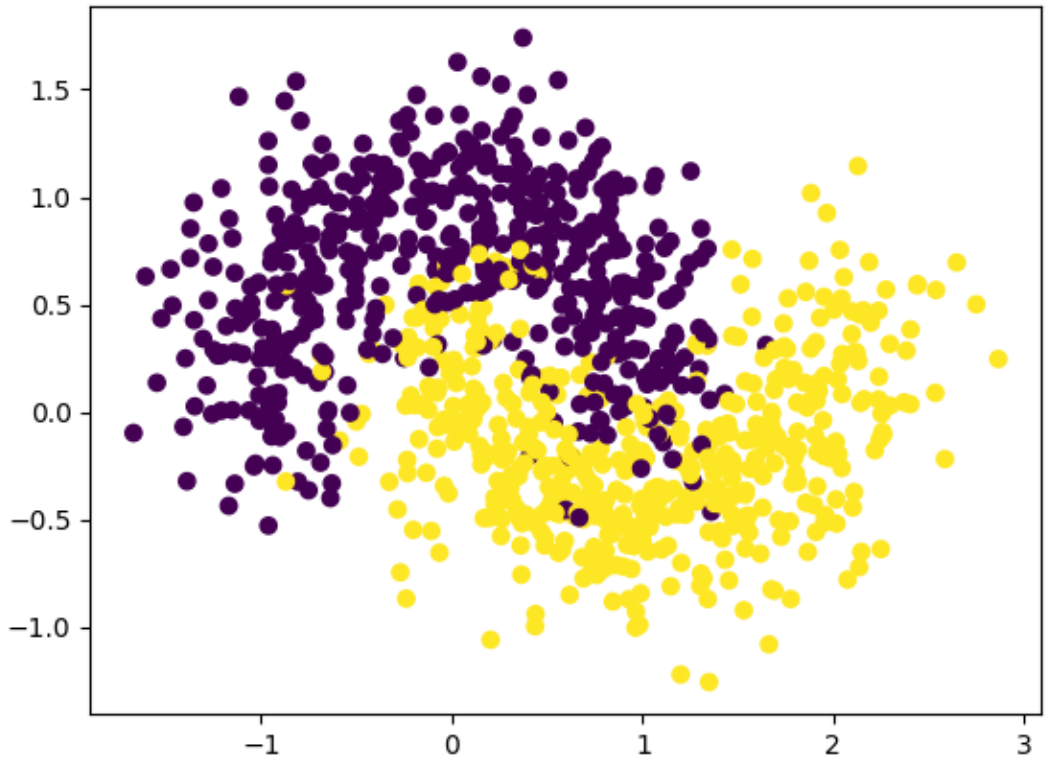
PRML LAB 5 REPORT

MANISH(B21CS044)

QUESTION 1

1. Create a dataset with 1000 samples, using the 'make_moon's function of sklearn (choose random_state=42, noise=0.3). Perform appropriate preprocessing, train and test split of the dataset.
   Plot the generated dataset.

Created the dataset using make_moon's function from sklearn library.
Here is the generated Dataset.



Train a simple decision tree classifier from sklearn and plot the decision boundary for the same. Perform hyperparameter tuning for finding the best value of max_depth of the decision tree. [5 marks]

Performed tuning of the max_depth and trained the Decision Tree Classifier for the same.
Plotted Decision boundary using mlxtend library.
Here are the values of best max_depth and corresponding accuracy and decision boundary.

Max_depth : 8

Accuracy : 0.9233333333333333

Train a BaggingClassifier from sklearn, on the same dataset, and plot the decision boundary obtained. [5 marks]

Trained the bagging classifier using sklearn.ensemble on the same dataset.

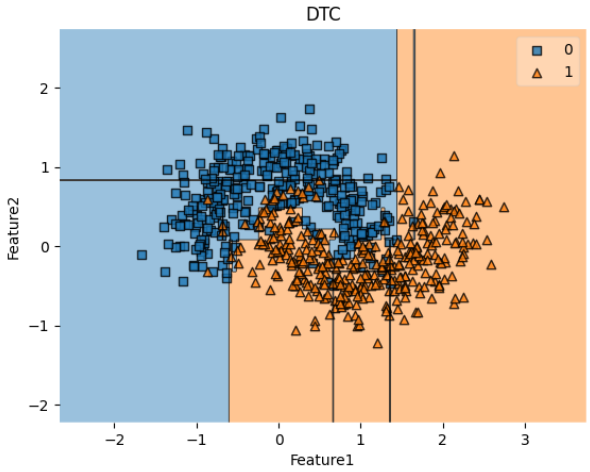
Accuracy and decision are below in comparison.

