Lab Assignment 8
Prakhar Gupta
B21AI027

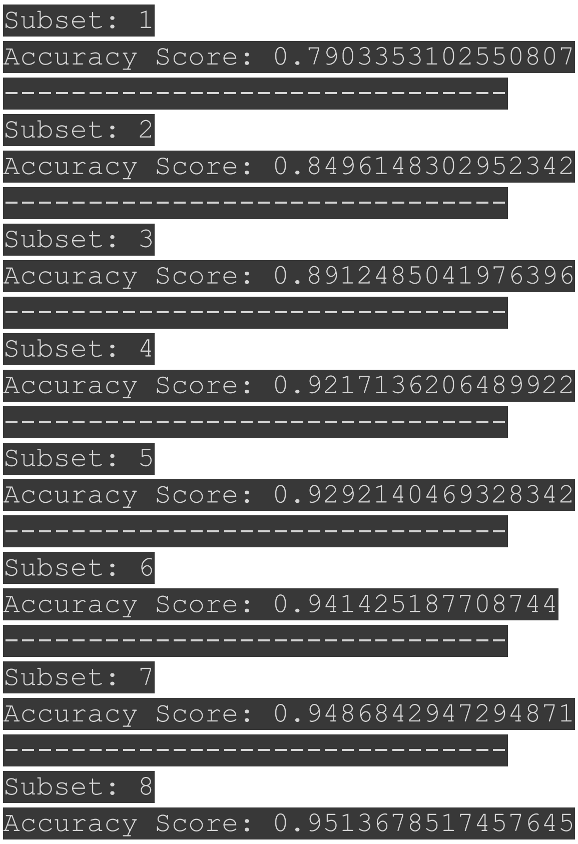
Question 1:

Part 1-

- Downloaded dataset using **wget** command called using **os.system**
- Loaded the **.csv** file into df using pd.read_csv
- Checked for not filled rows using **df.isnull().sum()**
- Used **df.describe()** to get insights about the dataset
- Dropped **'Id' and 'Unnamed : 0'** column
- Converted **['Gender', 'Customer Type', 'Type of Travel', 'Class', 'satisfaction']** to label using **LabelEncoder()**
- Applied **MinMaxScalar()** to every column to normalise data
- Converted **df to X,y**

Part 2-

- Created SFS by embedding Decision Tree



```
Subset: 1
Accuracy Score: 0.7903353102550807
------------------------------
Subset: 2
Accuracy Score: 0.8496148302952342
------------------------------
Subset: 3
Accuracy Score: 0.8912485041976396
------------------------------
Subset: 4
Accuracy Score: 0.9217136206489922
------------------------------
Subset: 5
Accuracy Score: 0.9292140469328342
------------------------------
Subset: 6
Accuracy Score: 0.941425187708744
------------------------------
Subset: 7
Accuracy Score: 0.9486842947294871
------------------------------
Subset: 8
Accuracy Score: 0.9513678517457645
```

```
------------------------------
Subset: 9
Accuracy Score: 0.9521497395022156
------------------------------
Subset: 10
Accuracy Score: 0.950653519908512

Best 10 features selected by SFS:
Customer Type
Type of Travel
Class
Inflight wifi service
Gate location
Online boarding
Seat comfort
Inflight entertainment
Baggage handling
Inflight service
```

Part 3-

- # SFS (forward=True, floating=False)

  ```
  cv_scores: [0.94883972 0.94972779 0.94999614 0.95134759]
  ```

- # SBS (forward=False, floating=False)

  ```
  cv_scores: [0.95084752 0.95104058 0.9507684  0.95281489]
  ```

- # SFFS (forward=True, floating=True)

