Lab 8 Report

Nakkina Vinay (B21AI023)

Question 1

- Installing all the necessary libraries

Subpart 1:

- Getting the dataset and using the head() to print it
- Getting the information of the dataset using info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Unnamed: 0                         103904 non-null  int64
 1   id                                 103904 non-null  int64
 2   Gender                             103904 non-null  object
 3   Customer Type                      103904 non-null  object
 4   Age                                103904 non-null  int64
 5   Type of Travel                     103904 non-null  object
 6   Class                              103904 non-null  object
 7   Flight Distance                    103904 non-null  int64
 8   Inflight wifi service              103904 non-null  int64
 9   Departure/Arrival time convenient  103904 non-null  int64
 10  Ease of Online booking             103904 non-null  int64
 11  Gate location                      103904 non-null  int64
 12  Food and drink                     103904 non-null  int64
 13  Online boarding                    103904 non-null  int64
 14  Seat comfort                       103904 non-null  int64
 15  Inflight entertainment             103904 non-null  int64
 16  On-board service                   103904 non-null  int64
 17  Leg room service                   103904 non-null  int64
 18  Baggage handling                   103904 non-null  int64
 19  Checkin service                    103904 non-null  int64
 20  Inflight service                   103904 non-null  int64
 21  Cleanliness                        103904 non-null  int64
 22  Departure Delay in Minutes         103904 non-null  int64
 23  Arrival Delay in Minutes           103594 non-null  float64
 24  satisfaction                       103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

- Dropping the unnecessary columns like ‘Unnamed: 0’
- Encoding the categorical variables using Label Encoding
- Replacing any ‘-‘ values with Nan and converting all columns to float and imputing the missing values with mean
- Splitting the dataset into X and Y variables

```
X_que1 = dataset1.drop('satisfaction', axis=1)
y_que1 = dataset1['satisfaction']
```

Subpart 2:

- Using the train_test_split() function the dataset was split into training and testing sets
- Created an object of SFS by embedding the Decision Tree classifier object, providing 10 features, forward as True, floating as False, and scoring = accuracy.
- Train SFS on the dataset and printed the best features

```
Best Features:
('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Ease of Online booking', 'Gate location', 'Online boarding', 'Baggage handling', 'Checkin service', 'Inflight service')
```

- Calculated accuracy using a DecisionTreeClassifier for the obtained 10 best features and all the features

```
Accuracy on best 10 features selected from sfs is : 94.7392129733904%
Accuracy on using all features is : 94.56546220920879%
```

Subpart 3:

- Using the forward and Floating parameter toggle between SFS(forward True, floating False), SBS (forward False, floating False), SFFS (forward True, floating True), SBFS (forward False, floating True), and choosing cross-validation = 4 for each configuration.
- Printed the cv scores for each configuration.

