PRML LAB 5
Bagging & Boosting

Vansh Agarwal
B21AI042

Q1. Bagging

1. Data preprocessing and visualisation

Data has been made with the `sklearn's make_moon function` and split into train test sizes of 0.8 and 0.2 respectively. The plot has been shown in Fig 1.

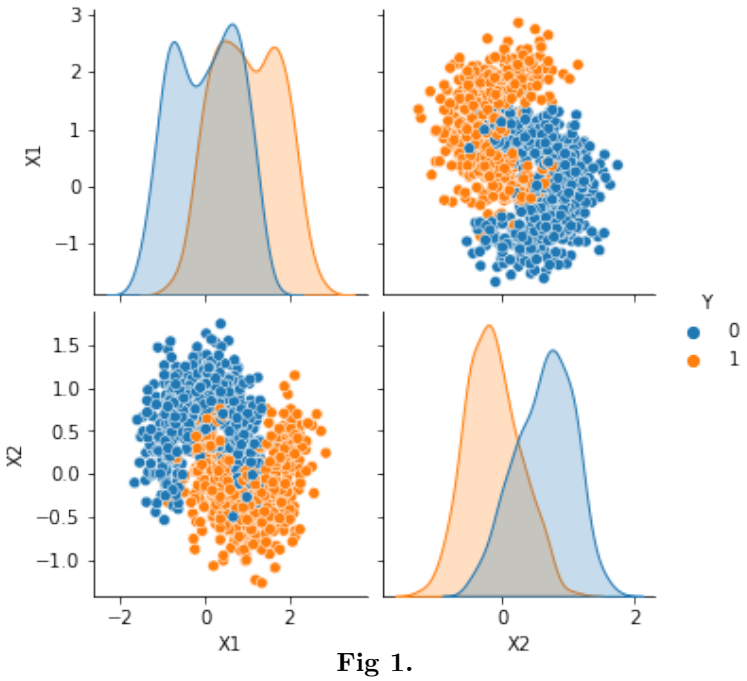


Fig 1.

Decision tree classifier

Hyper parameter tuning has been done with the following values of `max_depth: [5, 10, 15, 50, 100, 250, 500, 1000]`. Out of these the best accuracy has been obtained at the optimal **`max_depth = 250`**, giving an **accuracy of 0.925**. Figure 2 shows the confusion matrix & Decision Boundary for the same.

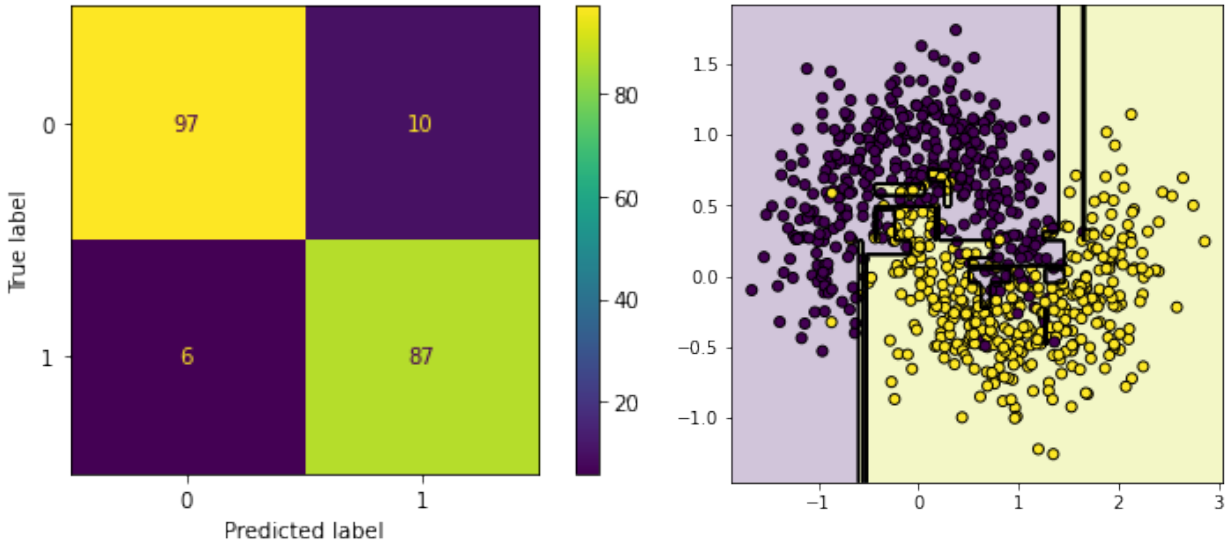


Fig 2.

