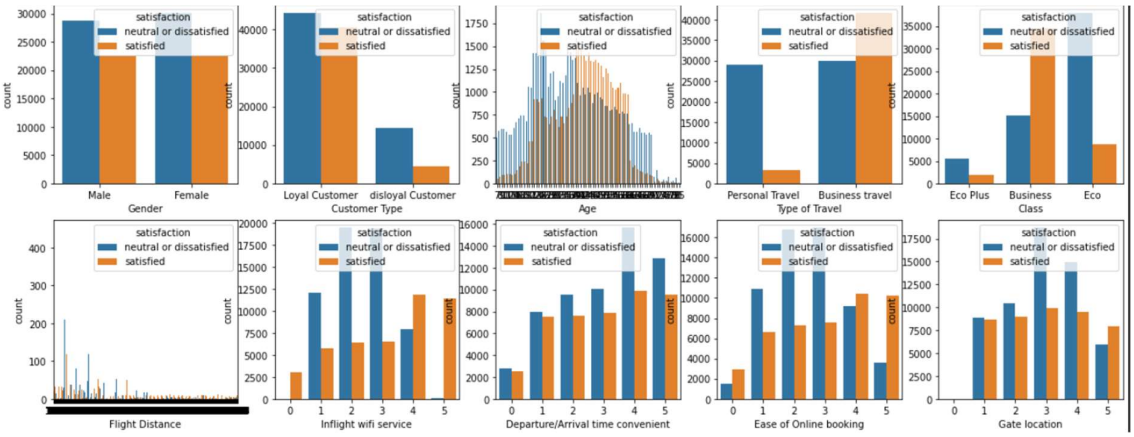
PRML LAB 8

Shalin Jain (B21CS070)

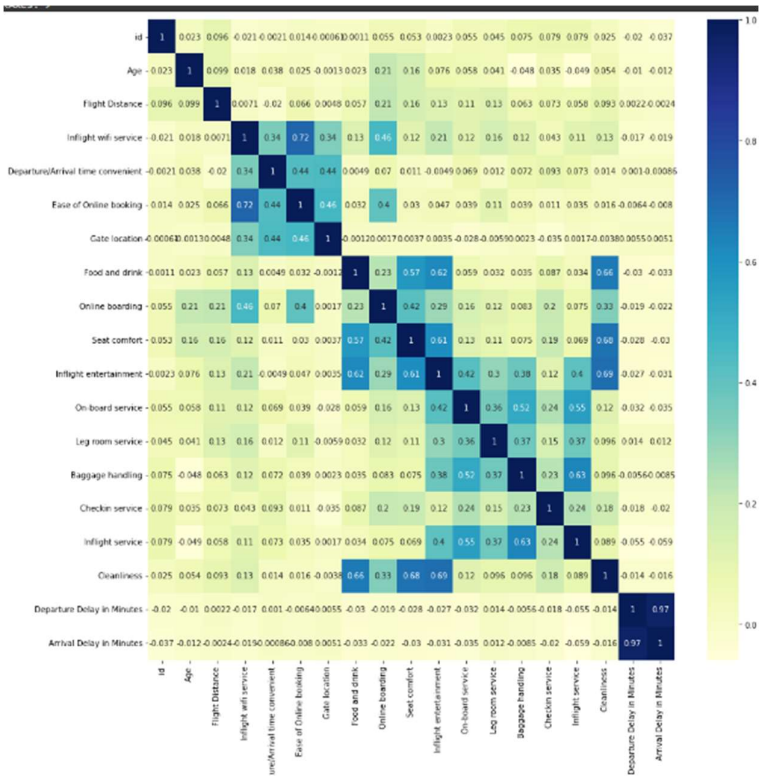
Problem 1

Part 1

In this part we have pre-processed the dataset as well as visualize the dataset for better understanding as shown below: -



The obtained heatmap is as follows: -



Part 2

In this part we have ran a sequential feature set generation SFS with 10 features, forward as True, floating as False and scoring = accuracy. Decision Tree Classifier is used for accuracy measurement. The obtained feature set is as follows: -

Accuracy with all 10 features: 0.9507237505348758
Best 10 features: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Gate location', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'Baggage handling', 'Inflight service')

Part 3

In this part we have used different combinations of floating and forward and obtained the feature set as follows: -

