PRML LAB 5

Name: - Shalin Jain

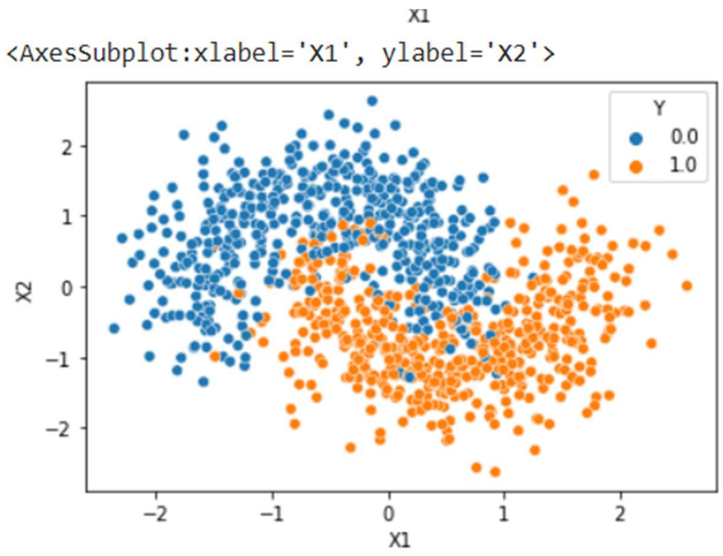
Roll No.: - B21CS070

Problem 1

Part 1

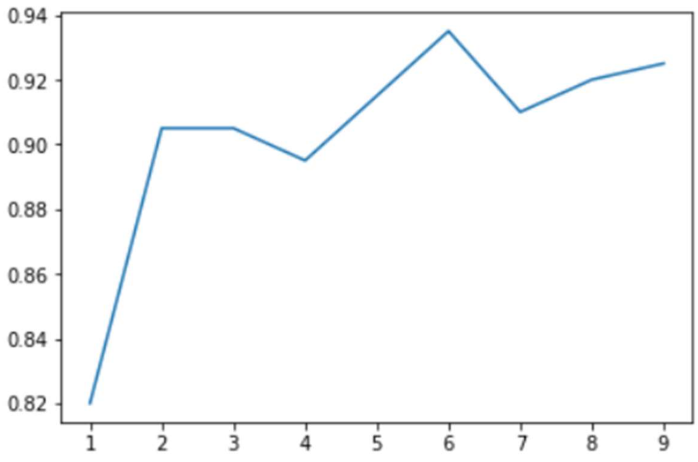
The make _moons is used make the dataset of 1000 points with random_state = 42 and noise = 0.3.

The obtained dataset is plotted as shown.



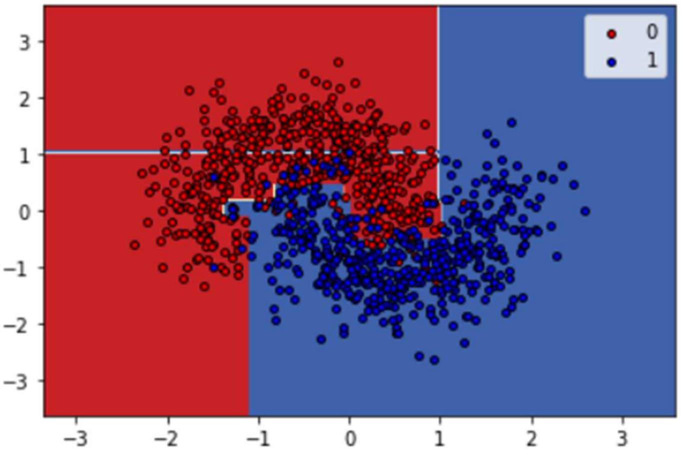
```
X1
<AxesSubplot:xlabel='X1', ylabel='X2'>
```

The obtained dataset is now trained via Decision Tree Classifier by varying the max_depth. We obtained max_depth vs accuracy graph as shown: -



From the graph we can see that best accuracy is at max_depth 6 so we take max_depth as 6 to train the classifier.

