Pattern Recognition and Machine Learning

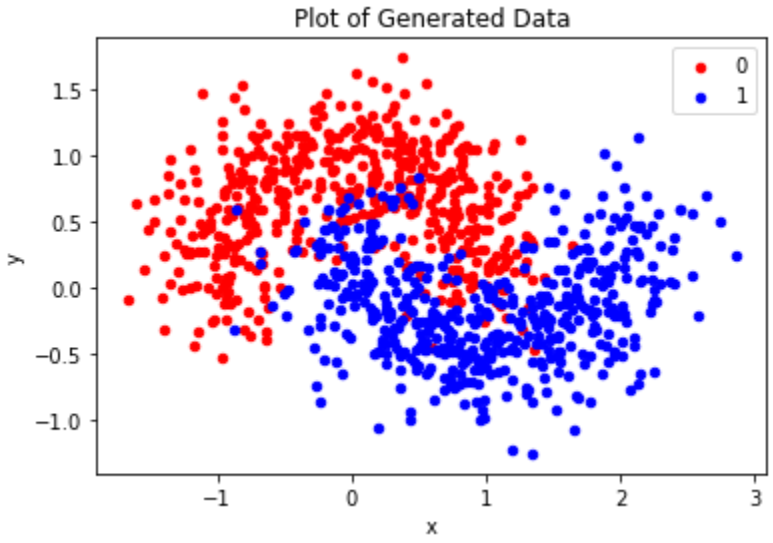
Lab - 5 Assignment Report

Aryan Himmatlal Prajapati (B21EE012)

Question 1.

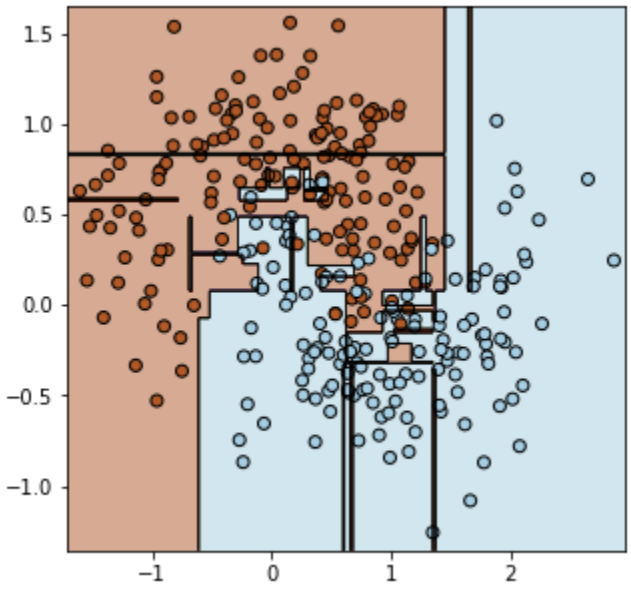
Part 1

Created a dataset with 1000 samples, using the ‘make_moon' function of sklearn with random_state=42 and noise=0.3. Performed appropriate preprocessing, train and test split of the dataset. Plotted the generated dataset.

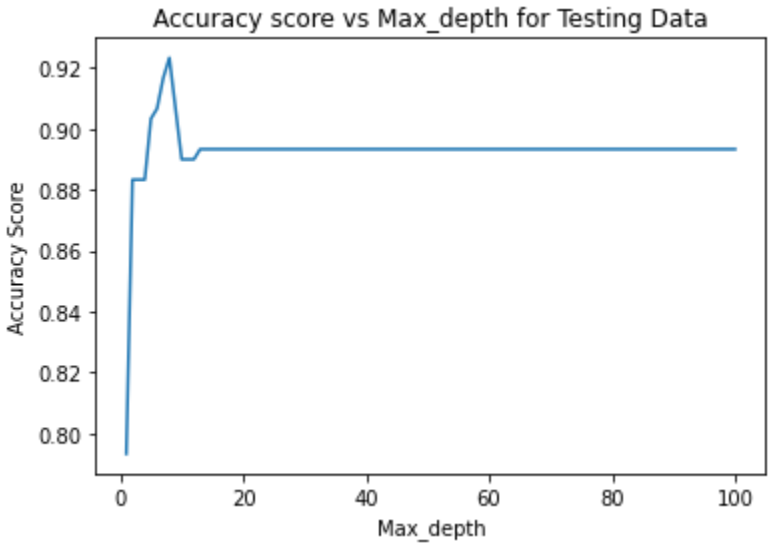


Trained a simple decision tree classifier from sklearn and plotted the decision boundary for the same.

