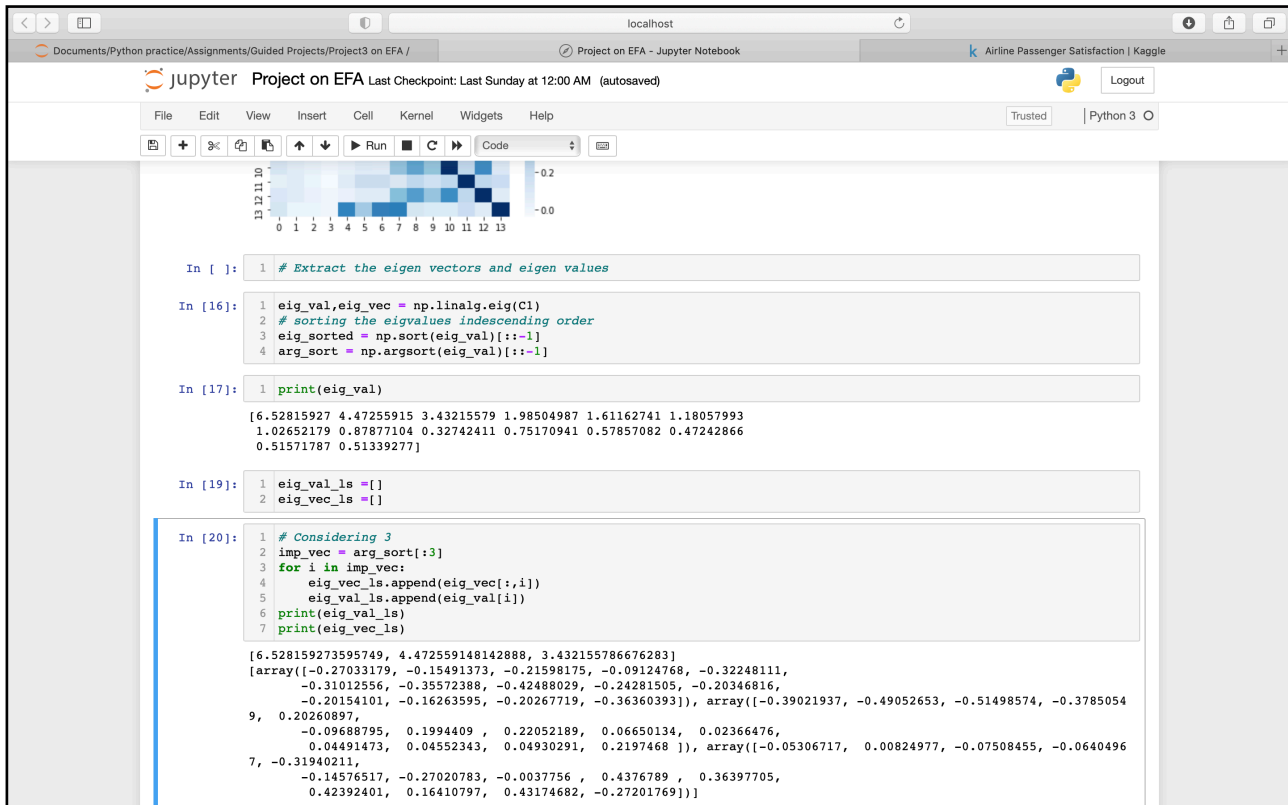


Visual Representation of correlation on heat map



Considering 3 correlation arbitrarily

Extracting the Eigen vectors and values



Calculating Variance

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project3 on EFA / Project on EFA - Jupyter Notebook Airline Passenger Satisfaction | Kaggle
jupyter Project on EFA Last Checkpoint: Last Sunday at 12:00 AM (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [22]:
1 ## Calculating V
2 eig_val_arr = np.array(eig_val_ls)
3 lambda_1 = np.diag(eig_val_arr)
4 print(lambda_1)
5 eig_vec_mat = np.matrix(eig_vec_ls).T
6 V = eig_vec_mat*np.sqrt(lambda_1)
7 print(V)