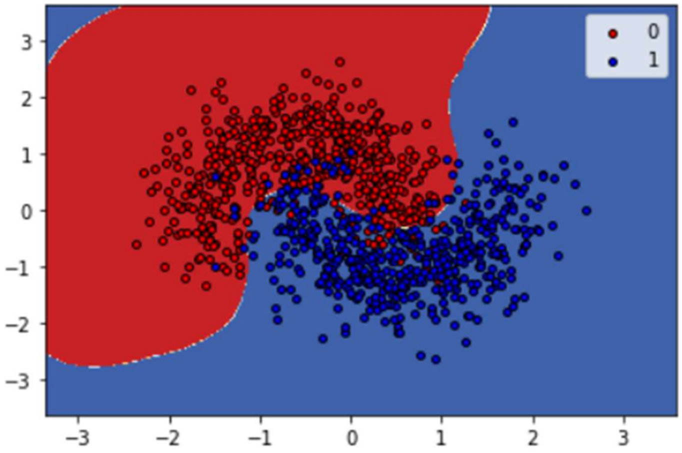
The obtained decision boundary is as shown in the figure



```
0.935
```

Now we have trained the Bagging Classifier with base_estimator as SVC() , n_estimator = 10 and random_state = 42.

The obtained decision boundary is as follows: -

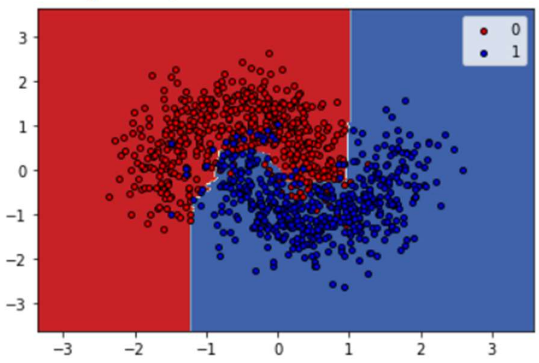


```
0.925
```

The decision is smoother than the decision tree classifier and accuracy is much what same as the decision tree classifier

Now we trained Random Forest Classifier with max_depth = 6.

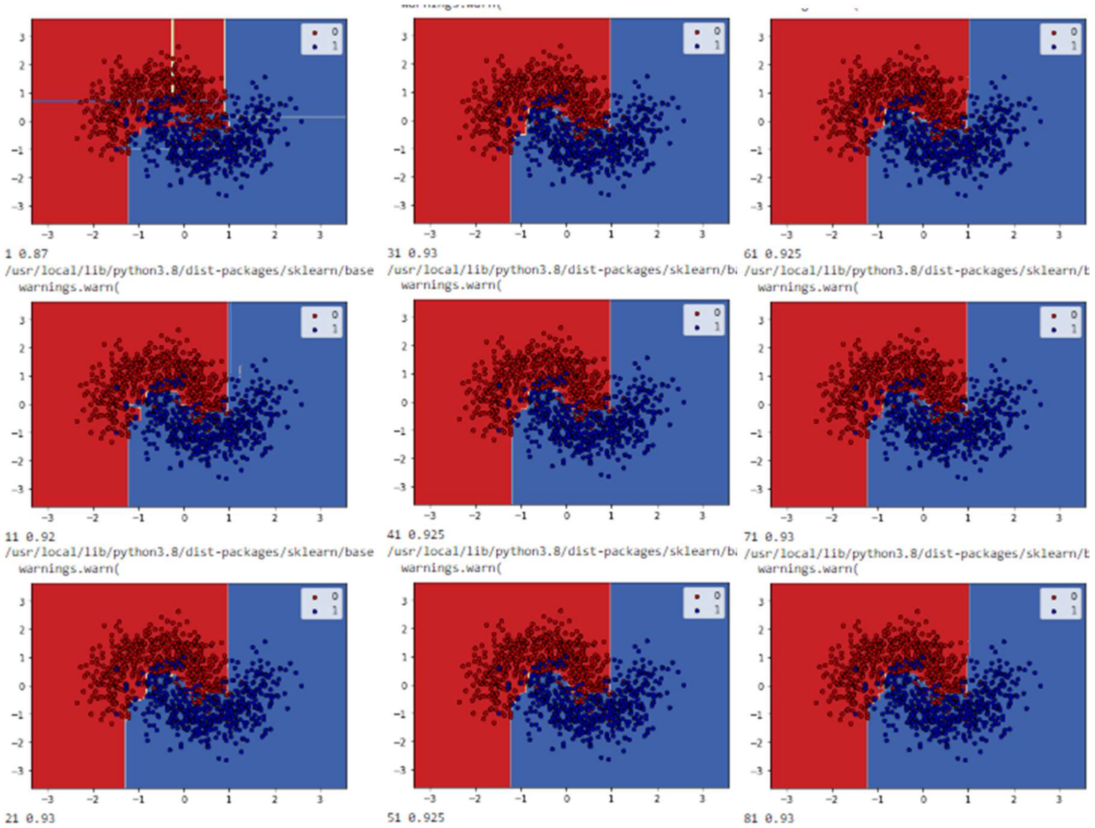
The obtained decision boundary is as shown: -



```
0.93
```