Obtained Decision Boundary:



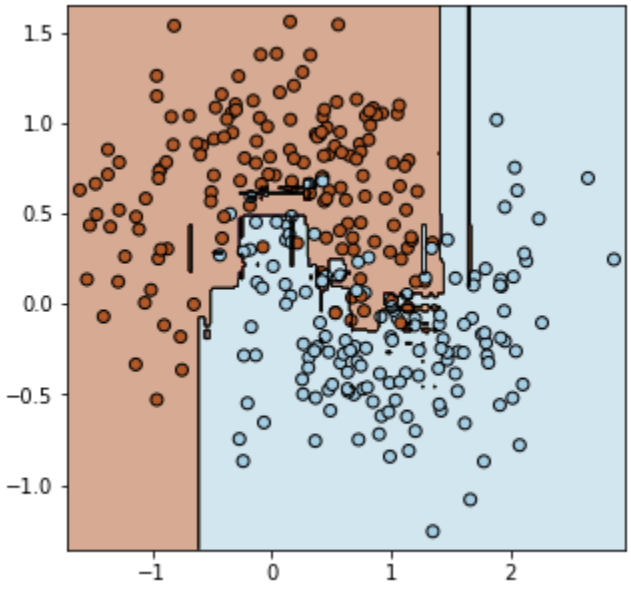
Performed hyperparameter tuning for finding the best value of max_depth of the decision tree.




```
Accuracy score is maximum, when max_depth = 8.0
```

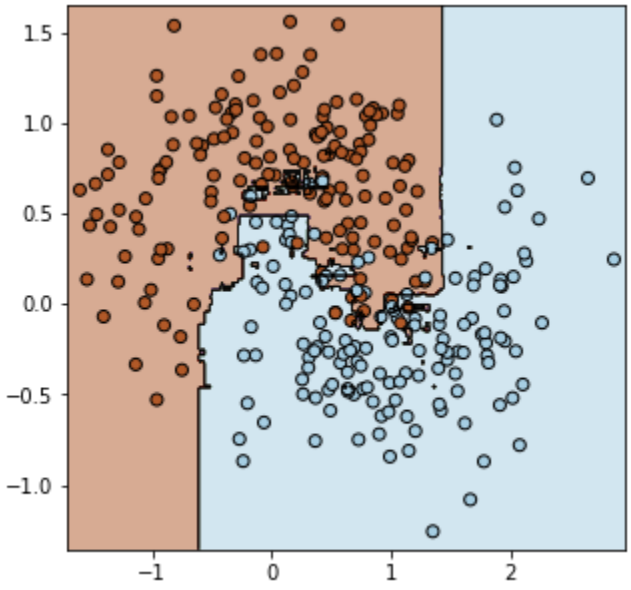
Trained a BaggingClassifier from sklearn, on the same dataset, and plotted the decision boundary obtained.

Obtained Decision Boundary:



Trained a RandomForest classifier from sklearn and plotted its decision boundary.

Obtained Decision Boundary:

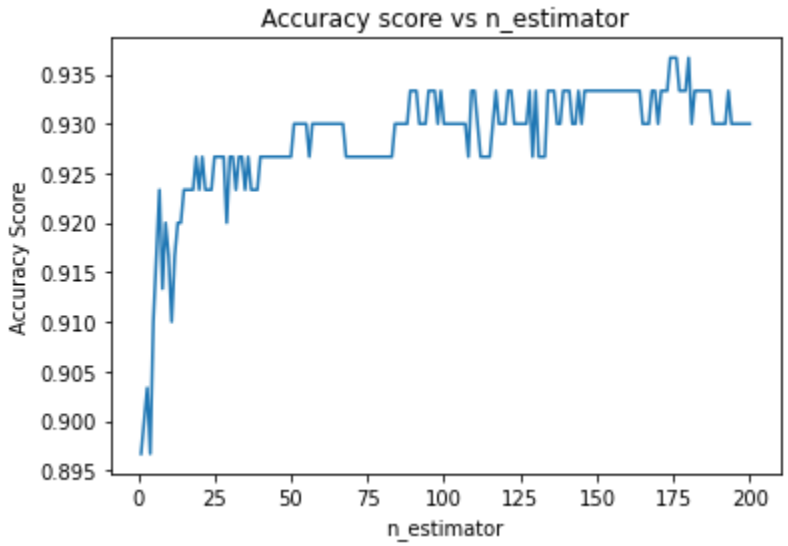


Accuracies:

Accuracy score for Decision Tree Classifier : `0.8933333333333333`
Accuracy score for Bagging Classifier: `0.9166666666666666`
Accuracy score for Random Forest Classifier: `0.9133333333333333`

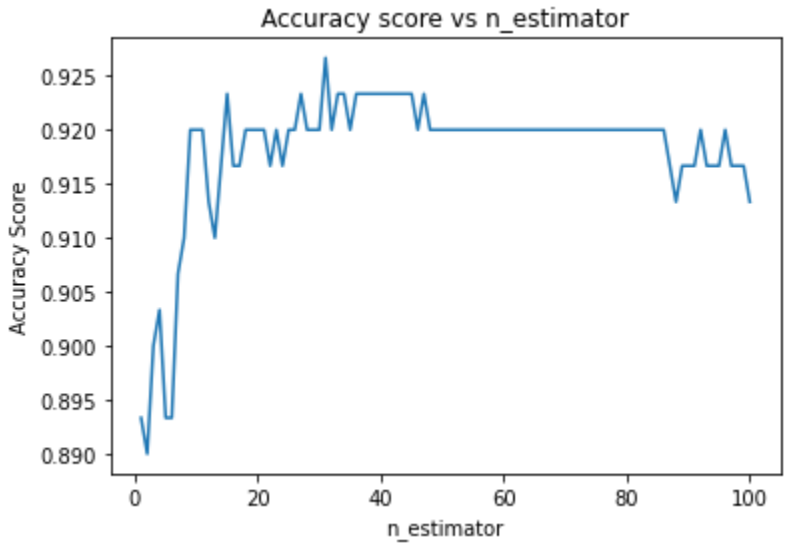
Varied the number of estimators for the BaggingClassifier and RandomForestClassifier.
Plotted Accuracy Score vs n_estimators.

Bagging Classifier:



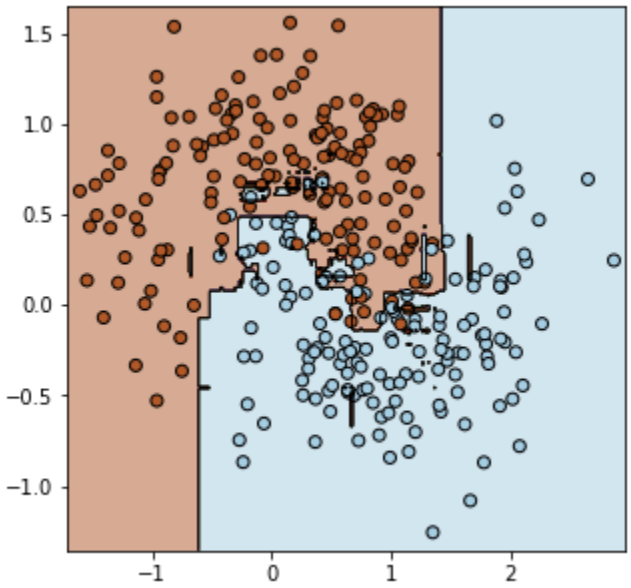
`Accuracy score is maximum, when n_estimators = 174.0`

Random Forest Classifier:

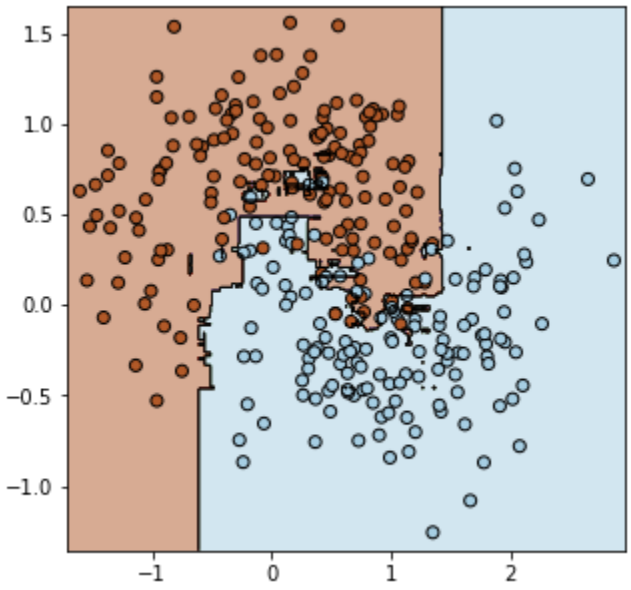


`Accuracy score is maximum, when n_estimators = 31.0`

Obtained Decision Boundary for Bagging classifier with Optimum n_estimators = 174:



Obtained Decision Boundary for RandomForest classifier with Optimum n_estimators = 31.0 :



As n_estimators increase, overfitting is decreasing and accuracy scores are increasing upto certain value of n_estimators.

Part 2

Implemented a Bagging algorithm from scratch. Applied the scratch bagging algorithm with n_estimators = 10 and trained it on the same dataset.

```
Accuracy Score: 0.93
```

Accuracy score for bagging classifiers is better than individual classifiers.

On Average Bagging classifiers work better than individual week learners.

Question 2.

Installed XGBoost and LightGBM.
Using the same dataset as in question 1

Part 1

Trained a AdaBoost Model.

Part 2

Train a XGBoost Model in which subsample=0.7.

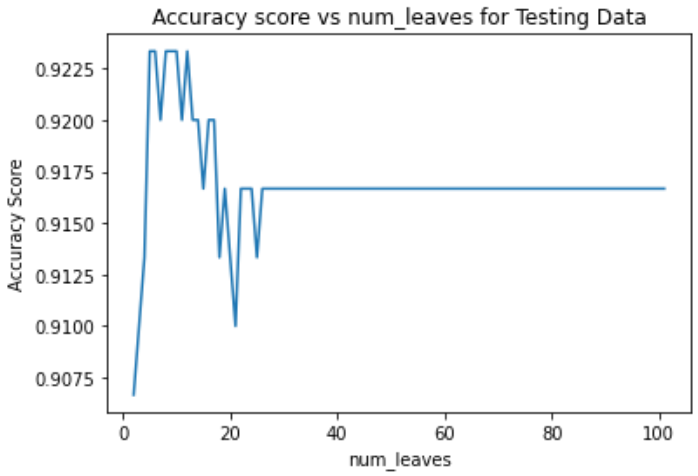
Part 3

```
For AdaBoost Model:-
Training Data Accuracy: 0.9485714285714286
Testing Data Accuracy: 0.9233333333333333

For XGBoost Model:-
Training Data Accuracy: 0.9957142857142857
Testing Data Accuracy: 0.9166666666666666
```

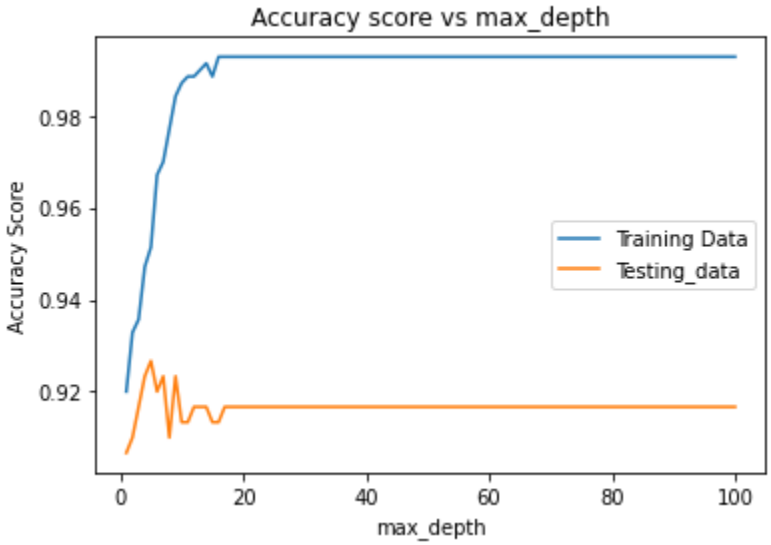
Part 4

Trained a LightGBM model and chose different values for num_leaves. Plotted Accuracy Score vs num_leaves.