[[6.52815927 0. 0. ]
 [0. 4.47255915 0. ]
 [0. 0. 3.43215579]]
[[-0.69070483 -0.82525254 -0.09831266]
 [-0.39580866 -1.03738639 0.01528359]
 [-0.55183905 -1.08911378 -0.13910223]
 [-0.23314023 -0.80047953 -0.11865892]
 [-0.8239477 0.42848608 -0.59172685]
 [-0.79237896 -0.20490276 -0.27004569]
 [-0.90888385 0.42178611 -0.50058914]
 [-1.08558032 0.46636909 -0.00699471]
 [-0.62039882 0.14063987 0.81084737]
 [-0.5198665 0.05004723 0.67430674]
 [-0.51494257 0.09498759 0.78536494]
 [-0.41553912 0.09627489 0.30402772]
 [-0.51784555 0.1042679 0.79985753]
 [-0.9290176 0.46472989 -0.50394211]]

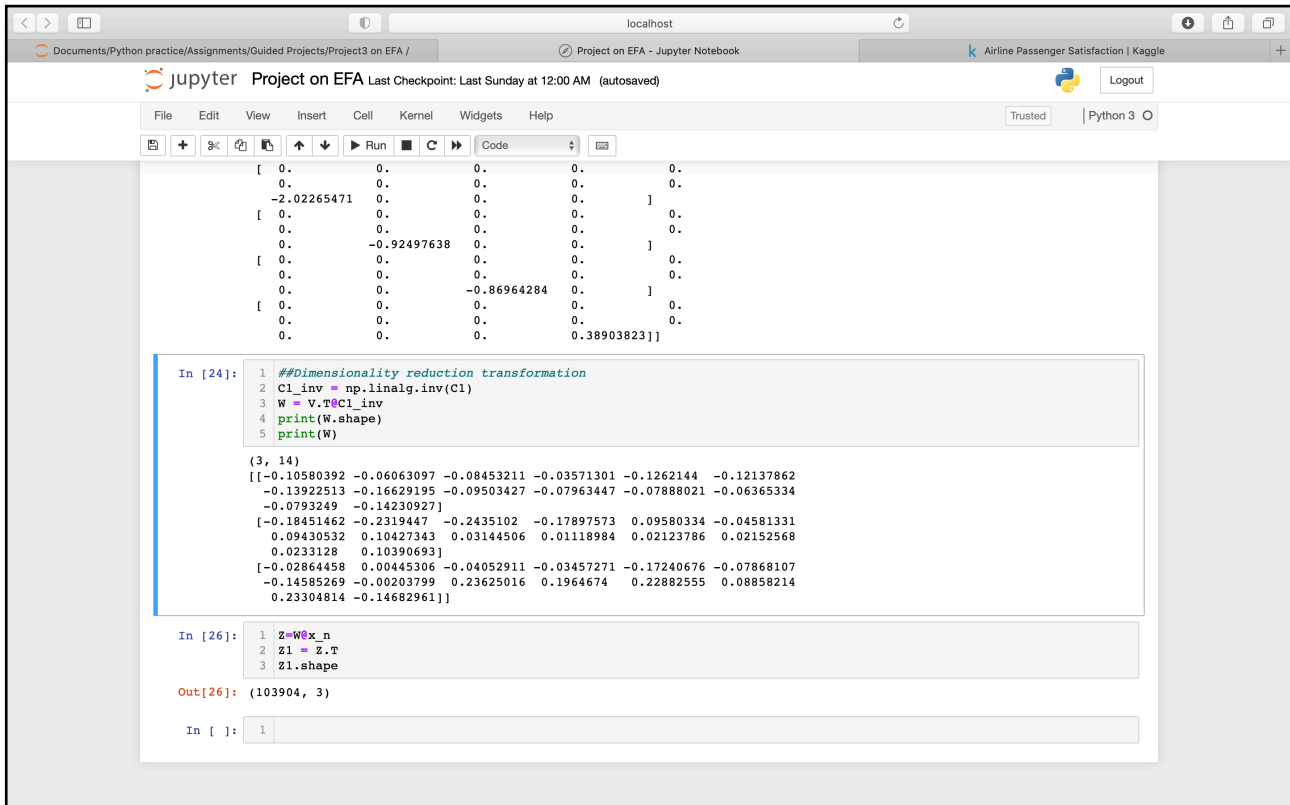
In [23]:
1 ## Calculating sigma square i = 0,1,...,13 and S
2 var_ls = []
3 x_var = np.var(x_n,axis=1)
4 x_var = np.ravel(x_var)
5 print(x_var.shape)
6 print(x_var)
7 for i in range (V.shape[0]):
8     s = np.sum(np.square(np.ravel(V[i:])))
9     sig_2 = x_var[i] - s
10    var_ls.append(sig_2)
11 var_ls = np.array(var_ls)
12 S = np.diag(var_ls)
13 print(S)