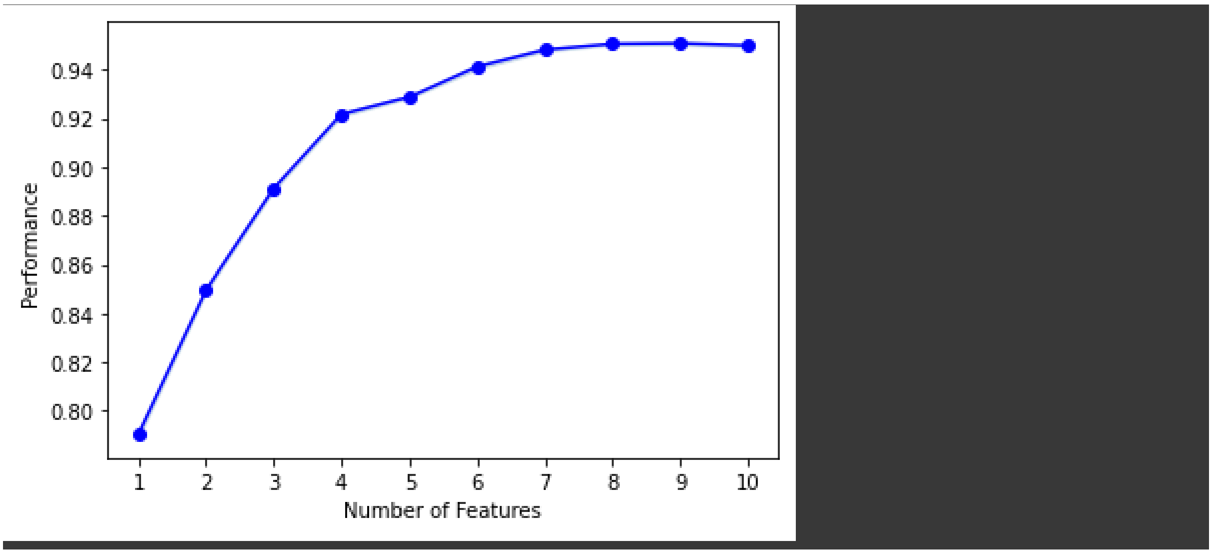
  ```
  cv_scores: [0.95107919 0.95127225 0.95011198 0.95331686]
  ```

- # SBFS (forward=False, floating=True)

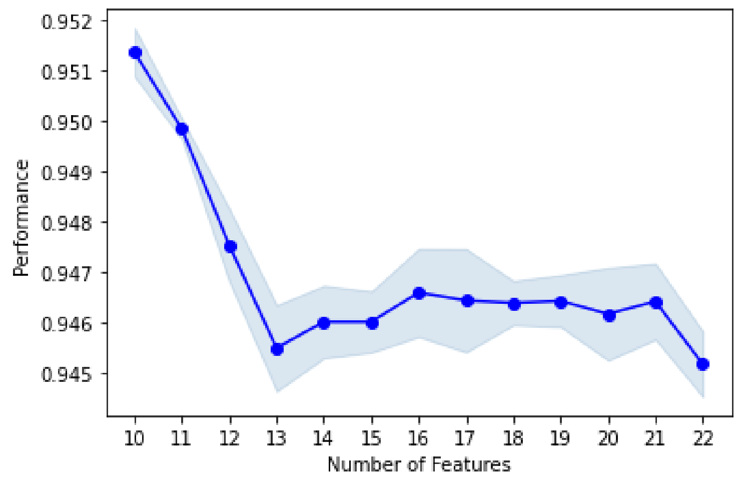
  ```
  cv_scores: [0.95123364 0.95100197 0.95100008 0.95316241]
  ```