We can see that the accuracy increases with the classifier. The random Forest Classifier shows highest accuracy score as it takes multiple decision trees' decision to classify the datapoints.

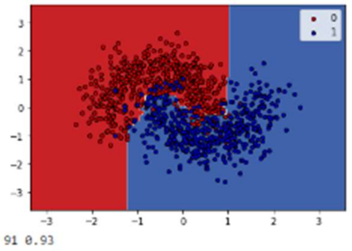
On varying the n_estimator in random forest classifier we can see that the accuracy first increases after a certain number of estimators it become constant. To check this, I have varied the number of estimators from 1 to 100 with a gap of 10. The plotted boundaries and their accuracies along with number of estimators are as shown below.



```
1 0.87
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
11 0.92
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
21 0.93
```

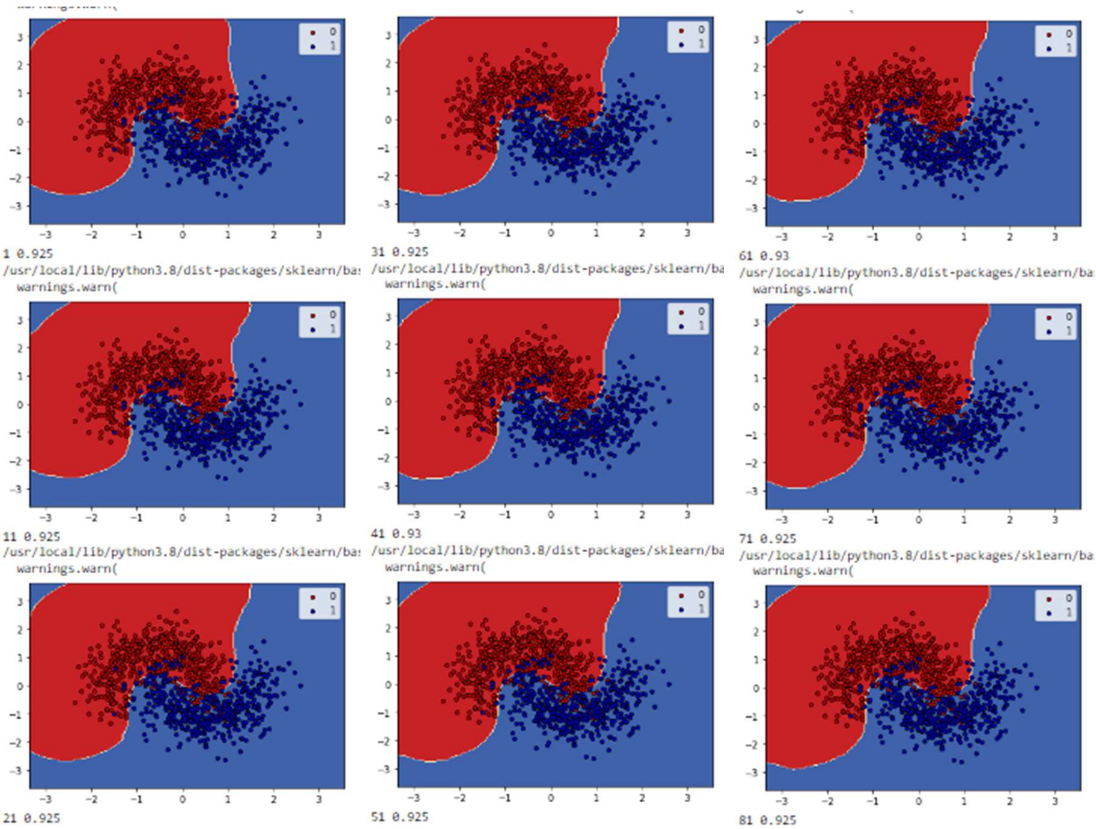
```
  warnings.warn(
31 0.93
/usr/local/lib/python3.8/dist-packages/sklearn/b
  warnings.warn(
41 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/b
  warnings.warn(
51 0.925
```

```
61 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/t
  warnings.warn(
71 0.93
/usr/local/lib/python3.8/dist-packages/sklearn/t
  warnings.warn(
81 0.93
```