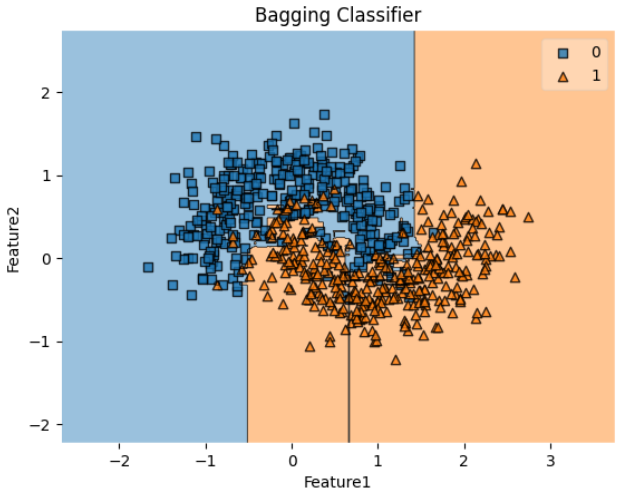
Train a RandomForest classifier from sklearn and plot its decision boundary. Compare the models (all 3), their decision boundaries, and their accuracy metrics. [5+ 5 marks]

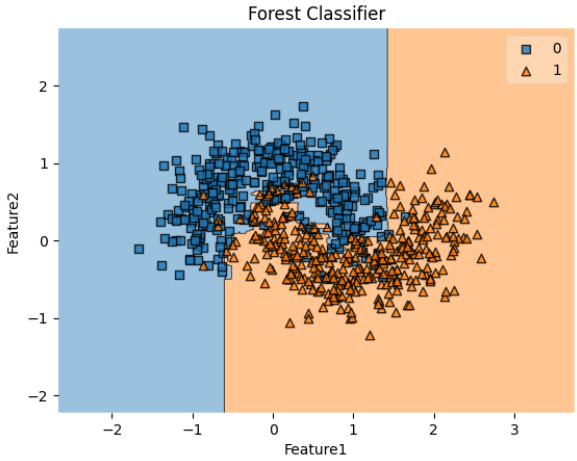
Trained all the Classifiers Normal DTC with best max_depth ,Bagging Classifier,Random forest Classifier.
Here are the accuracies and Decision Boundaries for all.
DTC : 0.9233333333333333

Bagging Classifier : 0.9166666666666666

Random Forest Classifier : 0.9166666666666666







Almost all of the classifiers have the same accuracy score, but the best classifier is the decision tree classifier with the best max depth. Because it goes until the dataset's maximum depth. But as soon as we increase the n_estimators in the Bagging Classifier and Random forest the accuracy of the both will increase.
The decision Boundary of DTC is slightly more separable than other two classifiers.

Vary the number of estimators for the BaggingClassifier and RandomForestClassifier, and comment on the obtained decision boundaries and their accuracies. [10 marks]

Varied the n_estimators in both the classifiers from 10 to 100 and here are the respective accuracies for each Classifier.

RandomForestClassifiers accuracy:

[0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333, 0.9233333333333333]

BaggingClassifier accuracy:

[0.9166666666666666, 0.9333333333333333, 0.92, 0.9266666666666666, 0.9266666666666666, 0.92, 0.9233333333333333, 0.9333333333333333, 0.9233333333333333, 0.9266666666666666]

In Bagging Classifier, as the number of estimators goes up, the decision boundary moves closer to the Decision Tree with the best maximum depth value. This is because the Bagging Classifier takes an average of the predictions of the Decision Trees with the best max depth value. As the number of estimators goes up, the decision boundary moves closer to the Decision Tree with the best max depth value.

Random Forest Classifier moves the decision boundary closer to the Decision Tree with the best max depth value as the number of estimators goes up. This is because the Random Forest Classifier takes an average of the predictions of the Decision Trees with the best max depth value. As the number of estimators goes up, the decision boundary moves closer to the Decision Tree with the best max depth value.