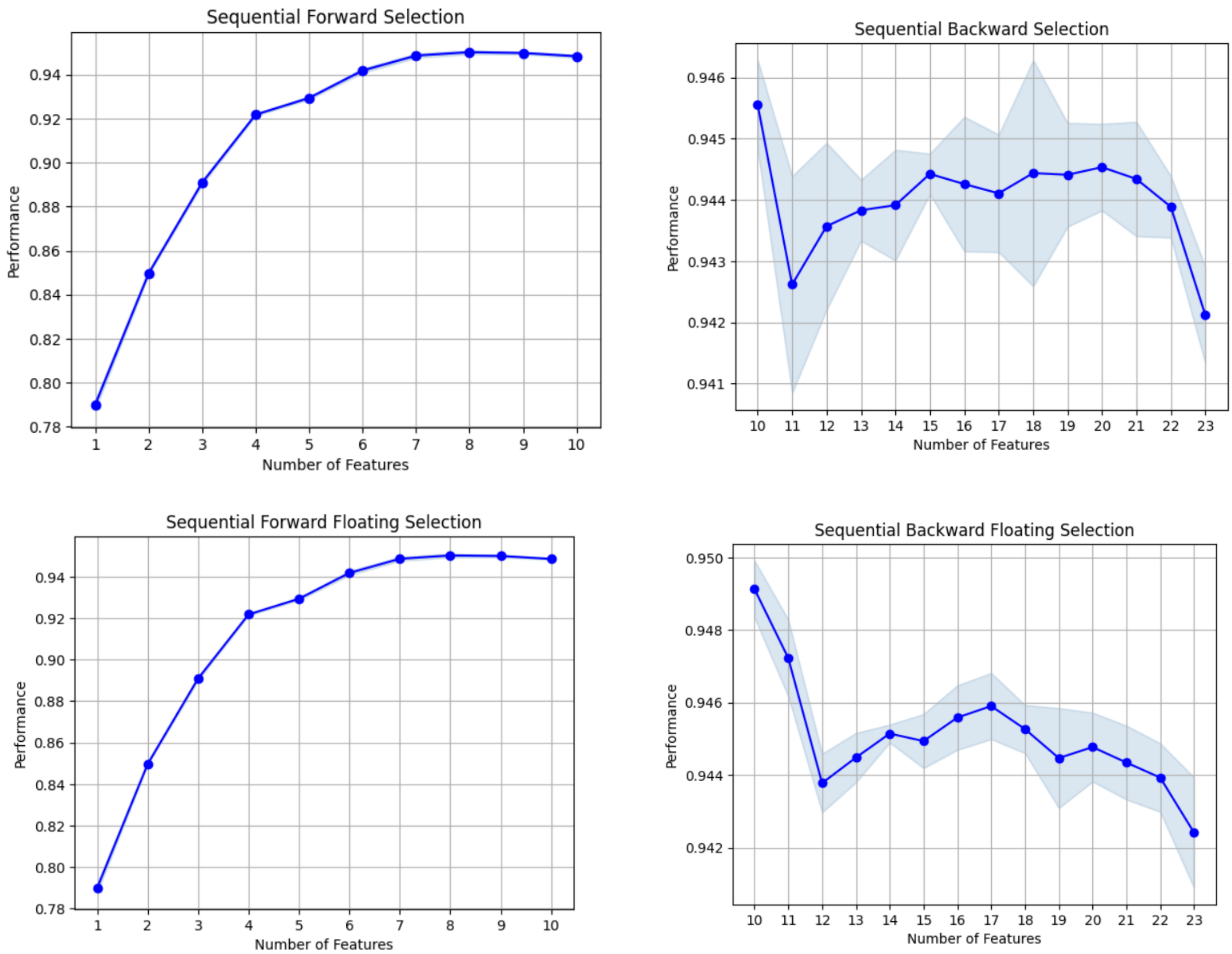
```
CV score for configuration SFS is: 0.9483417326423065
CV score for configuration SBS is: 0.9455560963762351
CV score for configuration SFFS is: 0.9486037415025601
CV score for configuration SBFS is: 0.9491277569409522
```

Subpart 4:

- Visualising the output from the feature selection in a pandas DataFrame format using the get_metric_dict for all four configurations.

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
SFS	[2, 4, 5, 7, 9, 10, 12, 17, 18, 19]	[0.9485906558552595, 0.9486458160957582, 0.947...	0.948342	[2, 4, 5, 7, 9, 10, 12, 17, 18, 19]	0.001238	0.000772	0.000446
SBS	[2, 4, 5, 7, 10, 13, 14, 17, 18, 19]	[0.9443433173368636, 0.9462739257543162, 0.945...	0.945556	[2, 4, 5, 7, 10, 13, 14, 17, 18, 19]	0.001167	0.000728	0.000420
SFFS	[2, 4, 5, 7, 9, 10, 12, 17, 18, 19]	[0.9489216172982514, 0.9488112968172541, 0.947...	0.948604	[2, 4, 5, 7, 9, 10, 12, 17, 18, 19]	0.000781	0.000487	0.000281
SBFS	[2, 4, 5, 7, 10, 12, 13, 14, 17, 19]	[0.9488112968172541, 0.9504661040322135, 0.948...	0.949128	[2, 4, 5, 7, 10, 12, 13, 14, 17, 19]	0.001265	0.000789	0.000455

- Finally, plotting the results for each configuration



Subpart 5 and 6:

- Bidirectional feature-set generational algorithm from scratch
- Using the selection criteria from the following:
    - Accuracy Measures: using Decision Tree and SVM Classifiers
    - Information Measures: Information gain
    - Distance Measure: Angular Separation, Euclidian Distance and City-Block Distance
    - Distance Measures. - Measures of separability, discrimination, or divergence measures. The most typical is derived from the distance between the class conditional density functions.)

