Pattern Recognition & Machine Learning
Lab-8 Assignment

Name: Tanish Pagaria
Roll No.: B21AI040

Question-1

Loading the data

The AirlinePassenger data set was downloaded and loaded as a Pandas dataframe.

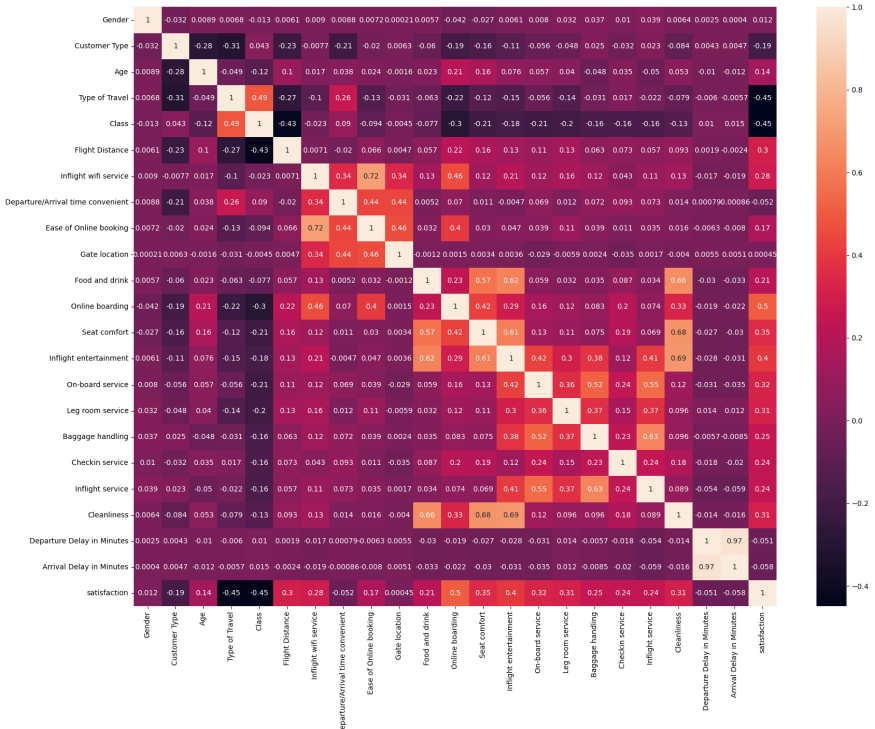
	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Che ser
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4	3	4	
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2	5	3	
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...	3	3	4	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
103899	103899	94171	Female	disloyal Customer	23	Business travel	Eco	192	2	1	...	2	3	1	4	
103900	103900	73097	Male	Loyal Customer	49	Business travel	Business	2347	4	4	...	5	5	5	5	
103901	103901	68825	Male	disloyal Customer	30	Business travel	Business	1995	1	1	...	4	3	2	4	
103902	103902	54173	Female	disloyal Customer	22	Business travel	Eco	1000	1	1	...	1	4	5	1	
103903	103903	62567	Male	Loyal Customer	27	Business travel	Business	1723	1	3	...	1	1	1	4	

103904 rows × 25 columns

AirlinePassengers Dataframe

Data analysis and preprocessing

The dataset did not contain any NULL values or NaN entries. The columns *"Unnamed: 0"* and *"id"* were dropped initially since they were irrelevant to the output class. The categorical features were then encoded. A correlation heatmap was plotted with the remaining columns.



Correlation Heatmap

(Since the features 'Departure Delay in minutes' and 'Arrival Delay in Minutes' were closely correlated, we could have decided to add and combine them into a new feature 'Total Delay' and drop the corresponding columns after adding the new one, but it was not performed since the lab itself was based on feature selection)

Histogram of the dataframe after encoding

The dataframe was also standardized using `sklearn.preprocessing.StandardScaler` because of the high variation in the ranges of certain features. The features and labels were separated as X and y respectively.

Selecting 10 best features using SFS by embedding the Decision Tree classifier

An SFS object was created using `mlxtend.feature_selection.SequentialFeatureSelector` by embedding the Decision Tree classifier object and providing 10 features: forward as True, floating as False, and scoring as "accuracy".