Also, the accuracy scores of the Bagging Classifier and the Random Forest Classifier are almost the same.

2. Implement a Bagging algorithm from scratch. [20 marks]
   Note: The code should be well commented and the role of each function should be mentioned clearly.
   Apply the above scratch bagging algorithm with n_estimators = 10, train it on the same dataset as above. Summarize how each of the separate trees performed (both numerically and visually). How do they perform on average? [10 marks]

Implemented Bagging Classifier from scratch using Class and made a class as BagClassifier.
def __init__(self,n_esitimators):
For initializing the classifier and give the number of estimators
def fit(self,X,y):
To fit the Classifier using all the small estimators as a Decision Tree.

def predict_bag(self,X):

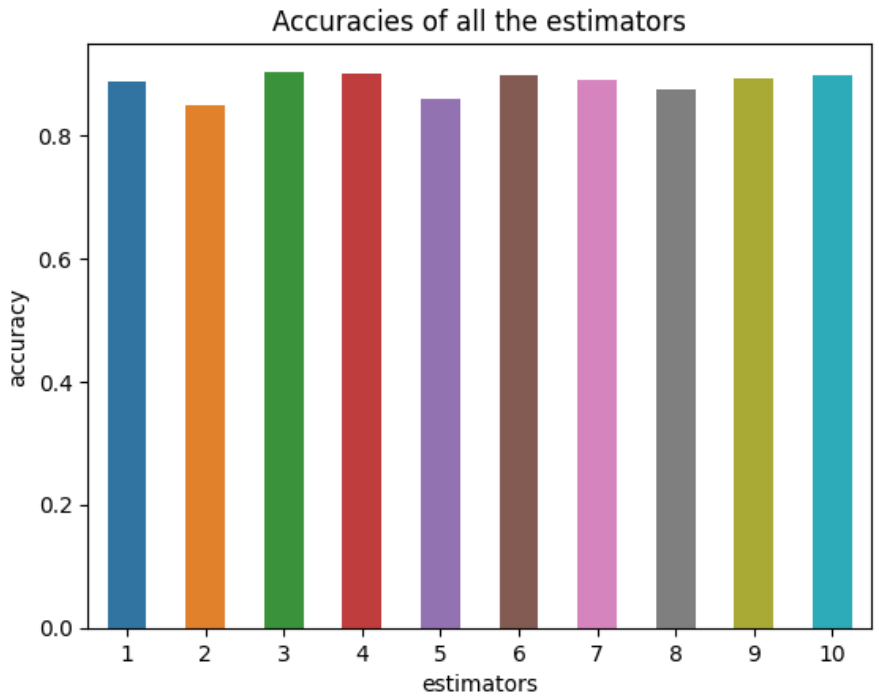
To predict the class of the datapoint using the majority rule.

Here is the accuracy of the implemented Bagging Classifier.

accuracy of bag_clf is: 0.9166666666666666

Here the accuracies of all the estimators and barplot for the visualization of the same.

Accuracies of all estimators are: [0.8866666666666667, 0.85, 0.9033333333333333, 0.9, 0.86, 0.8966666666666666, 0.89, 0.8733333333333333, 0.8933333333333333, 0.8966666666666666]



QUESTION 2

1. Train a AdaBoost Model.
2. Train a XGBoost Model in which subsample=0.7.

Trained the AdaBoost Model Classifier and XGBoost Classifier with subsample = 0.7

3. Print the accuracy on the training set and test set.

Here are the training and test accuracies for both models.

Accuracy of Adaboost on test set : 0.9233333333333333
Accuracy Xgboost on test set : 0.9133333333333333
Accuracy Adaboost on training set : 0.9485714285714286
Accuracy Xgboost on training set : 0.9857142857142858

4. Train a LightGBM model and choose different values for num_leaves.

Trained LightGBM model varying num_leaves.
Here are the accuracies for the same.

