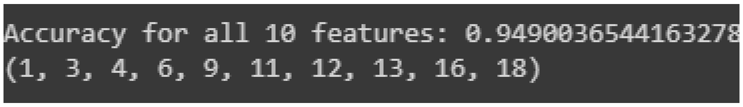
LAB Assignment 8
CSL 2050

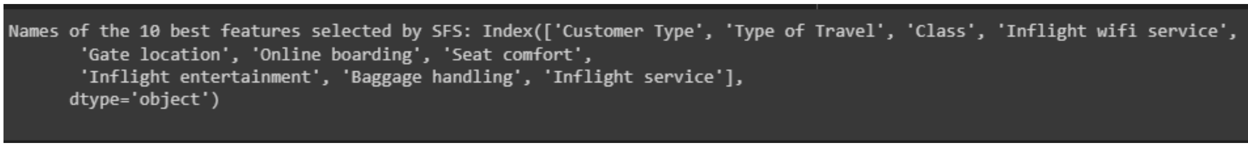
By :- Akshat Jain B21CS005

1. We have the Airline Passenger dataset, which we will use to classify whether someone was satisfied or not given attributes about a passenger.
    1.1. We preprocess the data by removing NaN values since their number is meagre compared to total data points. We then perform label encoding on categorical features and standard scaling on other features. We also plot a covarianv=ce matrix for the dataset. We then perform train test split and also separate features and labels.

    1.2. We then apply SFS with the given Parameters and model as a Decision tree. We then train the SFS on training data and find the names of the ten best features.

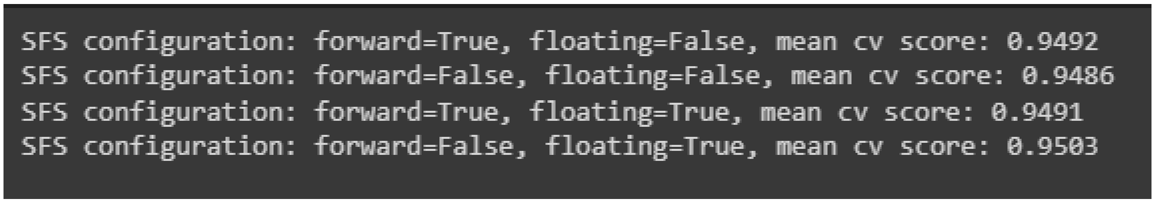


```
Accuracy for all 10 features: 0.9490036544163278
(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)
```



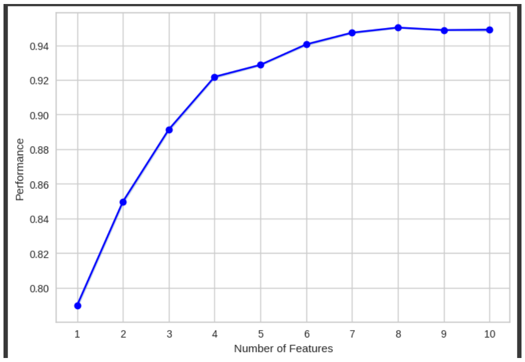
```
Names of the 10 best features selected by SFS: Index(['Customer Type', 'Type of Travel', 'Class', 'Inflight wifi service',
       'Gate location', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'Baggage handling', 'Inflight service'],
      dtype='object')
```

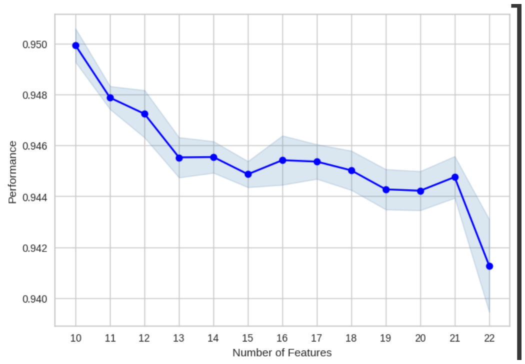
    1.3. We apply the four feature selectors and get the cross-validation scores for each configuration.

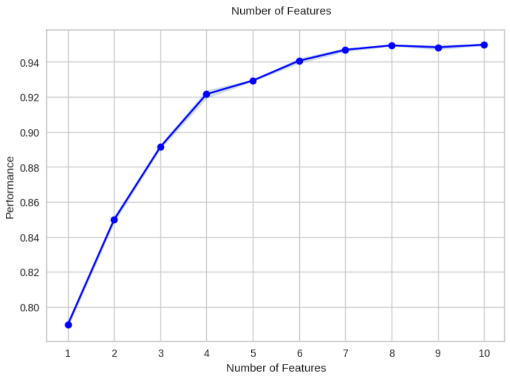


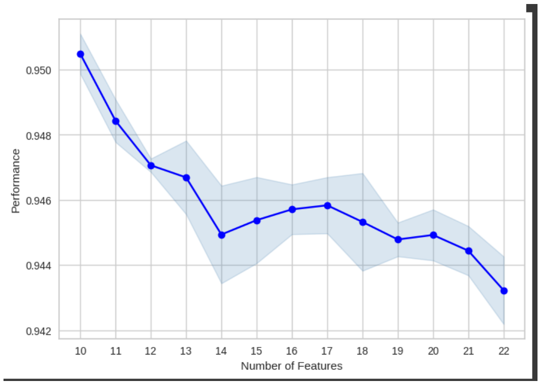
```
SFS configuration: forward=True, floating=False, mean cv score: 0.9492
SFS configuration: forward=False, floating=False, mean cv score: 0.9486
SFS configuration: forward=True, floating=True, mean cv score: 0.9491
SFS configuration: forward=False, floating=True, mean cv score: 0.9503
```

    1.4. We visualise the output from feature selection for all four configurations. We also plot the results for each configuration.