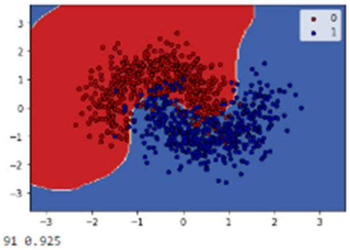


```
91 0.93
```

The above same procedure is applied to bagging classifier we can see that there is a little increase in the first two-three cases and the it became constant. To check this, I have varied the number of estimators from 1 to 100 with a gap of 10. The plotted boundaries and their accuracies along with number of estimators are as shown below.



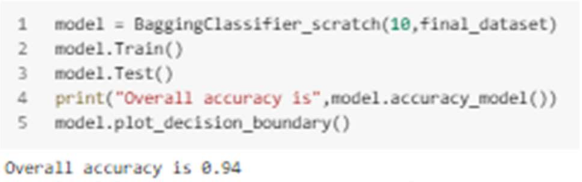
```
1 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/ba:
  warnings.warn(
11 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/ba:
  warnings.warn(
21 0.925
31 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
41 0.93
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
51 0.925
61 0.93
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
71 0.925
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
81 0.925
```



```
91 0.925
```

Part 2

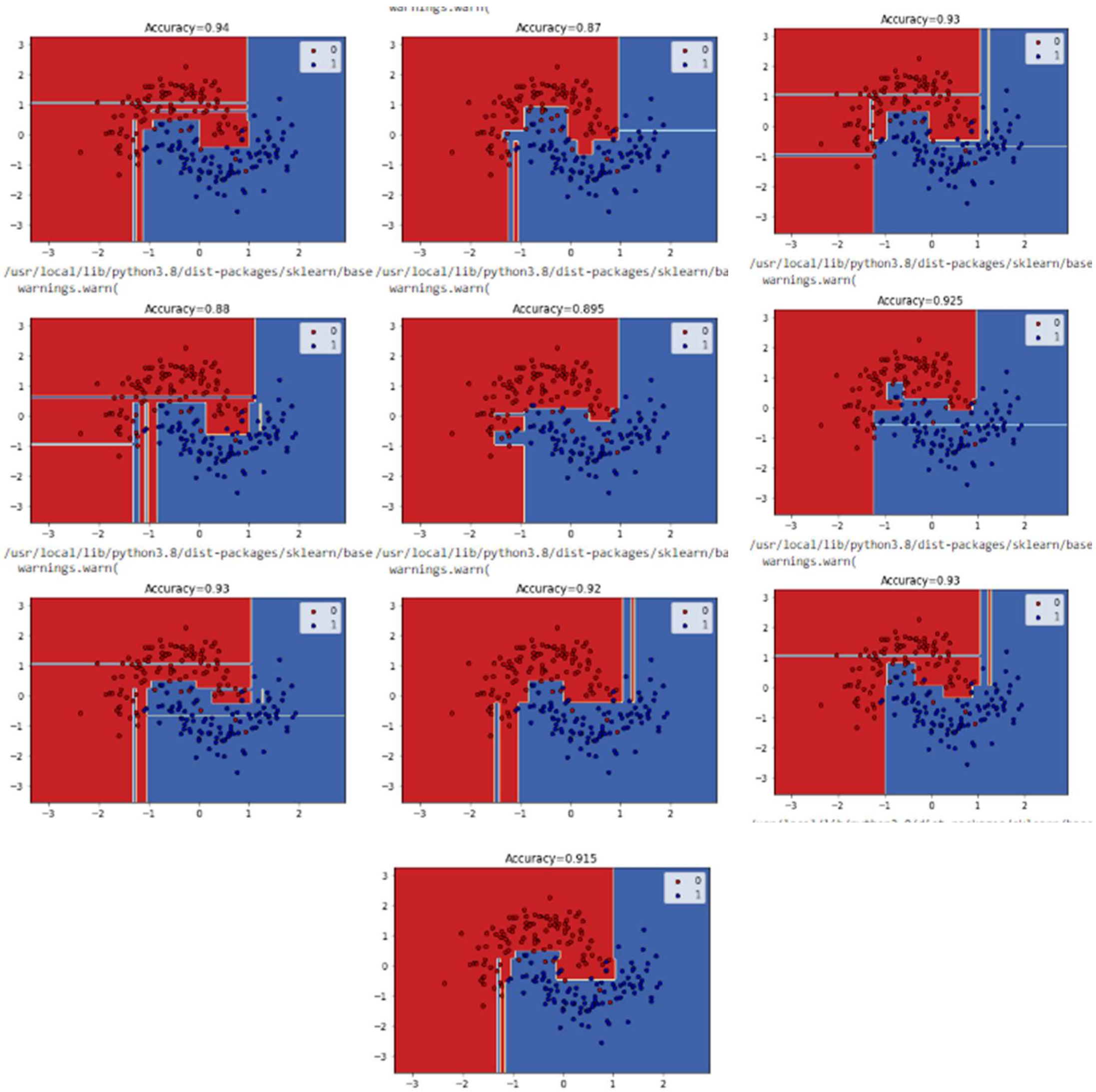
In part 2 we have to make bagging classifier from scratch and plot the decision boundary. I have made different dataset as show in the code and trained Decision Tree Classifiers on these datasets and used maximum voting to get accuracy. The obtain accuracy is 90.5% as shown.



```
model = BaggingClassifier_scratch(10,final_dataset)
model.Train()
model.Test()
print("Overall accuracy is",model.accuracy_model())
model.plot_decision_boundary()
```

```
Overall accuracy is 0.94
```