Part 4-

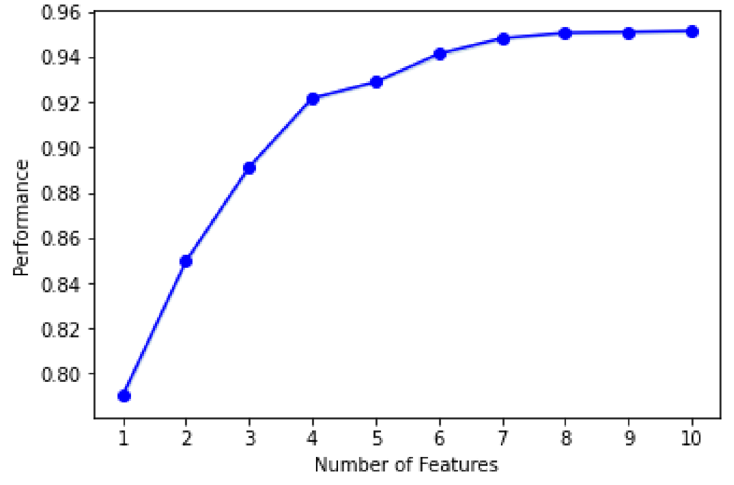
SFS



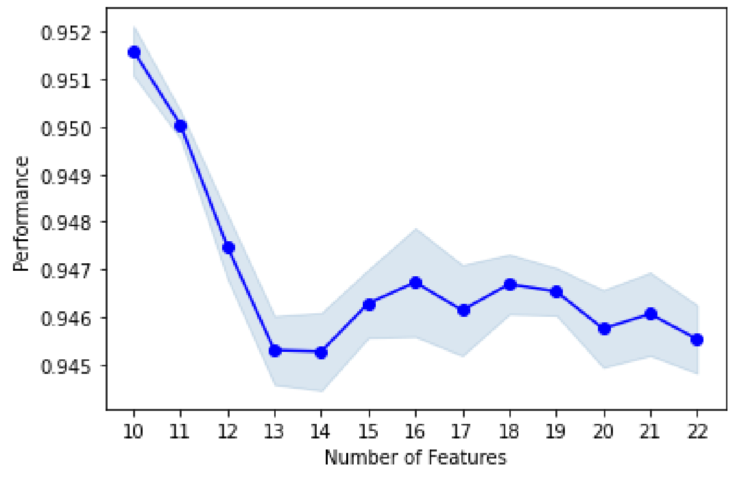
SBS



SFFS



SBFS



Part 5-

- Implemented **bi_directional_feature_selection** from scratch
- It performs forward_selection as well as backward_selection and also do feature selection based on the Metric given to it

Part 6-

- Reduced data size to 1000 examples as it was taking a lot of time on 1000 examples only

-
  ```
  Accuracy score using Decision Tree (metric - Accuracy Measure
  using SVM Classifier): 0.875
  ```

-
  ```
  Accuracy score using Decision Tree (metric - Information
  Measures: Information gain): 0.81
  ```

-
  ```
  Accuracy score using Decision Tree (metric - Distance Measure:
  City-Block Distance:) 0.855
  ```

Question 2:

Part 1-

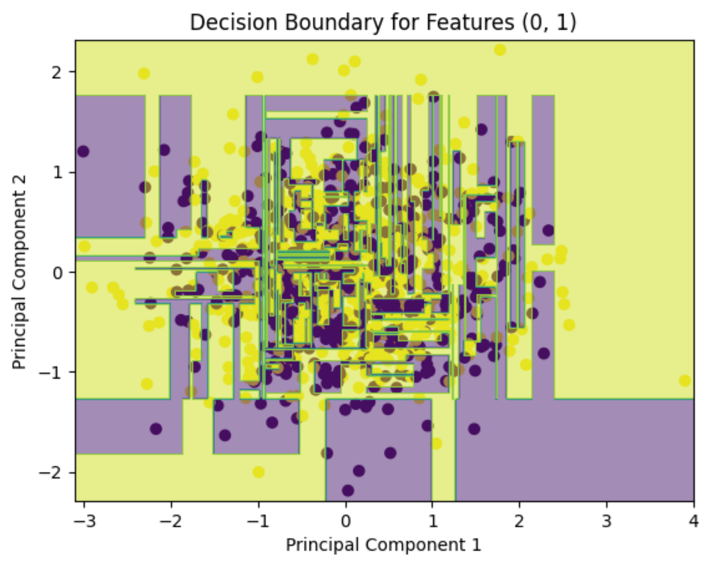
- Created **X of 1000 points** from given covariance matrix
- Made labels **y** using x.v as given in question
- Plotted 3D graph of data using **plotly.graph_objs**