Bagging classifier

Bagging Classifier is an ensemble learning model which takes into account the predictions of various estimators, in this case, decision trees and assigns the class based on majority votes. Hence, its performance is generally better than single decision trees. **Accuracy of model: 0.95.** The confusion matrix and decision boundary are shown in fig3.

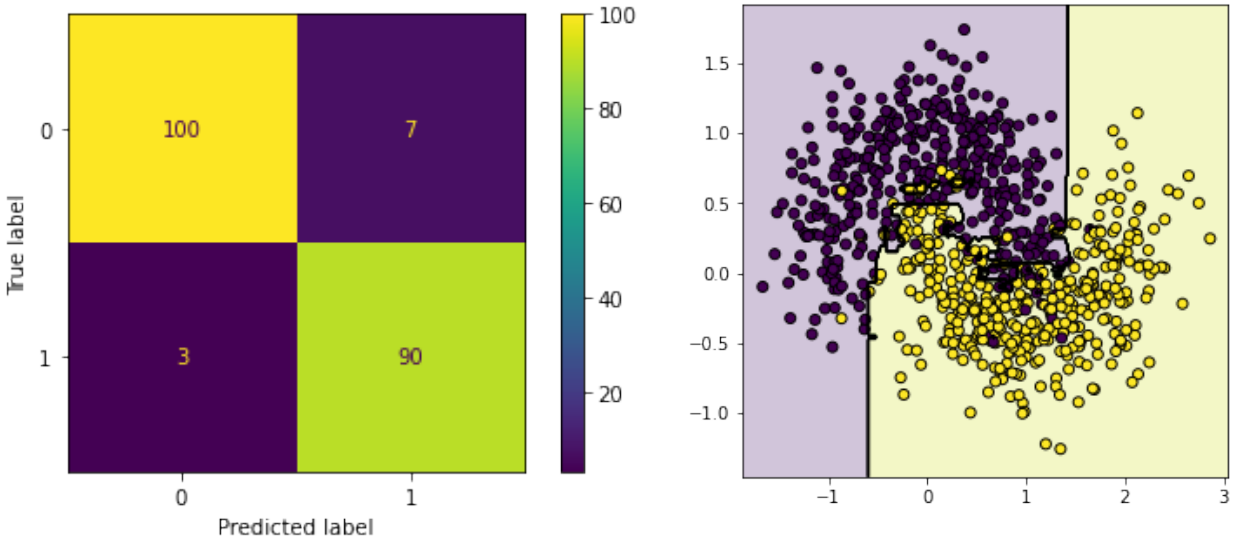


Fig 3.

Random forest classifier

Random forest classifier is also an ensemble learning model which takes into account the predictions of various estimators, in this case, decision trees and assigns the class based on majority votes. Hence, its performance is generally better than single decision trees. **Accuracy of model: 0.945.** The confusion matrix and decision boundary are shown in fig4.