Also, I have plotted the individual trees along with their accuracies as shown. It can be seen that the overall accuracy is better than most of the individual classifiers.



```
warnings.warn(
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
```

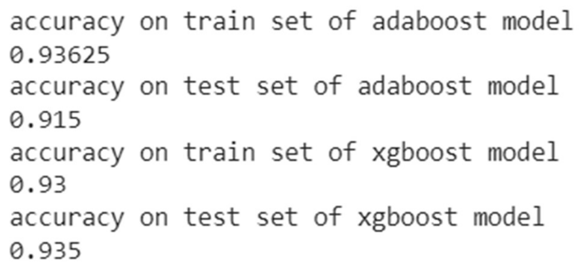
```
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/ba
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/sklearn/base
  warnings.warn(
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/base
```

Problem 2

Part 1,2,3

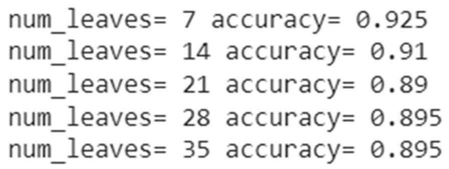
Installed the classifiers as stated in the problem and trained these classifier on the dataset used in problem above. The obtained accuracy on the training dataset as well as the testing dataset is as follows: -



```
accuracy on train set of adaboost model
0.93625
accuracy on test set of adaboost model
0.915
accuracy on train set of xgboost model
0.93
accuracy on test set of xgboost model
0.935
```

Part 4

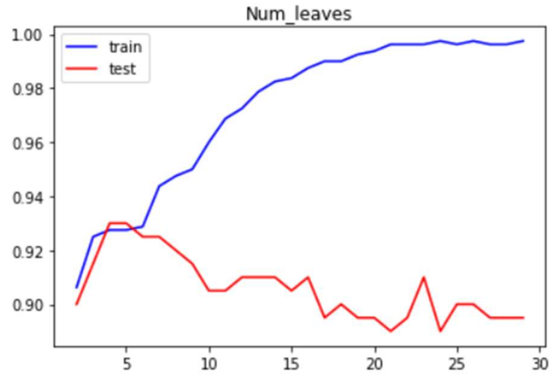
Trained LightGBM model on the dataset and got the accuracies for different values num_leaves as follows: -