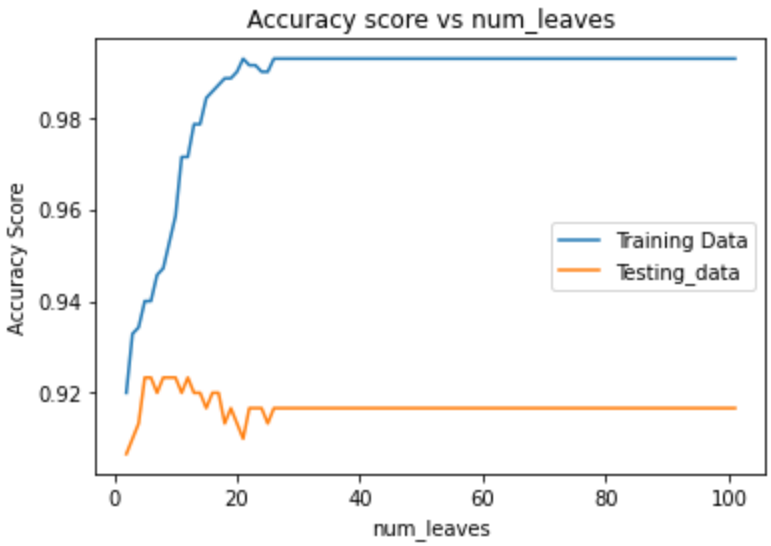
Part 5

Analyzed the relation between max_depth and num_leaves, and checked for which value the model starts overfitting.




```
Accuracy score is maximum for testing data, when max_depth = 5.0
```

After Max_depth = 5 when Accuracy score for Testing data is maximum, Testing Accuracy Score Starts decreasing while Training Accuracy score is Still Increasing. So After max_depth = 5 the model starts overfitting.




```
Accuracy score is maximum for testing data, when num_leaves = 5.0
```

After num_leaves = 5 when Accuracy score for Testing data is maximum, Testing Accuracy Score Starts decreasing while Training Accuracy score is Still Increasing. So After num_leaves = 5 the model starts overfitting.

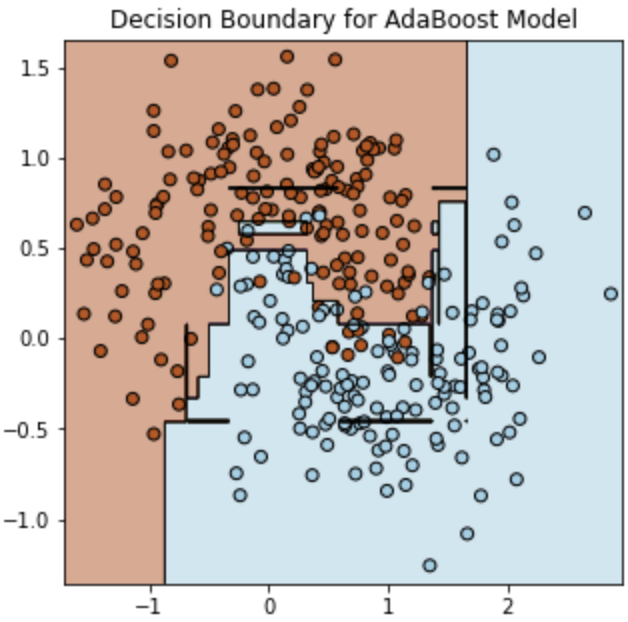
Part 6

The number of leaves, or num leaves, is one of the most important parameters that determines how complex the model will be. You can use it to determine the maximum number of leaves that each weak learner is allowed to have. A large num leaves not only increases the chance of getting hurt from overfitting but also increases the accuracy of the training set. In lightgbm, a leaf-wise tree is deeper than a level-wise tree, so you need to be careful not to overfit! According to the documentation, one simple way is that num leaves = 2^(max depth). As a direct consequence of this, it is essential to fine-tune num leaves in conjunction with max depth.