With the help of the `k_feature_names_` and `k_score_` attributes, the top 10 features and their accuracy were found.

```
('Customer Type',
 'Type of Travel',
 'Class',
 'Inflight wifi service',
 'Ease of Online booking',
 'Gate location',
 'Online boarding',
 'Seat comfort',
 'Baggage handling',
 'Inflight service')
```

10 best features

```
0.9506149056767821
```

Accuracy score for the features

Toggling between SFS, SBS, SFFS, and SBFS

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

Feature selection	Cross-validation scores
Sequential Forward Selection	0.9500839904382599
Sequential Backward Selection	0.9478734539163436
Sequential Forward Floating Selection	0.9512809635042744
Sequential Backward Floating Selection	0.9500839900655329

Output visualization as Pandas DataFrame and plotting the results for each configuration

The results from each feature selection were visualized as Pandas DataFrames using the `get_metric_dict` method.

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7897602224024094, 0.7918066334607514, 0.793...	0.790335	(Online boarding,)	0.004065	0.002536	0.001464
2	(3, 11)	[0.8483339125062743, 0.8511911656820726, 0.850...	0.849615	(Type of Travel, Online boarding)	0.002085	0.0013	0.000751
3	(3, 6, 11)	[0.8914629908490675, 0.8920421637901077, 0.892...	0.891249	(Type of Travel, Inflight wifi service, Online...	0.002214	0.001381	0.000797
4	(3, 6, 9, 11)	[0.9192246804895942, 0.9228927757828488, 0.922...	0.921733	(Type of Travel, Inflight wifi service, Gate l...	0.002346	0.001463	0.000845
5	(1, 3, 6, 9, 11)	[0.9277578284875864, 0.9284914475462374, 0.929...	0.928828	(Customer Type, Type of Travel, Inflight wifi ...	0.001202	0.00075	0.000433
6	(1, 3, 6, 9, 11, 16)	[0.9393412873083903, 0.9425074327194101, 0.939...	0.94128	(Customer Type, Type of Travel, Inflight wifi ...	0.002904	0.001811	0.001046
7	(1, 3, 4, 6, 9, 11, 16)	[0.946368585659678, 0.9486080543650334, 0.9485...	0.94824	(Customer Type, Type of Travel, Class, Infligh...	0.001827	0.00114	0.000658
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9496505656589057, 0.9504614077763362, 0.9508...	0.950644	(Customer Type, Type of Travel, Class, Infligh...	0.001154	0.00072	0.000416
9	(1, 3, 4, 6, 9, 11, 12, 16, 18)	[0.948646665894436, 0.9508861345997915, 0.9512...	0.950731	(Customer Type, Type of Travel, Class, Infligh...	0.002057	0.001283	0.000741
10	(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)	[0.9491486157766709, 0.9500366809529326, 0.949...	0.950084	(Customer Type, Type of Travel, Class, Infligh...	0.001311	0.000818	0.000472

SFS

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7897602224024094, 0.7918066334607514, 0.793...	0.790335	(Online boarding,)	0.004065	0.002536	0.001464
2	(3, 11)	[0.8483339125062743, 0.8511911656820726, 0.850...	0.849615	(Type of Travel, Online boarding)	0.002085	0.0013	0.000751
3	(3, 6, 11)	[0.8914629908490675, 0.8920421637901077, 0.892...	0.891249	(Type of Travel, Inflight wifi service, Online...	0.002214	0.001381	0.000797
4	(3, 6, 9, 11)	[0.9192246804895942, 0.9228927757828488, 0.922...	0.921733	(Type of Travel, Inflight wifi service, Gate l...	0.002346	0.001463	0.000845
5	(1, 3, 6, 9, 11)	[0.9277578284875864, 0.9284914475462374, 0.929...	0.928828	(Customer Type, Type of Travel, Inflight wifi ...	0.001202	0.00075	0.000433
6	(1, 3, 6, 9, 11, 16)	[0.9393412873083903, 0.9425074327194101, 0.939...	0.94128	(Customer Type, Type of Travel, Inflight wifi ...	0.002904	0.001811	0.001046
7	(1, 3, 4, 6, 9, 11, 16)	[0.946368585659678, 0.9486080543650334, 0.9485...	0.94824	(Customer Type, Type of Travel, Class, Infligh...	0.001827	0.00114	0.000658
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9496505656589057, 0.9504614077763362, 0.9508...	0.950644	(Customer Type, Type of Travel, Class, Infligh...	0.001154	0.00072	0.000416
9	(1, 3, 4, 6, 9, 11, 12, 16, 18)	[0.948646665894436, 0.9508861345997915, 0.9512...	0.950731	(Customer Type, Type of Travel, Class, Infligh...	0.002057	0.001283	0.000741
10	(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)	[0.9491486157766709, 0.9500366809529326, 0.949...	0.950084	(Customer Type, Type of Travel, Class, Infligh...	0.001311	0.000818	0.000472