(14,)
[1.76311414 2.32583197 1.95698483 1.63229974 1.76764022 1.82115689
 1.73997514 1.77684714 1.65984098 1.73079886 1.39451944 1.60121119
 1.38217027 1.72204345]
[[-12.66976007 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.
 0. -10.93926195 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. -10.07504049 0. 0.
 0. 0. 0. 0. 0.
 0. 0. 0. -8.88968099 0.
 0. 0. 0. 0. 0.
 0. 0. 0. 0. -8.04513873
 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 -6.77899125 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. -6.11739877 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. -4.82596389 0.
 0. 0. 0. 0. ]
```

```
localhost
Documents/Python practice/Assignments/Guided Projects/Project3 on EFA / Project on EFA - Jupyter Notebook Airline Passenger Satisfaction | Kaggle
jupyter Project on EFA Last Checkpoint: Last Sunday at 12:00 AM (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [23]:
1 ## Calculating sigma square i = 0,1,...,13 and S
2 var_ls = []
3 x_var = np.var(x_n,axis=1)
4 x_var = np.ravel(x_var)
5 print(x_var.shape)
6 print(x_var)
7 for i in range (V.shape[0]):
8     s = np.sum(np.square(np.ravel(V[i:])))
9     sig_2 = x_var[i] - s
10    var_ls.append(sig_2)
11 var_ls = np.array(var_ls)
12 S = np.diag(var_ls)
13 print(S)

(14,)
[1.76311414 2.32583197 1.95698483 1.63229974 1.76764022 1.82115689
 1.73997514 1.77684714 1.65984098 1.73079886 1.39451944 1.60121119
 1.38217027 1.72204345]
[[-12.66976007 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.
 0. -10.93926195 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. -10.07504049 0. 0.
 0. 0. 0. 0. 0.
 0. 0. 0. -8.88968099 0.
 0. 0. 0. 0. 0.
 0. 0. 0. 0. -8.04513873
 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 -6.77899125 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. -6.11739877 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. -4.82596389 0.
 0. 0. 0. 0. ]
```

Dimensionality Reduction Transformation



The screenshot shows a Jupyter Notebook interface with the following content:

Project on EFA Last Checkpoint: Last Sunday at 12:00 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

```
[ 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 -2.02265471 0. 0. 0. ]
[ 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. -0.92497638 0. 0. ]
[ 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. -0.86964284 0. ]
[ 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0.
 0. 0. 0. 0.38903823]]
```

In [24]:

```
1 ##Dimensionality reduction transformation
2 C1_inv = np.linalg.inv(C1)
3 W = V.T@C1_inv
4 print(W.shape)
5 print(W)
```

```
(3, 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]]
```

In [26]:

```
1 Z=W@X_n
2 Z1 = Z.T
3 Z1.shape
```

Out[26]: (103904, 3)

In []:

```
1
```