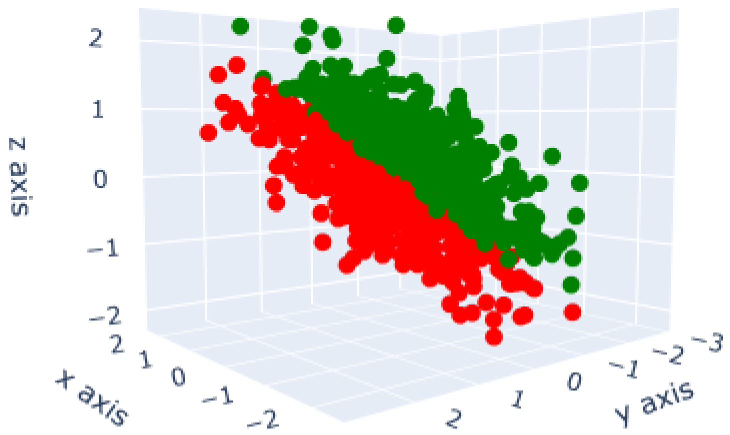


1.5. We apply bdfs from scratch.

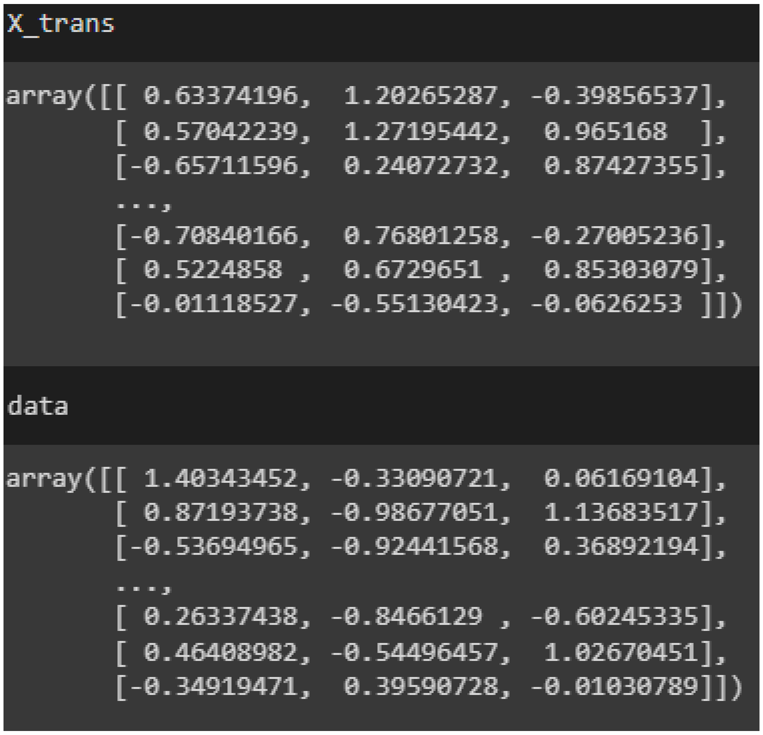
1.6. We also create a helper function for getting similarity measure and then using it in bdfs.

2. We want to compare PCA and pairwise complete feature selection in this question.

2.1. We create a dataset X of size 1000 sampled from a zero-centred Gaussian distribution with the given covariance matrix for this question. We then generate class labels for the points per the instructions. Afterwards, we visualise the data in a 3-dimensional scatter plot using the Plotly library of Python.