SBS

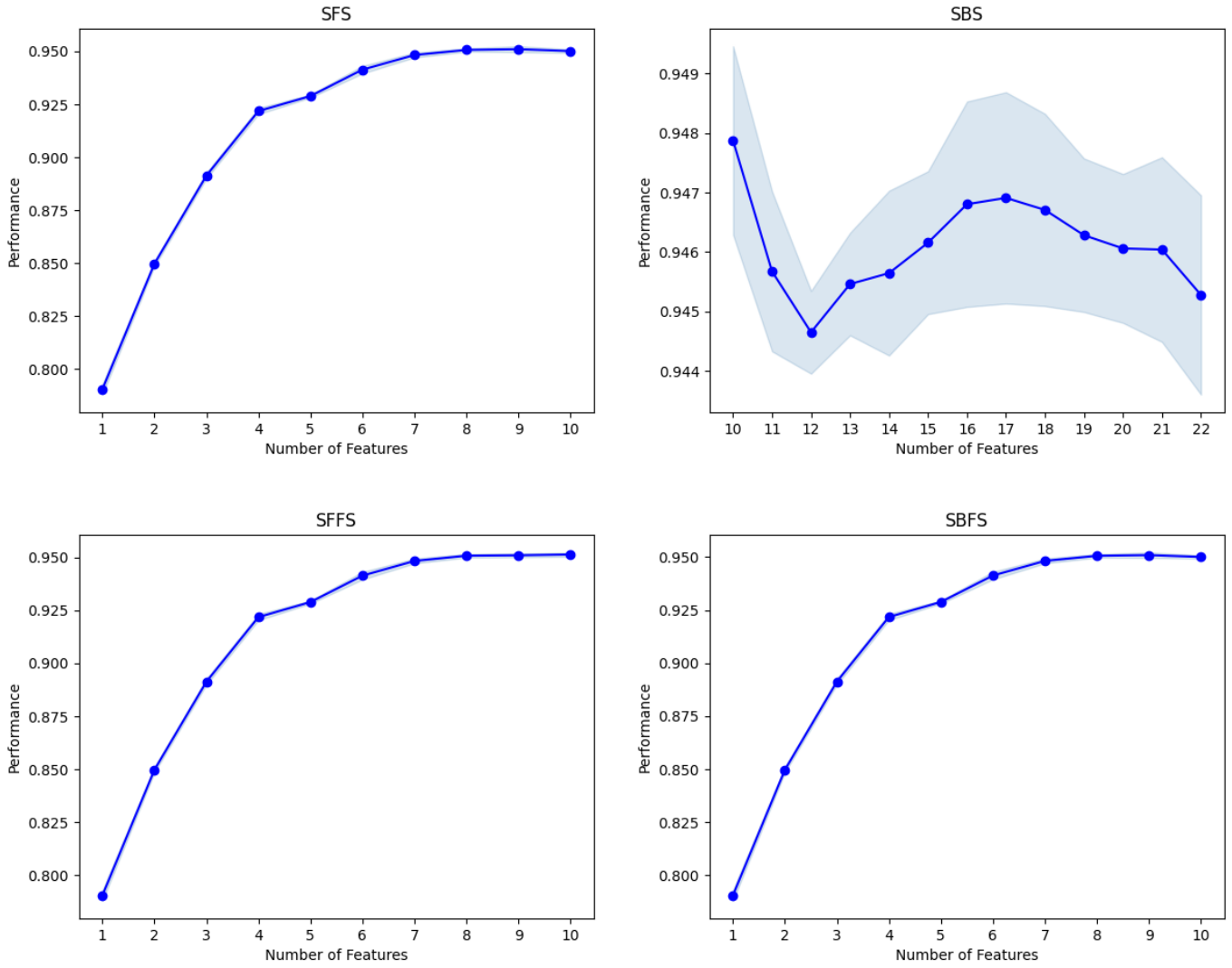
	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
1	(11,)	[0.7897602224024094, 0.7918066334607514, 0.793...	0.790335	(Online boarding,)	0.004065	0.002536	0.001464
2	(3, 11)	[0.8483339125062743, 0.8511911656820726, 0.850...	0.849615	(Type of Travel, Online boarding)	0.002085	0.0013	0.000751
3	(3, 6, 11)	[0.8914629908490675, 0.8920421637901077, 0.892...	0.891249	(Type of Travel, Inflight wifi service, Online...	0.002214	0.001381	0.000797
4	(3, 6, 9, 11)	[0.9192246804895942, 0.9228927757828488, 0.922...	0.921733	(Type of Travel, Inflight wifi service, Gate l...	0.002346	0.001463	0.000845
5	(1, 3, 6, 9, 11)	[0.9277578284875864, 0.9284914475462374, 0.929...	0.928828	(Customer Type, Type of Travel, Inflight wifi ...	0.001202	0.00075	0.000433
6	(1, 3, 6, 9, 11, 16)	[0.9393412873083903, 0.9425074327194101, 0.939...	0.94128	(Customer Type, Type of Travel, Inflight wifi ...	0.002904	0.001811	0.001046
7	(1, 3, 4, 6, 9, 11, 16)	[0.946368585659678, 0.9486080543650334, 0.9485...	0.94824	(Customer Type, Type of Travel, Class, Infligh...	0.001827	0.00114	0.000658
8	(1, 3, 4, 6, 9, 11, 16, 18)	[0.9496505656589057, 0.9504614077763362, 0.9508...	0.950644	(Customer Type, Type of Travel, Class, Infligh...	0.001154	0.00072	0.000416
9	(1, 3, 4, 6, 9, 11, 12, 16, 18)	[0.948646665894436, 0.9508861345997915, 0.9512...	0.950731	(Customer Type, Type of Travel, Class, Infligh...	0.002057	0.001283	0.000741
10	(1, 3, 4, 6, 9, 11, 12, 13, 16, 18)	[0.9491486157766709, 0.9500366809529326, 0.949...	0.950084	(Customer Type, Type of Travel, Class, Infligh...	0.001311	0.000818	0.000472