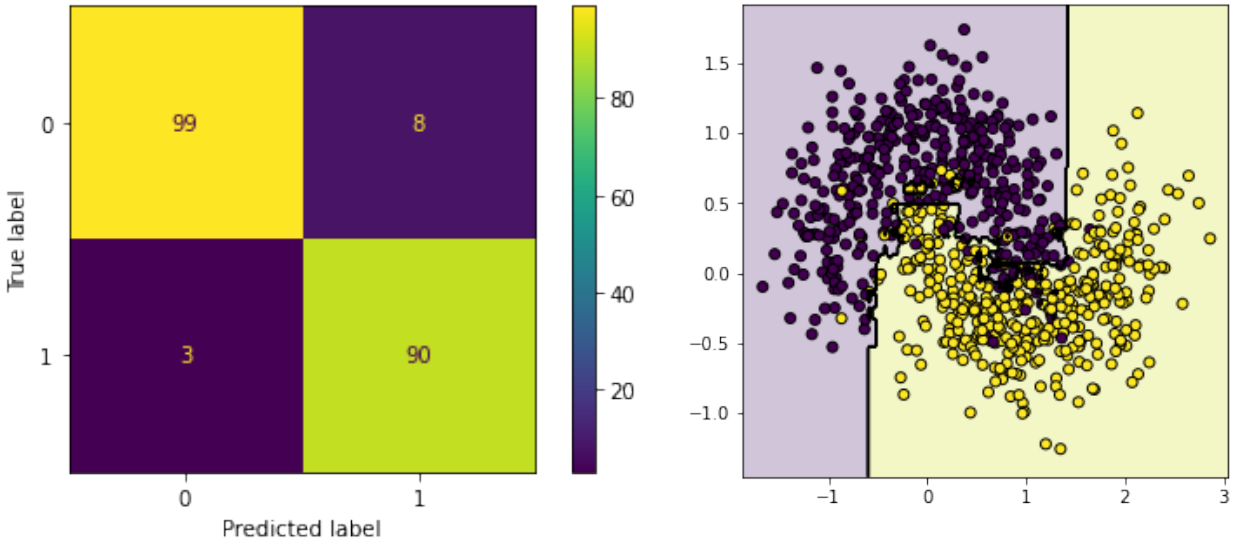


Fig 4.

Variation in `n_estimators` for Bagging Classifier

The n_estimators have been ranged from: **`[1, 10, 25, 50, 100, 500].`** For very low values of n_estimators the model might underfit. For very high values of n_estimators the model might overfit the train set.

The **best accuracy of 0.955** has been achieved with **`n_estimators` set to 50**. The decision boundary plots and respective accuracies have been shown in Fig 5.

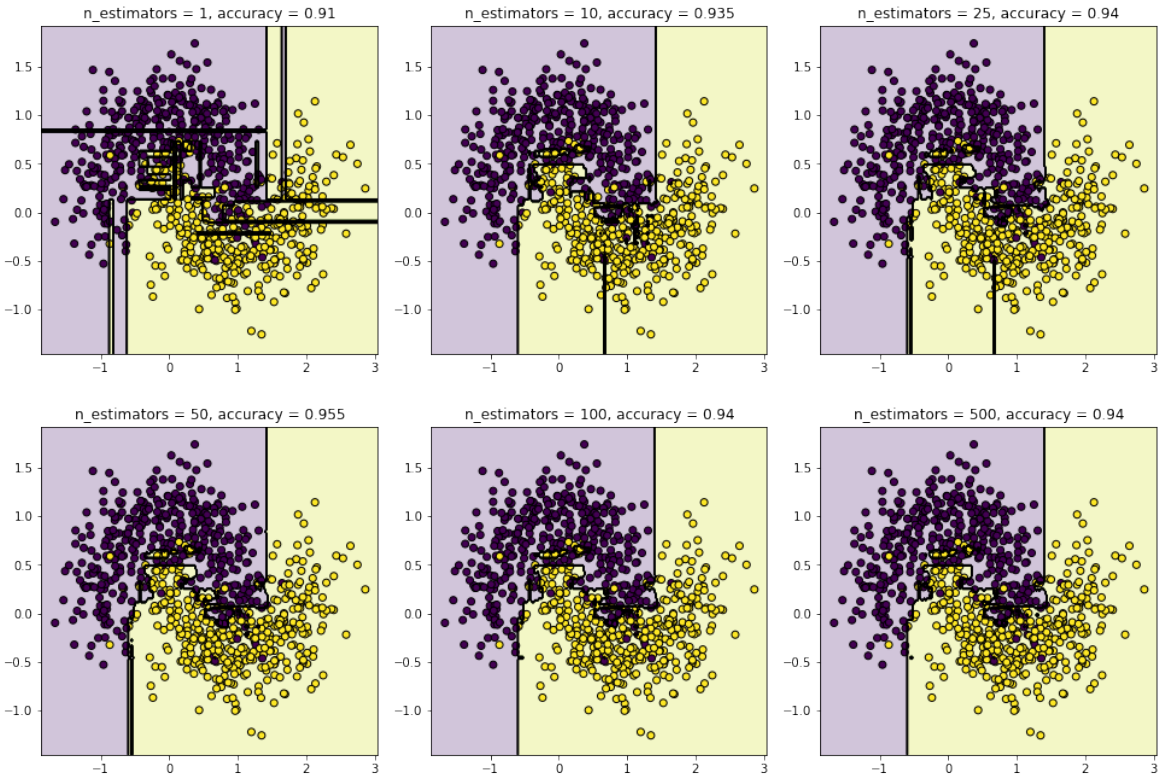


Fig 5 Variation in n_estimators Bagging Clf

Variation in `n_estimators` for Random Forest Classifier

The n_estimators have been ranged from: **`[1, 10, 25, 50, 100, 500].`** For very low values of n_estimators the model might under fit. For very high values of n_estimators the model might overfit the train set.