1) SFS

   Accuracy with all features: 0.9499441792423776
   Best 10 features: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Gate location', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'Baggage handling', 'Inflight service')

2) SBS

   Accuracy with all features: 0.9486834000615953
   Best 10 features: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Gate location', 'Online boarding', 'Seat comfort', 'On-board service', 'Baggage handling', 'Inflight service')

3) SFFS

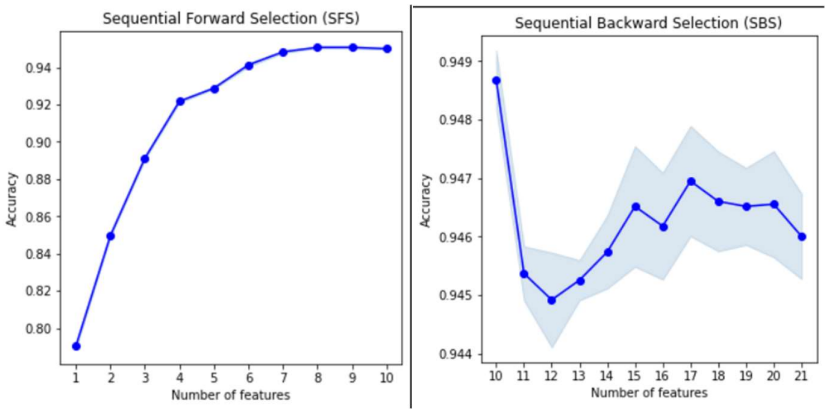
   Accuracy with all features: 0.9513396981829381
   Best 10 features: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'Baggage handling', 'Inflight service', 'Cleanliness')

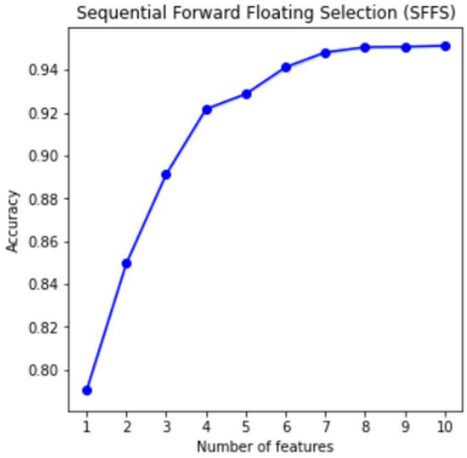
4) SBFS

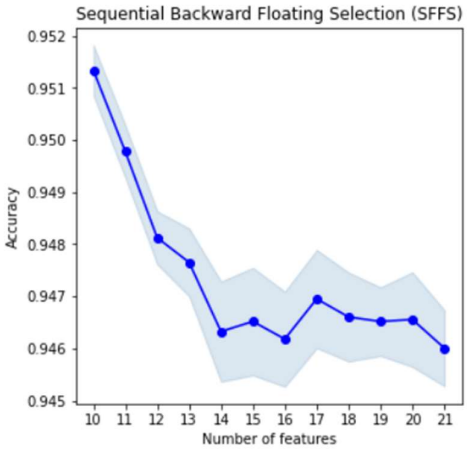
   Accuracy with all features: 0.9513396981829381
   Best 10 features: ('Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service', 'Online boarding', 'Seat comfort', 'Inflight entertainment', 'Baggage handling', 'Inflight service', 'Cleanliness')

Part 4

In this part we plotted accuracy vs number of features for all the above configs: -







Part 5

In this we have implemented bidirectional feature set generation technique. It is a combination of forward as well as backward elimination of features. The bidirectional function call forward selection to select the next feature and the backward function to eliminate the function it has a similarity function according to which feature are selected as well as removed.

Part 6

In this part we have implemented different similarity measures and generated the feature set for the respective similarity measure.