SFFS

SBFS

DataFrame visualization for each feature selection

The `mlxtend.plotting.plot_sequential_feature_selection` function was used to plot the results for each configuration.



Plots of the results for each configuration

Bi-directional Feature Set Generation Algorithm implementation from scratch

We created a class named `BidirectionalFeatureSetGeneration`. The constructor for the class took in the similarity measure to be used for the feature selection. The default similarity measure was set to accuracy score by Decision Tree Classifier. Also, the constructor took in an optional input regarding the estimator model the user would like to select in case of accuracy similarity measure. Its default value was set to "Decision Tree". The user could also select "SVM" as the model.

A `fit` method was defined that took in the dataset and labels to be used for feature set generation. The method would create two measure variables for the forward selection and the backward selection, to be used in comparison with the best or worst measure value at any point in the selection process in order to decide on halting or continuing the process.

For similarity measures including **accuracy** and **information gain**, the criterion was that the current best measure value should be greater than the previous measure value in order to continue.

And it was the opposite for the distance measures, including **angular separation**, **Euclidean distance,** and **city-block distance**. The `fit` method would then call the `algo` method to carry out the feature set generation algorithm.

The `algo` method made use of the `find_next` and `get_next` helper methods to find the next feature to add or remove, respectively.

- `find_next` method:
  It would use the similarity measure parameter to decide the next best feature to add to the set. It would also update the current best value of the measure for the forward selection accordingly.

- `get_next` method:
  Similar to the `get_next` method, it would return the next feature to be removed and update the measure value for the backward selection.

The algorithm would stop when the criterion would break for either the forward or backward selection or if all the bad features would get removed. The returned features would depend on the breaking condition. If the breaking condition occurs for only forward selection, then only the best features selected would get returned, or else the union of these features with the remaining feature would get returned, also taking into consideration the case when the breaking condition occurs for both forward and backward selection, where the one with the better measure value would get returned.
Other methods were defined for the calculations of the similarity measures. Another method was defined to return the classification results for the selected feature set of the dataset using a classifier based on the user's choice.