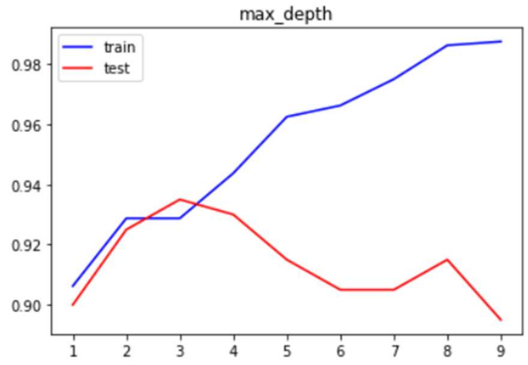


```
num_leaves= 7 accuracy= 0.925
num_leaves= 14 accuracy= 0.91
num_leaves= 21 accuracy= 0.89
num_leaves= 28 accuracy= 0.895
num_leaves= 35 accuracy= 0.895
```

Part 5

I trained model on different values of num_leaves and max_depth and obtained accuracy vs respective quantities graph as follows.





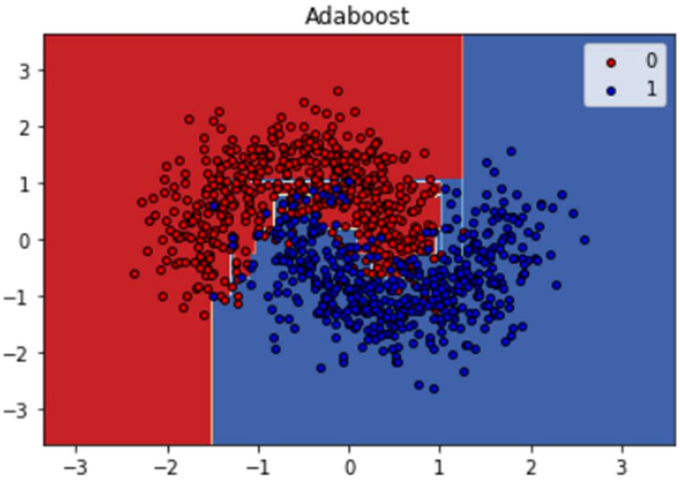
Part 6

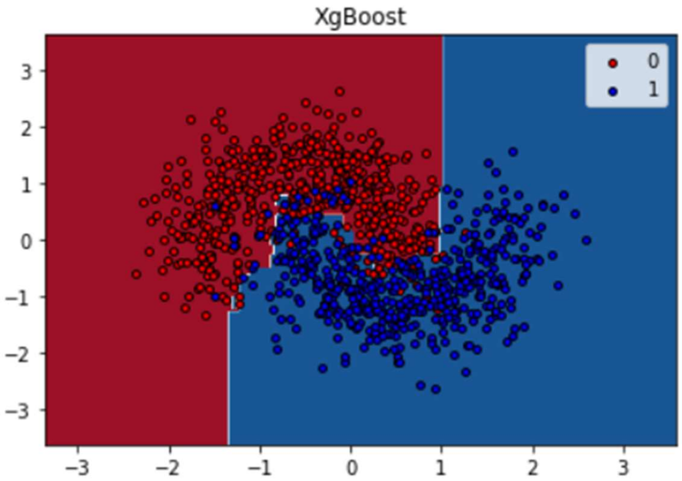
We can see that accuracy of both test and the train increases with increase in respective quantities but after particular values i.e. num_leaves = 5 and max_depth = 3 the model starts to overfit as accuracy of train increases but test starts to decrease and does not increase more than the obtained maximum as shown in the graph. So we can say that at these particular values model starts to overfit. We can use combination of num_leaves and max_depth as stated above for better accuracy and max_depth for overcoming the overfitting.

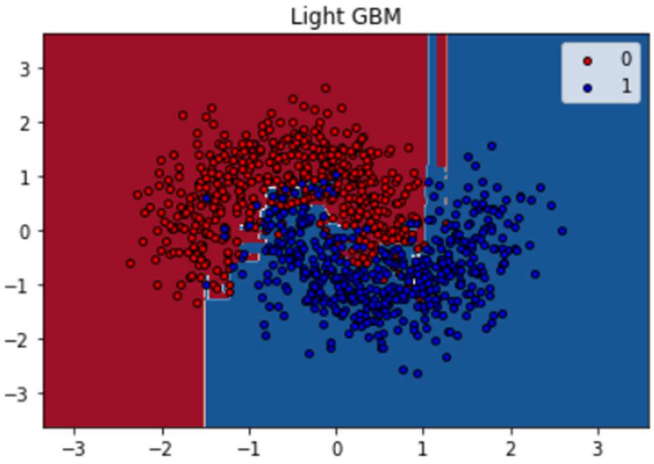
Part 7

Plotting the decision tree boundaries and using the accuracy we can say that xgboost and adaboost performs similar while the LightGBM performs better with less num_leaves.