- Selected features from accuracy measure and information measure selection criteria are

```
Selected features (accuracy measure): ['id', 'Customer Type', 'Type of Travel', 'Class', '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', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness', 'Arrival Delay in Minutes', 'Gender', 'Age', 'Leg room service', 'Flight Distance', 'Departure Delay in Minutes']

Selected features (information measure): ['id', 'Flight Distance', 'Gender', 'Customer Type', 'Age', 'Type of Travel', 'Class', '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']
```

Subpart 7:

- Training DecisionTreeClassifier on the Selected features generated from each measure and calculating its accuracy

```
Accuracy using Decision Tree and Random Forest: 0.961938909191202
Accuracy using Information Gain and Random Forest: 0.962297455698821
```

Question 2

Subpart 1:

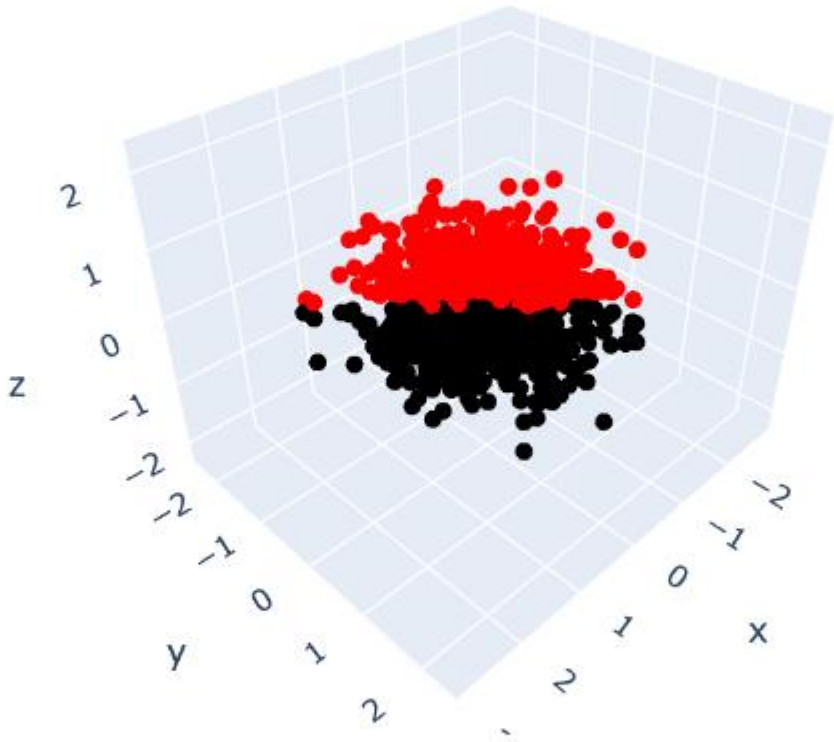
- Generating a dataset of 1000 points from a zero-centered Gaussian distribution with a covariance matrix

$$\sum = \begin{bmatrix} 0.6006771 & 0.14889879 & 0.244939 \\ 0.14889879 & 0.58982531 & 0.24154981 \\ 0.244939 & 0.24154981 & 0.48778655 \end{bmatrix}$$