The following feature sets and classification results were obtained for all the similarity measures (using Decision Tree Classifier as the classifier model):-

```
similarity measure: accuracy
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
final feature set:-
['Online boarding', 'Type of Travel', 'Inflight wifi service', 'Gate location', 'Customer Type', 'Age', 'Class', 'De
parture/Arrival time convenient', 'Ease of Online booking', 'Food and drink', 'Seat comfort', 'Inflight entertainmen
t', 'On-board service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness', 'Departure Delay i
n Minutes', 'Arrival Delay in Minutes']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
no. of features: 19
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
accuracy score: 0.9495347310706977

####################################################################################################

similarity measure: information gain
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
final feature set:-
['Online boarding', 'Type of Travel', 'Inflight wifi service', 'Gender', 'Customer Type', 'Age', 'Class', 'Departure
/Arrival time convenient', 'Ease of Online booking', 'Gate location', 'Seat comfort', 'Inflight entertainment', 'On-
board service', 'Leg room service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness', 'Depar
ture Delay in Minutes', 'Arrival Delay in Minutes']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
no. of features: 20
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
accuracy score: 0.9477972122475772

####################################################################################################

similarity measure: angular separation
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
final feature set:-
['Arrival Delay in Minutes']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
no. of features: 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
accuracy score: 0.5652341789258273

####################################################################################################

similarity measure: euclidean distance
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
final feature set:-
['Online boarding', 'Type of Travel', 'Inflight wifi service', 'Gender', 'Customer Type', 'Age', 'Class', 'Departure
/Arrival time convenient', 'Ease of Online booking', 'Gate location', 'Seat comfort', 'Inflight entertainment', 'On-
board service', 'Leg room service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cleanliness', 'Depar
ture Delay in Minutes', 'Arrival Delay in Minutes']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
no. of features: 20
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
accuracy score: 0.9476813776593691

####################################################################################################
```

```
similarity measure: city-block distance
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
final feature set:-
['Online boarding', 'Type of Travel', 'Gender', 'Customer Type', 'Age', 'Class', 'Flight Distance', 'Inflight wifi s
ervice', 'Departure/Arrival time convenient', 'Ease of Online booking', 'Gate location', 'Food and drink', 'Seat com
fort', 'Inflight entertainment', 'On-board service', 'Baggage handling', 'Checkin service', 'Inflight service', 'Cle
anliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
no. of features: 21
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
accuracy score: 0.9471794277771343

####################################################################################################
```

From the results, we observed that most of the features got selected for this dataset based on our stopping criteria. The accuracy scores obtained were high for most of the similarity measures, and the number of features selected is also very high. The scores ranged close to 0.95. However, in the case of angular separation, the accuracy score was very low.
The possible reason for the bad feature selection in this case may be because angular separation refers to the angle between two vectors, which may not be relevant here in judging the loss between predicted labels and true labels.

Question-2

Dataset creation and visualization

A dataset of 1000 points sampled from a zero-centered Gaussian distribution has the covariance matrix shown below:

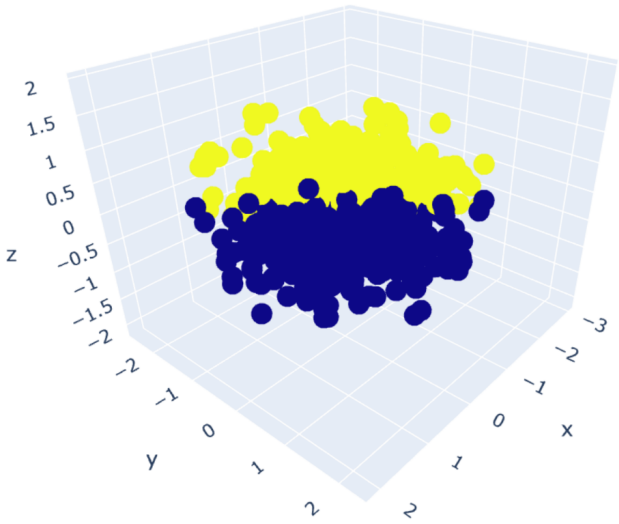
$$\sum = \begin{bmatrix} 0.6006771 & 0.14889879 & 0.244939 \\ 0.14889879 & 0.58982531 & 0.24154981 \\ 0.244939 & 0.24154981 & 0.48778655 \end{bmatrix}$$

The points were labeled based on:

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

(where x is the data point)

This was performed using the numpy.random.multivariate_normal function and taking the dot product of the data points with the given vector.
The `plotly.express.scatter_3d function` was used to visualize the data as a 3D scatter plot.