The **best accuracy of 0.95** has been achieved with **`n_estimators` set to 100**. The decision boundary plots and respective accuracies have been shown in Fig 6.

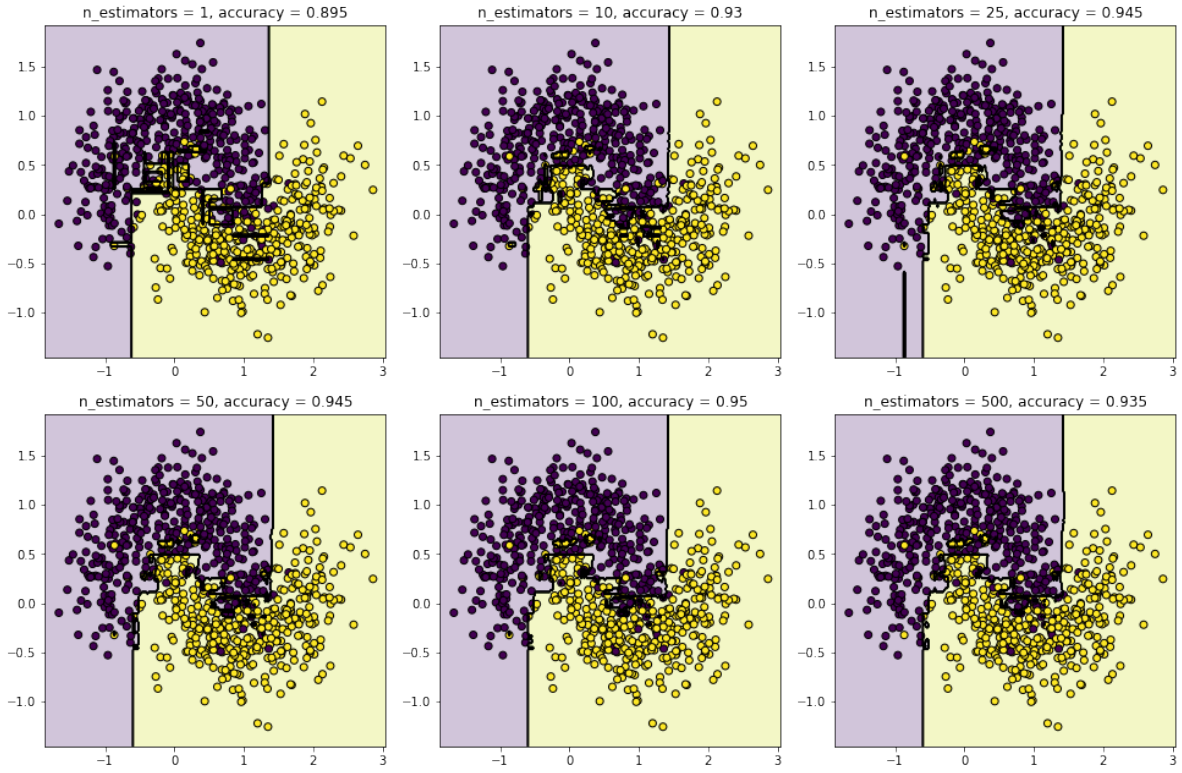


Fig 6 Variation in n_estimators Random Forest Clf

2. Scratch Implementation

The bagging classifier trained from scratch gives has been trained with **`n_estimators = 10`**
It gives an **average accuracy of 0.91.** The decision boundaries and respective accuracies of each tree in the forest has been summarised in Fig7.