If we use a large value of max_depth, our model will likely be over fit to the train set.

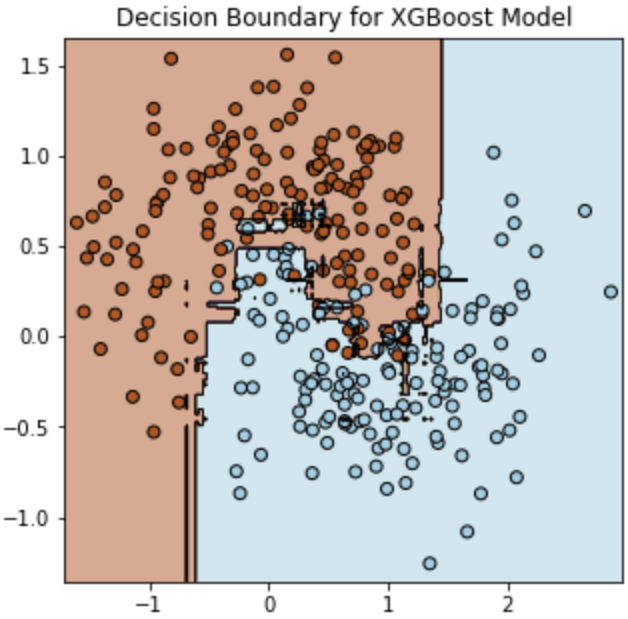
Part 7

Adaboost Model:



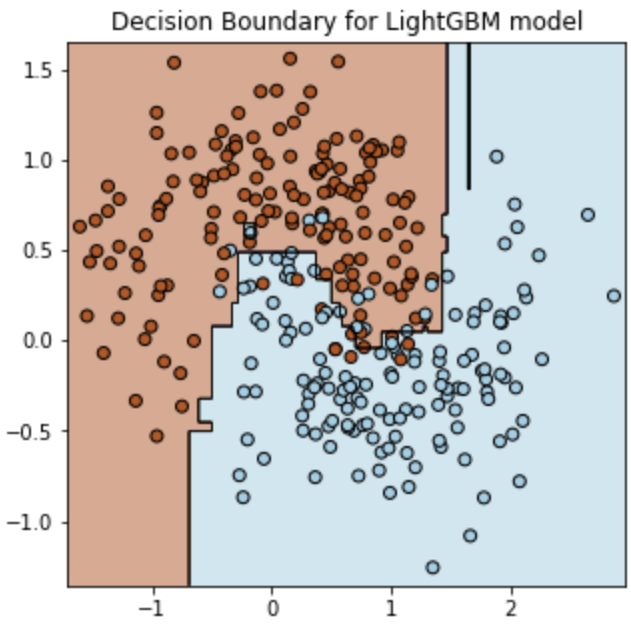
Accuracy Score: 0.9233333333333333

XGBoost Model:



Accuracy Score: 0.9166666666666666

LightGBM Model:




```
Accuracy Score: 0.9233333333333333
```

Question 3.

Trained a Bayes classification model on the dataset, (using sklearn)(tuned the hyperparameters accordingly)

Hyperparameters:

```
'var_smoothing': [1e-9, 1e-8, 1e-7, 1e-6],
'priors': [[0.25, 0.75], [0.33, 0.67], [0.5, 0.5], [0.67, 0.33],
           [0.75,0.25]]
```

Best Hyperparameter:

```
{'priors': [0.5, 0.5], 'var_smoothing': 1e-09}
```

I chose 3 models of my choice (which were giving good accuracy). Grouped them along with the trained Bayes Classification model, in a Voting Classifier from sklearn. Trained the VotingClassfier again.

Three chosen models:

1. Bagging Classifier
2. XGBoost Model
3. Adaboost Model

Accuracy score for Bagging Classifier: 0.9166666666666666
Accuracy score for XGBoost Model: 0.9166666666666666
Accuracy score for XGBoost Model: 0.9233333333333333
Accuracy score for VotingClassifier: 0.92