1) Decision Tree Accuracy

   Features: - {'Baggage handling', 'Class', 'Ease of Online booking', 'Inflight entertainment', 'Inflight wifi service', 'Leg room service', 'On-board service', 'Online boarding', 'Seat comfort', 'Type of Travel'}

2) SVM Accuracy

   Features: - {'Class', 'Ease of Online booking', 'Flight Distance', 'Inflight entertainment', 'Inflight wifi service', 'Leg room service', 'On-board service', 'Online boarding', 'Seat comfort', 'Type of Travel'}

3) Information Gain

   Features: - {'Age', 'Class', 'Ease of Online booking', 'Flight Distance', 'Inflight wifi service', 'Leg room service', 'On-board service', 'Online boarding', 'Seat comfort', 'Type of Travel'}

4) City Block Distance

   Features: - {'Baggage handling', 'Class', 'Cleanliness', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Inflight wifi service', 'Leg room service', 'On-board service', 'Online boarding', 'Type of Travel'}

5) Euclidean Distance

   Features: - {'Baggage handling', 'Class', 'Cleanliness', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Inflight wifi service', 'Leg room service', 'On-board service', 'Online boarding', 'Type of Travel'}

6) Angular Distance

   Features: - {'Age', 'Class', 'Customer Type', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Flight Distance', 'Food and drink', 'Gate location', 'Inflight wifi service', 'Type of Travel'}

Part 7

In this part we have used those feature set and used Decision Tree Classifier to obtain respective accuracy for different similarity measure.

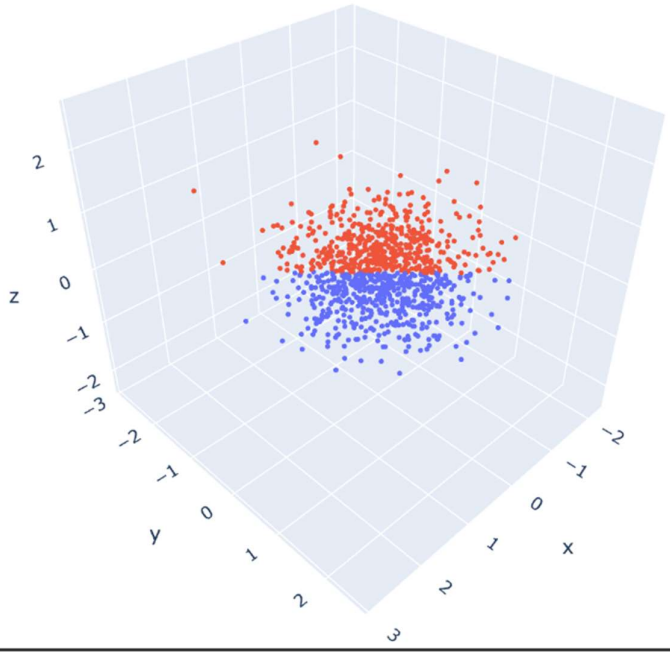
1) Decision Tree Accuracy: - 0.9352094298027753
2) SVM Accuracy: - 0.8719999999999999
3) Information Gain Accuracy: - 0.9160378847024649
4) City Block Distance Accuracy: - 0.78
5) Euclidean Accuracy: - 0.77
6) Angular Distance Accuracy: - 0.6799999999999999

Form the obtained accuracy we can say that the Decision Tree Accuracy helped to select the best features for the given dataset. While in distance measure the city-block and the Euclidean performed almost same.

Problem 2

Part 1

In this part we have made a dataset of 1000 samples using the covariance matrix and mean in question. We have used multivariate normal distribution to create the sample dataset. The 3D plot of the datapoints looks as shown below: -