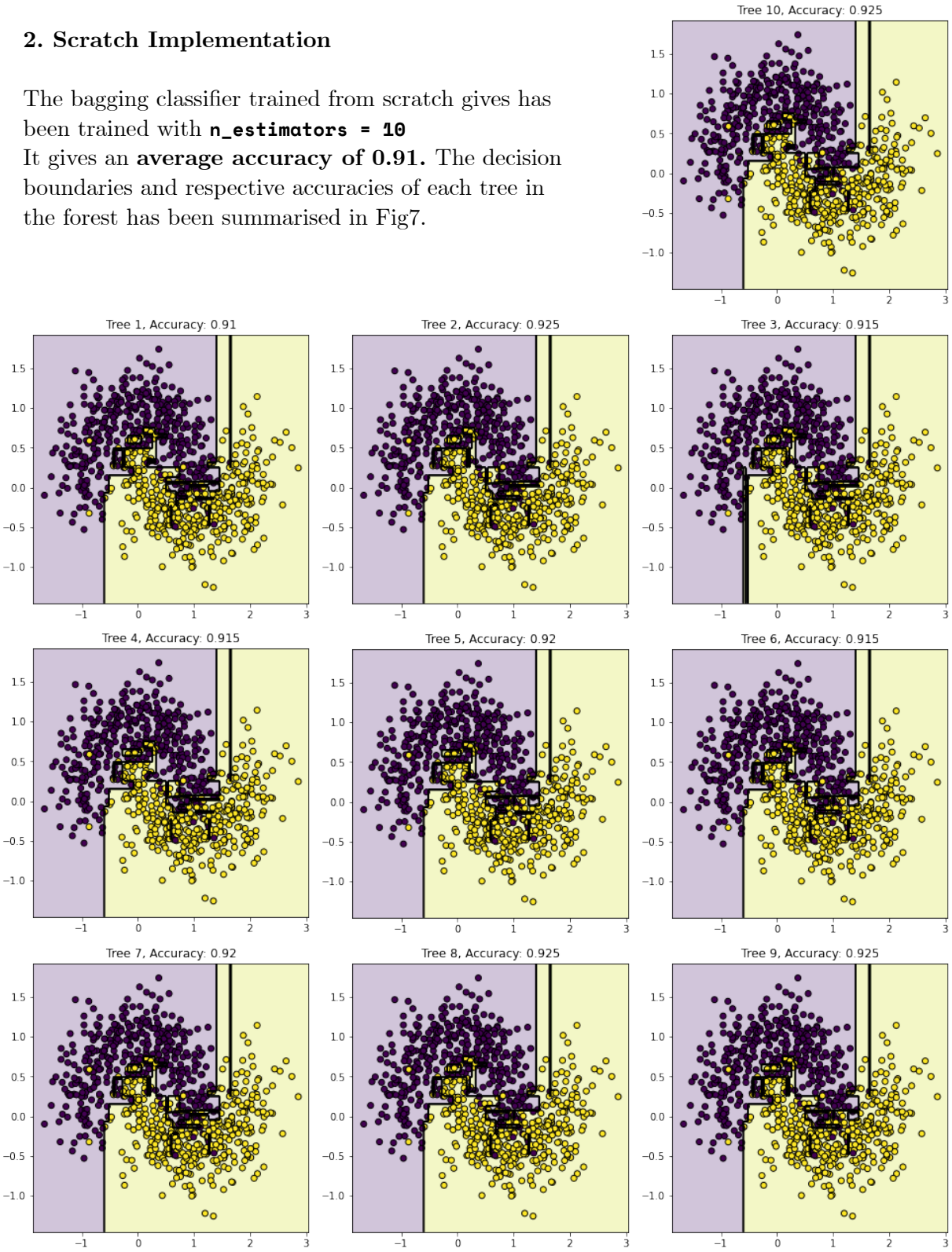
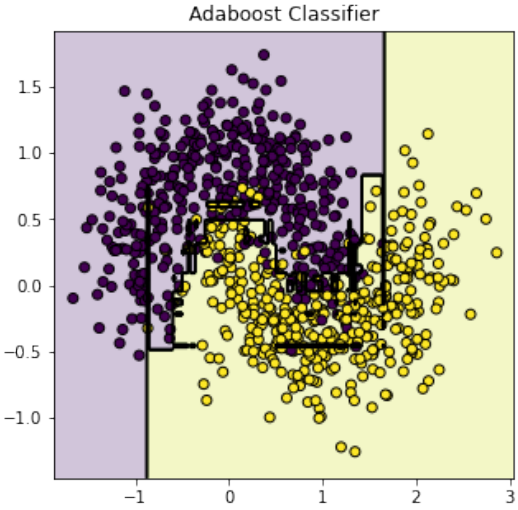


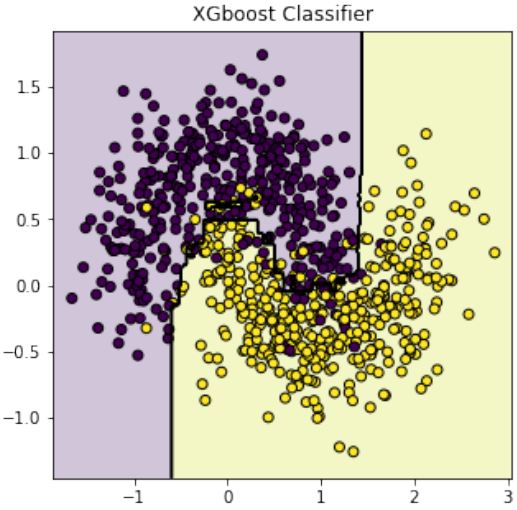
Fig 7 Summarising separate trees of Bagging Classifier