The obtained decision boundaries of all 3 models are as follows: -



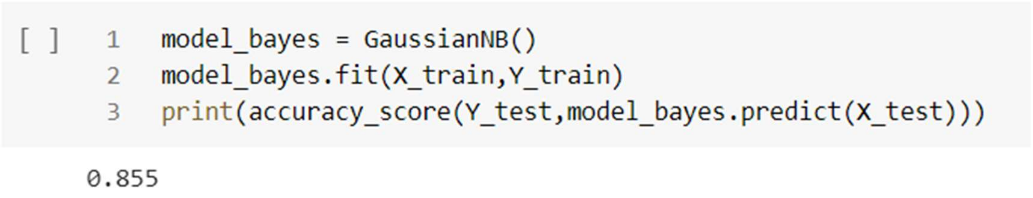




Problem 3

Part 1

Trained gaussian classifier as done in the code. The obtained accuracy is as shown: -

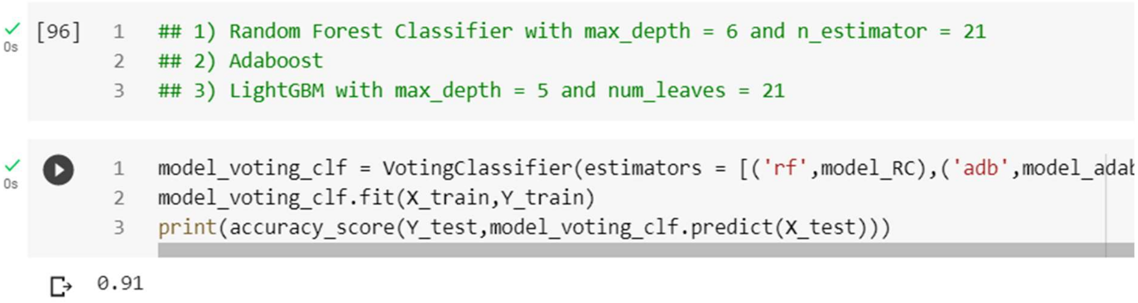


```
[ ]
model_bayes = GaussianNB()
model_bayes.fit(X_train,Y_train)
print(accuracy_score(Y_test,model_bayes.predict(X_test)))
```

```
0.855
```

Part 2

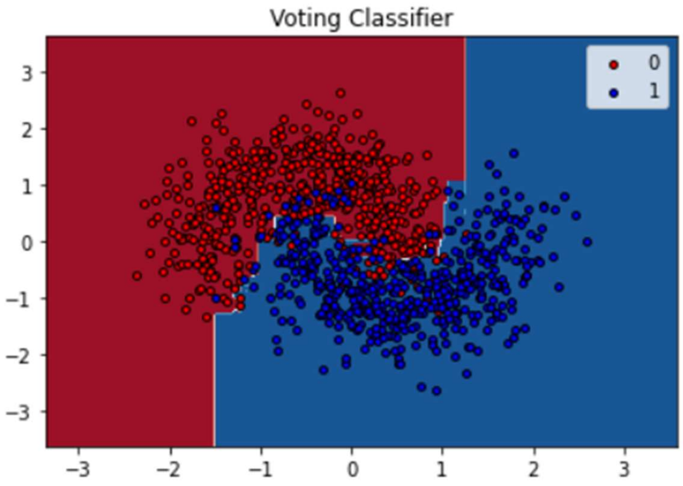
Used Voting classifier as shown.



```
[96]
## 1) Random Forest Classifier with max_depth = 6 and n_estimator = 21
## 2) Adaboost
## 3) LightGBM with max_depth = 5 and num_leaves = 21
```

```
model_voting_clf = VotingClassifier(estimators = [('rf',model_RC),('adb',model_adab
model_voting_clf.fit(X_train,Y_train)
print(accuracy_score(Y_test,model_voting_clf.predict(X_test)))
```

```
0.91
```


Voting Classifier