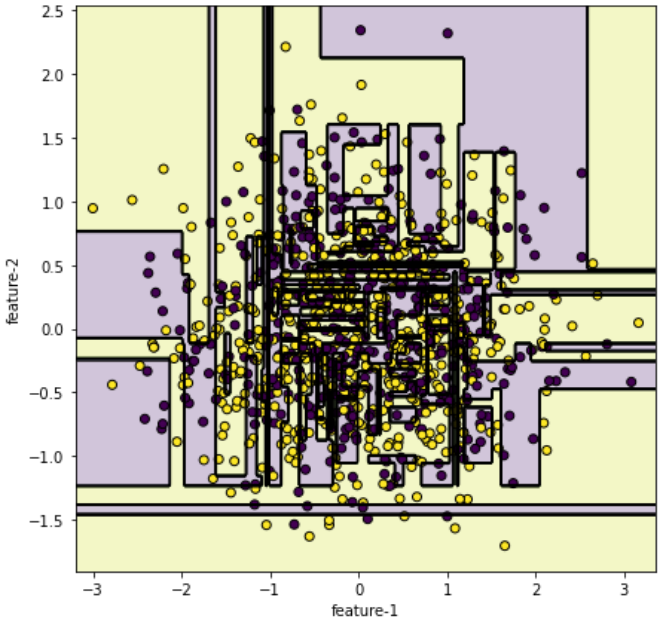
3D scatter-plot of the data

Principal Component Analysis (n_components=3) on the data

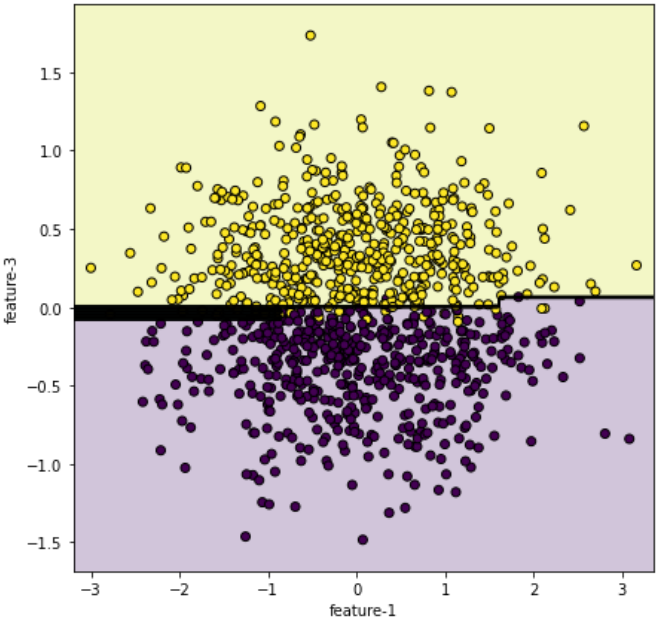
The data was transformed using `sklearn.decomposition.PCA`, taking `n_components` as 3.