Part 2-

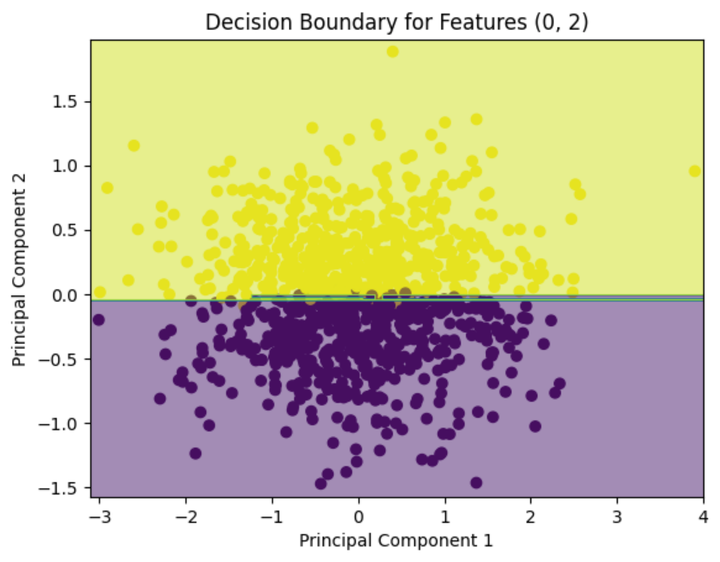
- Performed **PCA** with **3 components** using **sklearn**
- Changed the X with this new transformed data

Part 3-

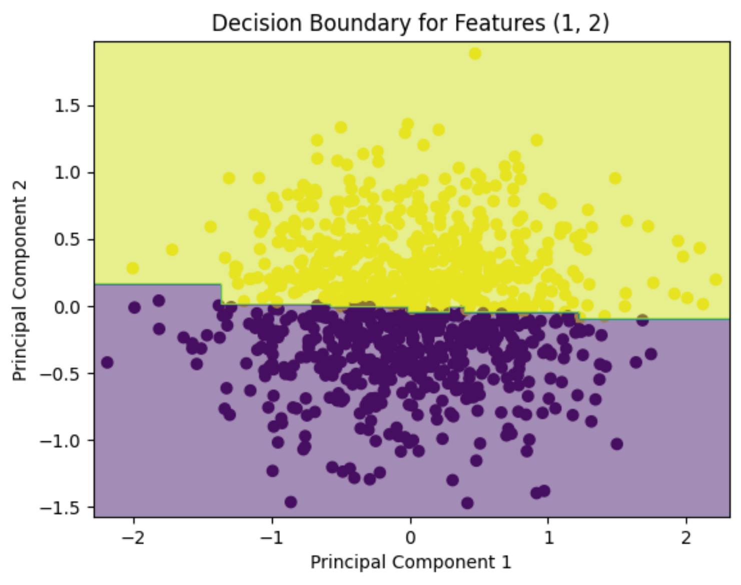
- Fitted Decision Tree for every subset-set of features of size 2 and plotted their decision boundaries superimposed with the data.



Decision Boundary for Features (0, 1)



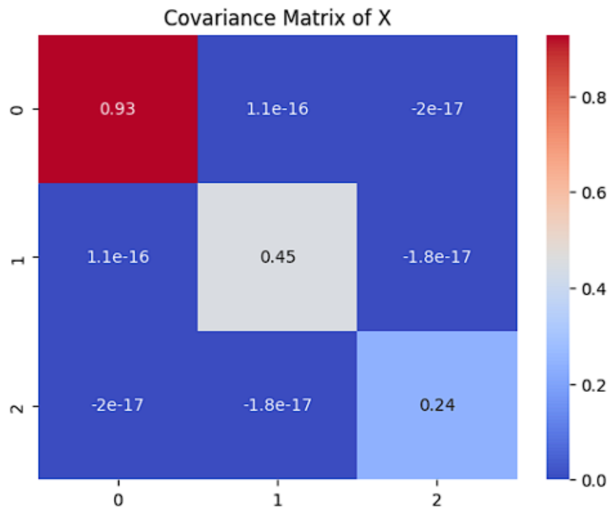
Decision Boundary for Features (0, 2)



Decision Boundary for Features (1, 2)

Part 4-

- Plotted **heat_map** of **covariance matrix** of transformed_data



	0	1	2
0	0.93	1.1e-16	-2e-17
1	1.1e-16	0.45	-1.8e-17
2	-2e-17	-1.8e-17	0.24

- The features (0,1) would be selected if runned PCA with 2 components
- # Features (0,2) `Test Accuracy: 0.985`
- # Features (1,2) `Test Accuracy: 1.0`
- # Features (0,1) `Test Accuracy: 0.52`