- Labelling the points as shown below

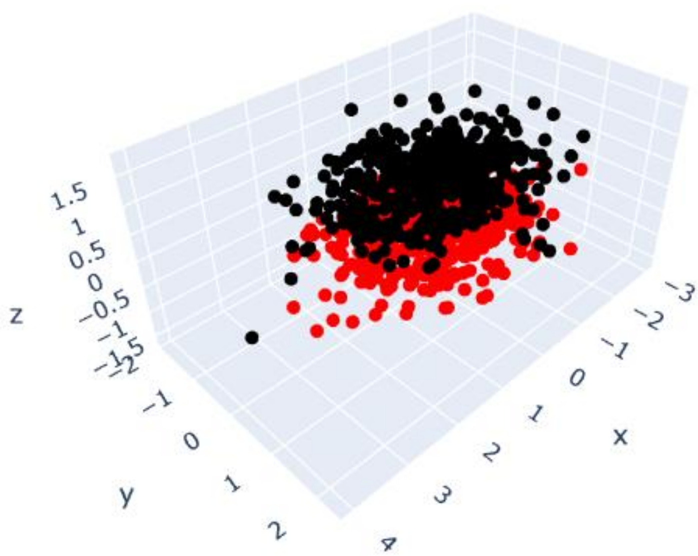
$$class = \begin{cases} 0 & \vec{x}.\vec{v} > 0 \\ 1 & \vec{x}.\vec{v} <= 0 \end{cases} where \vec{v} = \begin{bmatrix} 1/sqrt(6) \\ 1/sqrt(6) \\ -2/sqrt(6) \end{bmatrix}$$

- Creating s 3D scatter plot using Plotly's scatter3d function
- Resulting visualisation shows two distinct regions corresponding to the two labels



Subpart 2:

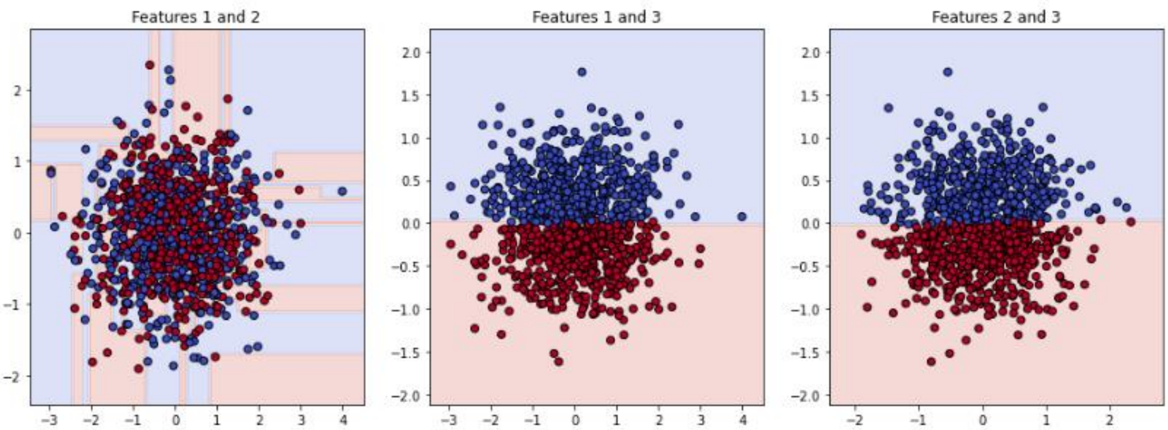
- Applying Principal component analysis with n_components=3 on the input data and transforming the data accordingly
- Plotting the 3D plot for the transformed data



- Resulting visualisation shows how the data has been transformed into a new coordinate system that captures the most important variation in the data

Subpart 3:

- Performing complete feature selection on the transformed data with a number of features in subset equal to two.
- Fit a Decision Tree for every subset-set of features of size 2 and plot their decision boundaries superimposed with the data.



Subpart 4:

- Selecting the subset of features obtained by applying PCA with n_components=2 and fit a decision tree
- Calculating the accuracy of this decision tree
- Now fitting a decision tree for every subset of features of size 2 and calculating their accuracies and plotting a bar graph of their accuracies

```
PCA Features: Accuracy = 0.4900
Features 1 and 2: Accuracy = 0.4750
Features 1 and 3: Accuracy = 0.9700
Features 2 and 3: Accuracy = 0.9800
```