Complete FS on the transformed data

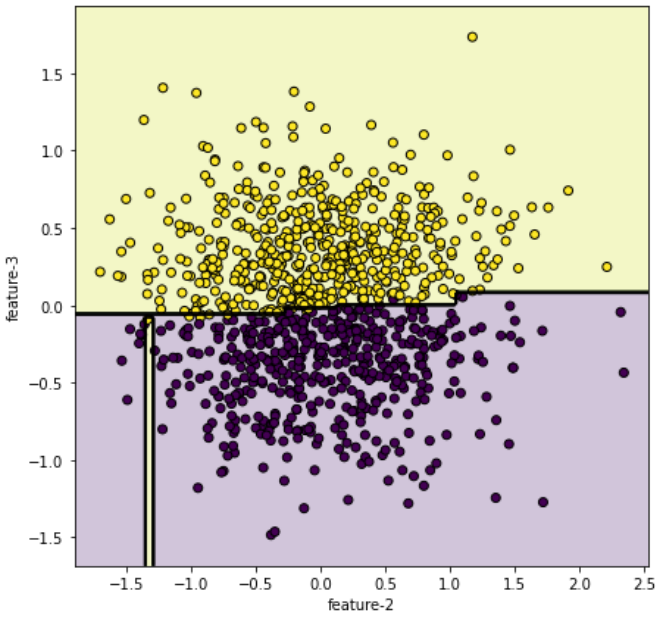
Complete feature selection was performed on the transformed data, with the number of features in each subset being 2. Separate decision tree classifiers were made for each of the three possible subsets, and decision boundaries were plotted superimposed with the data for each of the three classifiers.
Accuracy scores were also reported simultaneously by performing train-test splits after fitting the data into the models.



Accuracy score: 0.488



Accuracy score: 0.968

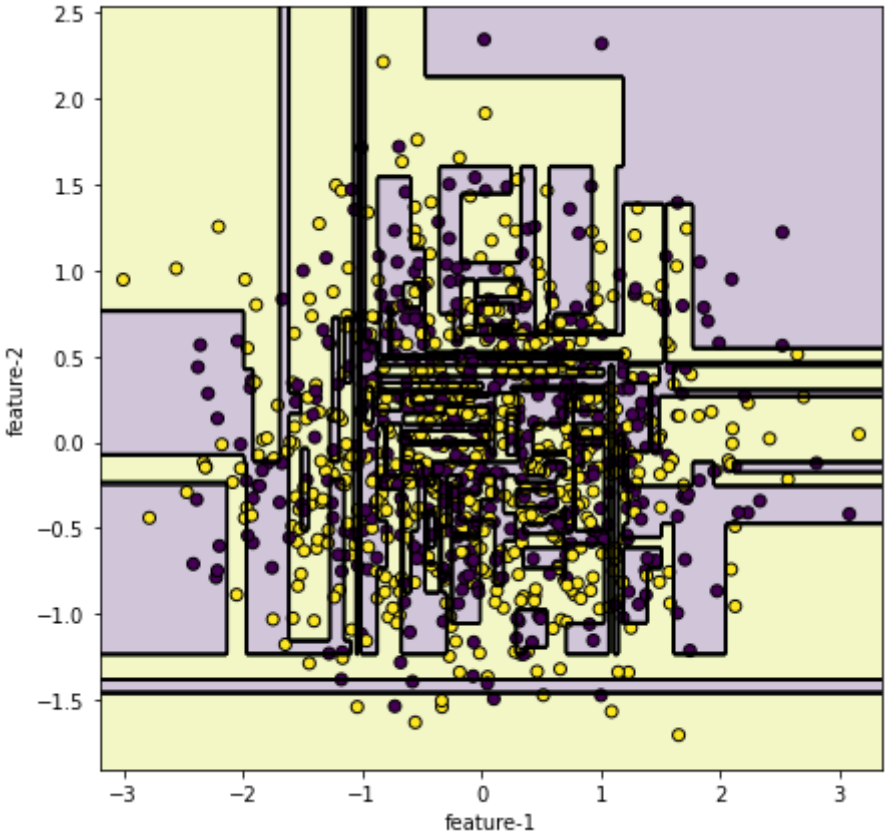


Accuracy score: 0.976

Decision boundary plots and accuracy scores for each feature subset

Principal Component Analysis (n_components=2) on the data

The data was transformed using `sklearn.decomposition.PCA`, taking `n_components` as 2. Similar operations were performed on this transformed data as were done for `n_components` = 3.



Accuracy score: 0.492

Decision boundary plot and the accuracy score obtained

The features 'feature-1' and 'feature-2' in the PCA(n_components=3) represent the features obtained by applying PCA(n_components=2). They also show almost the same accuracy scores. The decision boundary plots are also similar in both of them. These scores were very low as compared to the other two subsets.

Experiments to show the difference in the accuracies

- **Evaluation Metrics:**

The difference between the performance of the subset with features "feature-1" and "feature-2" was clearly greater than the other two subsets and almost equal to the newly transformed dataset.
The accuracy scores for them were close to 0.49, in contrast to accuracy scores of around 0.97 for the other subsets.