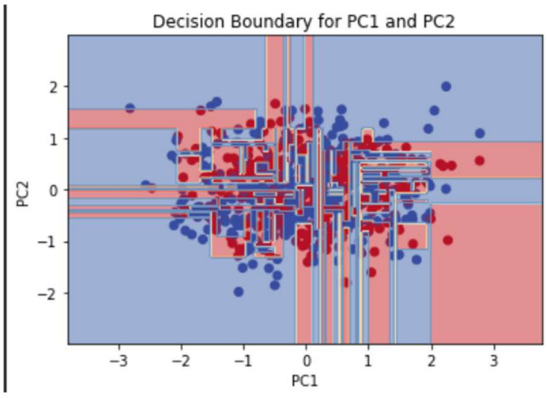


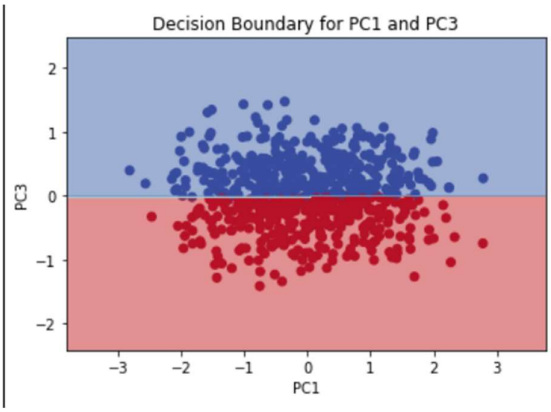
Part 2

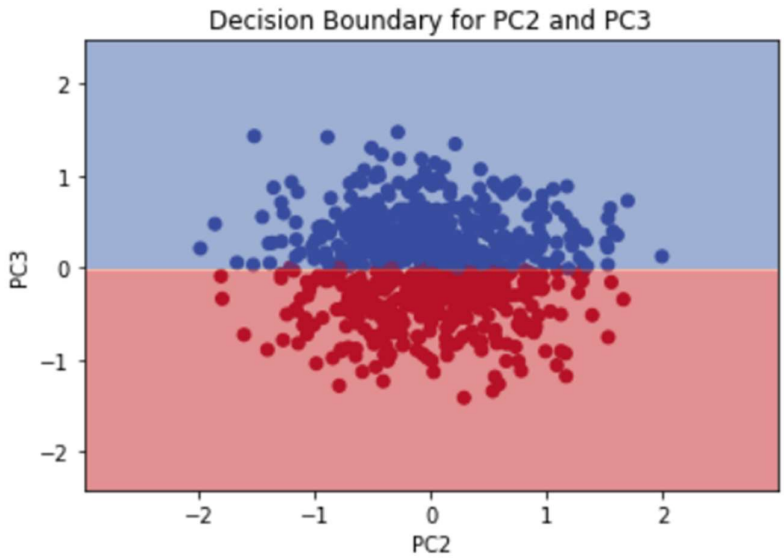
In this part we have applied PCA having 3 components and then transformed the dataset.

Part 3

In this part first we have split the dataset into testing and training datasets and then we have created sub dataset having only two features and plotted the Decision boundaries. The obtained graphs are as follows: -







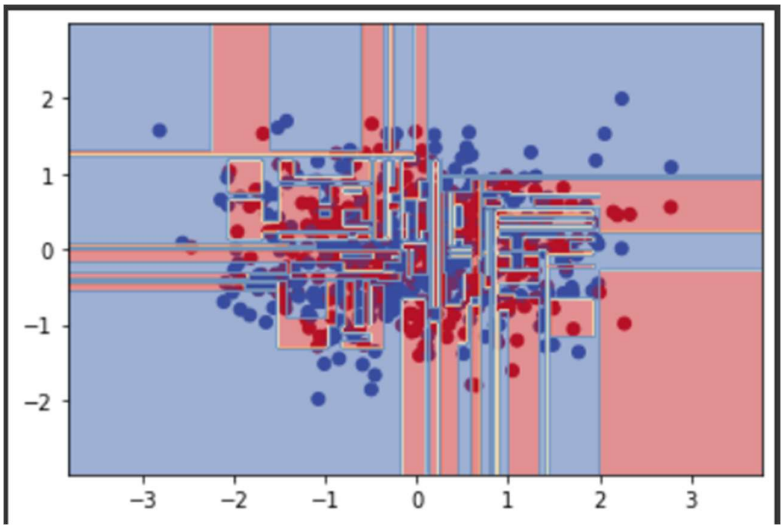
Part 4

In this part we have again used PCA and transformed the dataset having only 2 components and again plotted the decision boundaries. The obtained accuracies and decision boundary is as follows: -

```
Subset: (0, 1) Accuracy: 0.5
Subset: (0, 2) Accuracy: 0.9966666666666667
Subset: (1, 2) Accuracy: 0.99
Subset: (0, 1, 2) Accuracy: 0.9966666666666667
```

The 2 components accuracy

```
0.5033333333333333
```



The obtained decision boundary and accuracy are similar to the first case.

```
[ ]  1  for feature in feature_subsets:
     2    print(feature)
     3    dis = np.linalg.norm(X_train[:, feature] -X_train_2,ord = 'fro')
     4    print(dis,end = "\n")
```

```
(0, 1)
2.5574087207399166e-14
(0, 2)
22.273645629671492
(1, 2)
37.91751289476408
```

We can see that the distance between the same matrix components is very less (~0) for the case of feature (0,1) hence on applying n_componets = 2 and 3 we get similar accuracy for this particular case. And for the other cases we have high accuracy due to high distances.