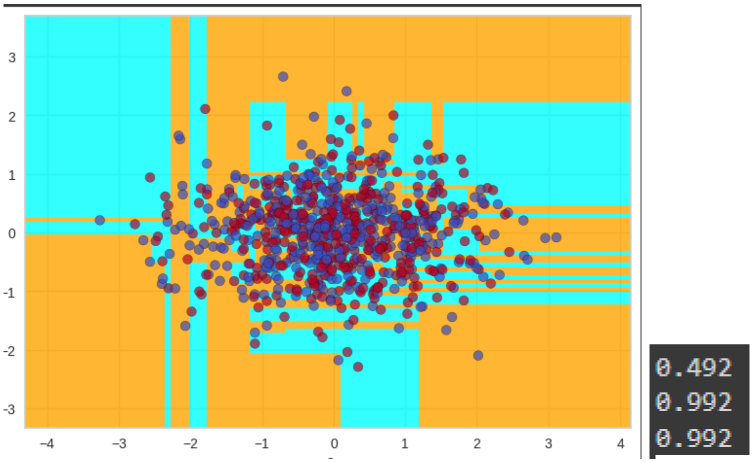


2.2. We apply PCA(n_components = 3) on X and transform the data.

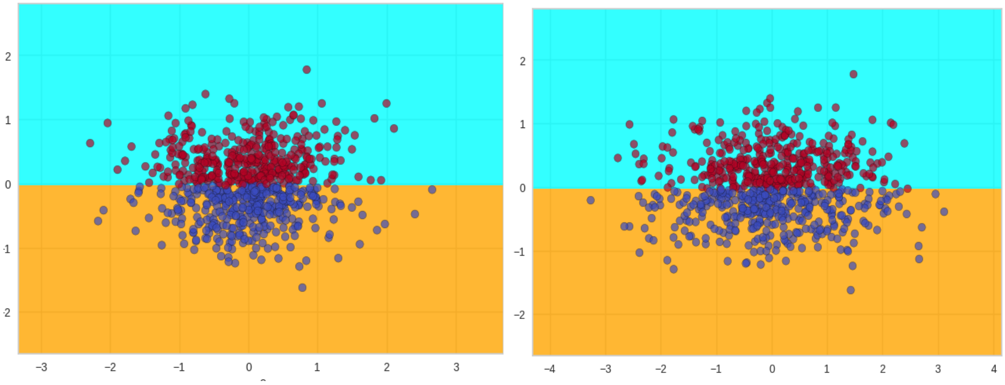


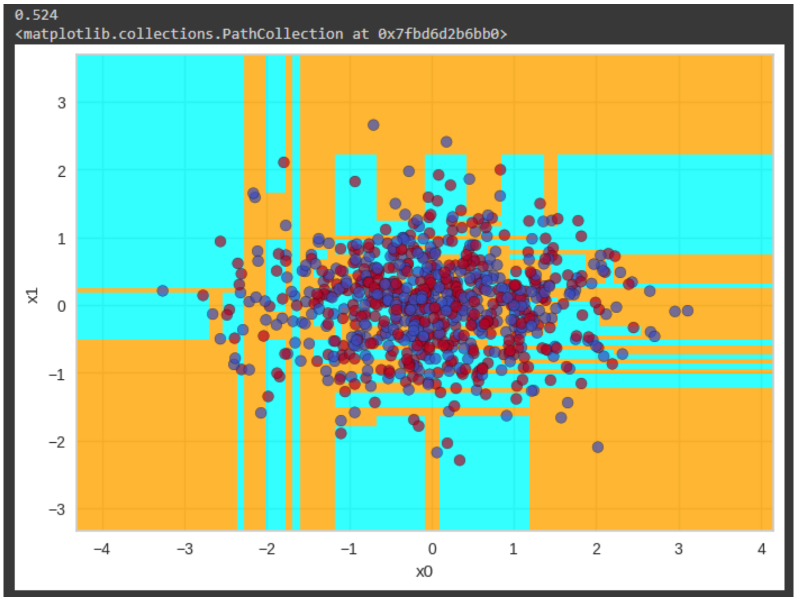
```
X_trans

array([[ 0.63374196,  1.20265287, -0.39856537],
       [ 0.57042239,  1.27195442,  0.965168  ],
       [-0.65711596,  0.24072732,  0.87427355],
       ...,
       [-0.70840166,  0.76801258, -0.27005236],
       [ 0.5224858 ,  0.6729651 ,  0.85303079],
       [-0.01118527, -0.55130423, -0.0626253 ]])

data

array([[ 1.40343452, -0.33090721,  0.06169104],
       [ 0.87193738, -0.98677051,  1.13683517],
       [-0.53694965, -0.92441568,  0.36892194],
       ...,
       [ 0.26337438, -0.8466129 , -0.60245335],
       [ 0.46408982, -0.54496457,  1.02670451],
       [-0.34919471,  0.39590728, -0.01030789]])
```

2.3. We then plot the 3 Subsets of features along with the decision boundary of a corresponding fitted decision tree. We also calculate the accuracy of the decision tree for the three subsets.



```
0.492
0.992
0.992
```





```
0.524
<matplotlib.collections.PathCollection at 0x7fbd6d2b6bb0>
```

2.4.

The vector v we have selected is a unit vector that divides our dataset along its normal plane. Since the variances of the x-axis column and the y-axis column is more than the z-axis column, when we put n_component = 2 in PCA, we see that it transforms the feature to be as same as the x-y axes dataset. The decision boundary and the accuracy for the above-stated 2 cases are very similar. Now the vector v that we have selected for applying classification is of the nature where when we take it along the X - Z axes or the Y- Z axes, we get a very high value of accuracy as compared to PCA(n_components =2) or the X-Y axes because of the value of its z component being higher than its x and y components. We can see this along the X and Y axes in our three-dimensional scatter plots. We can see that the imaginary plane dividing the two classes is very much visible when we see along the above two axes; however, when we see along the z-axis, we see that the distribution of these points becomes some sort of random distribution; thus we see that the accuracy for this case is about 50% as compared to more than 95% for the other 2 cases.