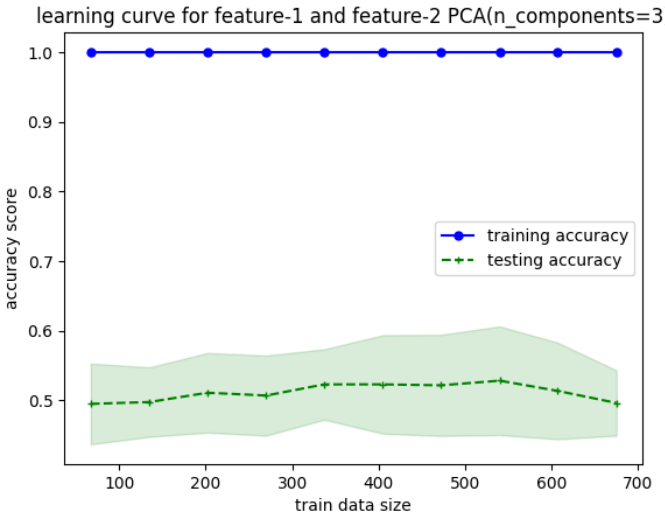
- **Variance:**

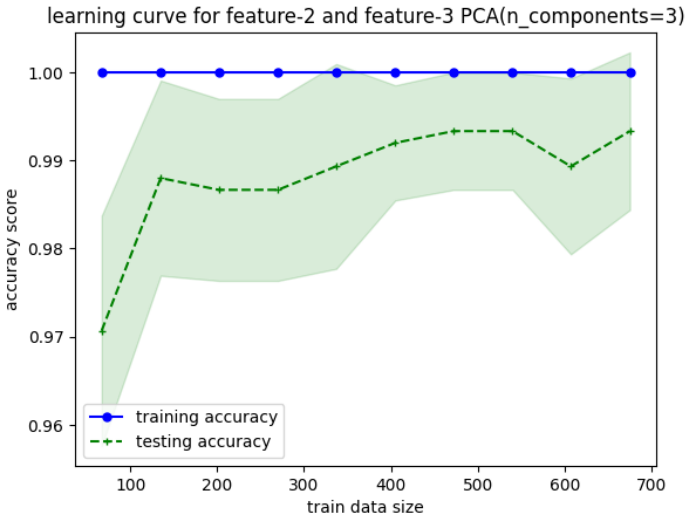
The features ‘feature-1’ and ‘feature-2’ show the highest variances among the three features for PCA(n_components=3). The same is true for the features of the PCA(n_components=2) transformed dataset. Hence, the difference between this subset and the others is clearly evident.

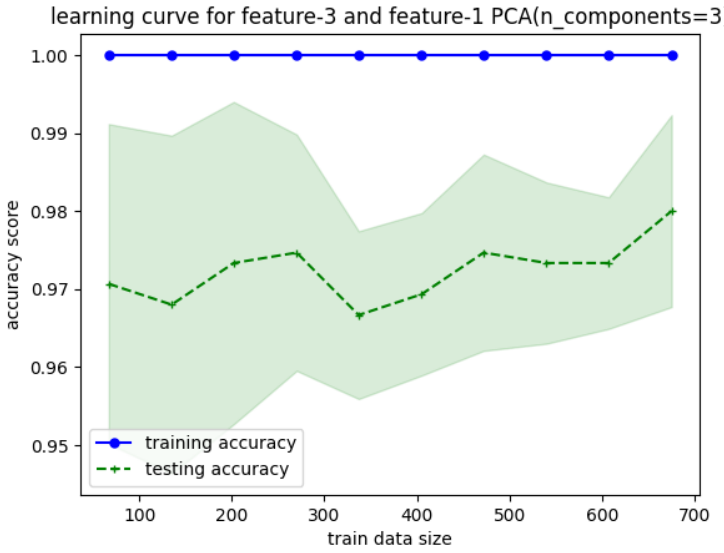
- **Learning Rate:**

The learning curve shows the relationship between the performance of the model and the size of the training set. We plotted the learning curves for the models trained on the subsets with PCA(n_components=3) and the newly transformed data with PCA(n_components=2) to analyze the bias-variance trade-off of the models and compare their performance.

for PCA(n_components=3)







for PCA(n_components=2)