Q2. Boosting

AdaBoost Model
Accuracy: 0.93



XGBoost Model
Accuracy: 0.945



LightGBM Classifier

Hyperparameter tuning has been done with two parameters: **`max_depth`** and **`num_leaves`** simultaneously using `sklearn's gridSearchCV`. The best performing model is obtained with **num_leaves = 10** and **max_depth = 5**, **giving an accuracy of 0.92 (Fig 8)**. When we increase the values of hyper parameters, the variance starts to increase i.e. the model starts overfitting.

For each level of depth d the leaves at that level is less than equal to 2^d. Summing over all the levels we arrive at the relation shown below:

$$numLeaves <= 2^{maxDepth}$$

Parameters to avoid overfitting

- Using **small `num_leaves`**. A large value of `num_leaves` can increase the complexity of the model and makes it overfitting to the training data.
- Setting **`min_data_in_leaf` to larger values** reduces the depth of the tree and hence reduces overfitting of the model.
- Set **less value to `max_depth`** to avoid tree becoming complex and overfitting of train data.
- Use **small value for `max_bin`**. `Max_bin` determines the number of bins(categories) the features will be divided into.

Parameters to improve Accuracy

- Use **large value for `max_bin`**. `Max_bin` determines the number of bins(categories) the features will be divided into. The feature will be divided into more bins and hence gives better accuracy i.e. decision boundaries become more sensitive.
- Use **large value for `n_estimators`**. `N_estimators` is the number of estimators (for eg. decision trees in a random forest). The more estimators we use to predict and vote, the more is the tendency for accuracy to be better.
- Using **large `num_leaves`** leads to more splits in the tree and improves accuracy. This however leads to overfitting data.
- Use **small `learning_rate`** so that we don't skip any values while training the model.

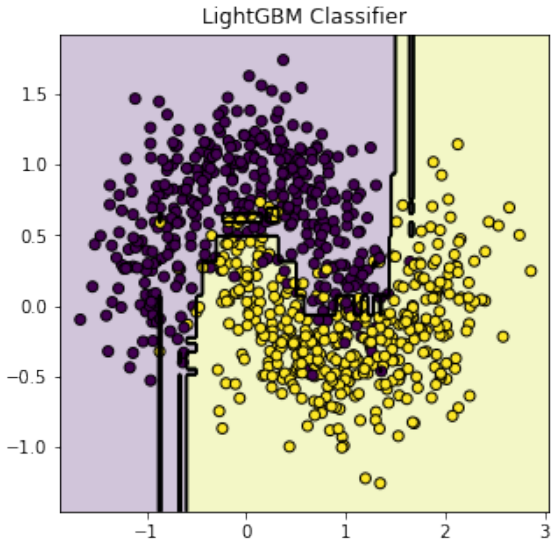


Fig8

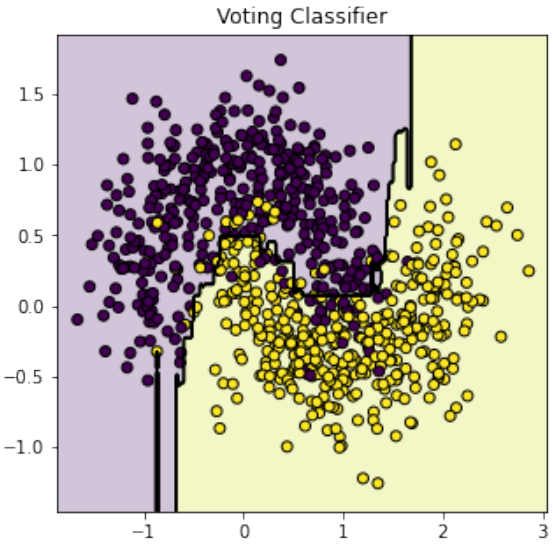


Fig9

Q3. Voting Classifier

Naïve Bayes classifier alone gives an accuracy of 0.88

For voting Classifier we have chosen Naive Bayes along with LightGBM model, XGBoost Model, AdaBoost Model (purely on the basis of their good individual accuracies). **Voting classifier gives an accuracy of 0.94 . Decision Boundary as shown in Fig 9.**