accuracy Lgbm with 2 leaves : 0.9066666666666666

accuracy Lgbm with 3 leaves : 0.91

accuracy Lgbm with 4 leaves : 0.9166666666666666

accuracy Lgbm with 5 leaves : 0.9233333333333333

accuracy Lgbm with 6 leaves : 0.92

accuracy Lgbm with 7 leaves : 0.9233333333333333

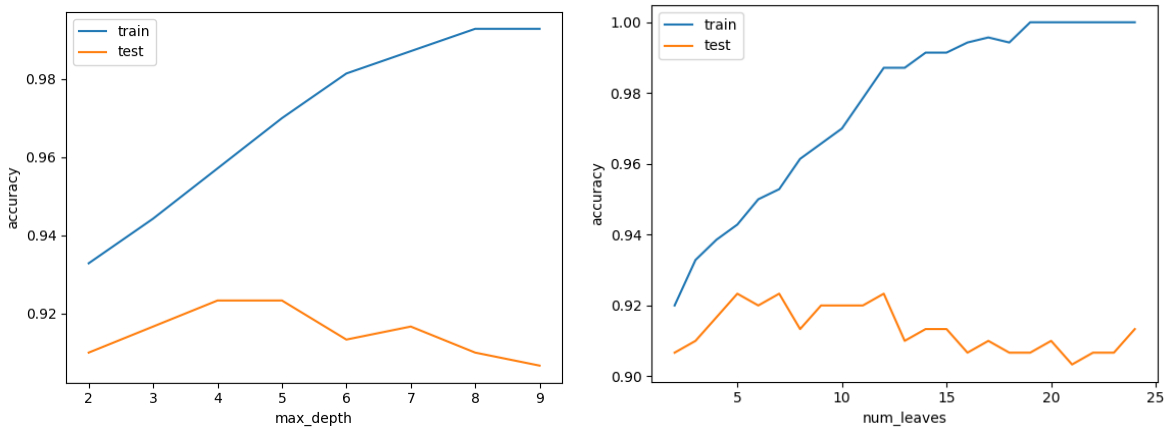
accuracy Lgbm with 8 leaves : 0.9133333333333333

accuracy Lgbm with 9 leaves : 0.92

5. Analyze the relation between max_depth and num_leaves, and check for which value the model starts overfitting.

To check for the value from where it starts overfitting the data we have to calculate accuracy for both training and test datasets.

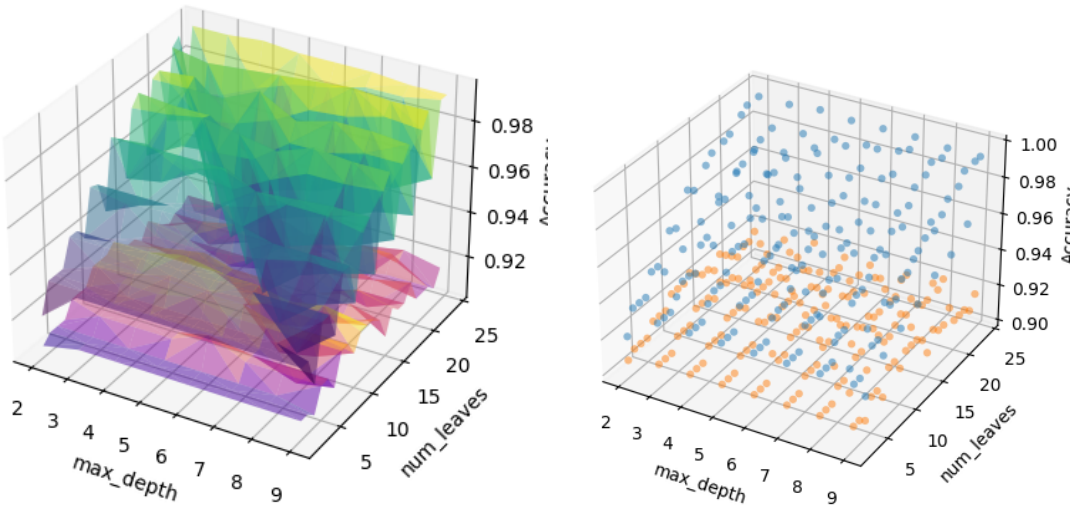
Here are the graphs for the same for both max_depth and num_leaves.



Here I am varying max_depth from 2 to 10 and num_leaves from 2 to 25.
From the above graph we can clearly see that after max_depth = 4 and num_leaves = 5 the model starts overfitting the data.

Best max_depth = 4 and num_leaves = 5

Here is the visualization of the accuracies and max_depth and num_leaves together in a 3d curve:



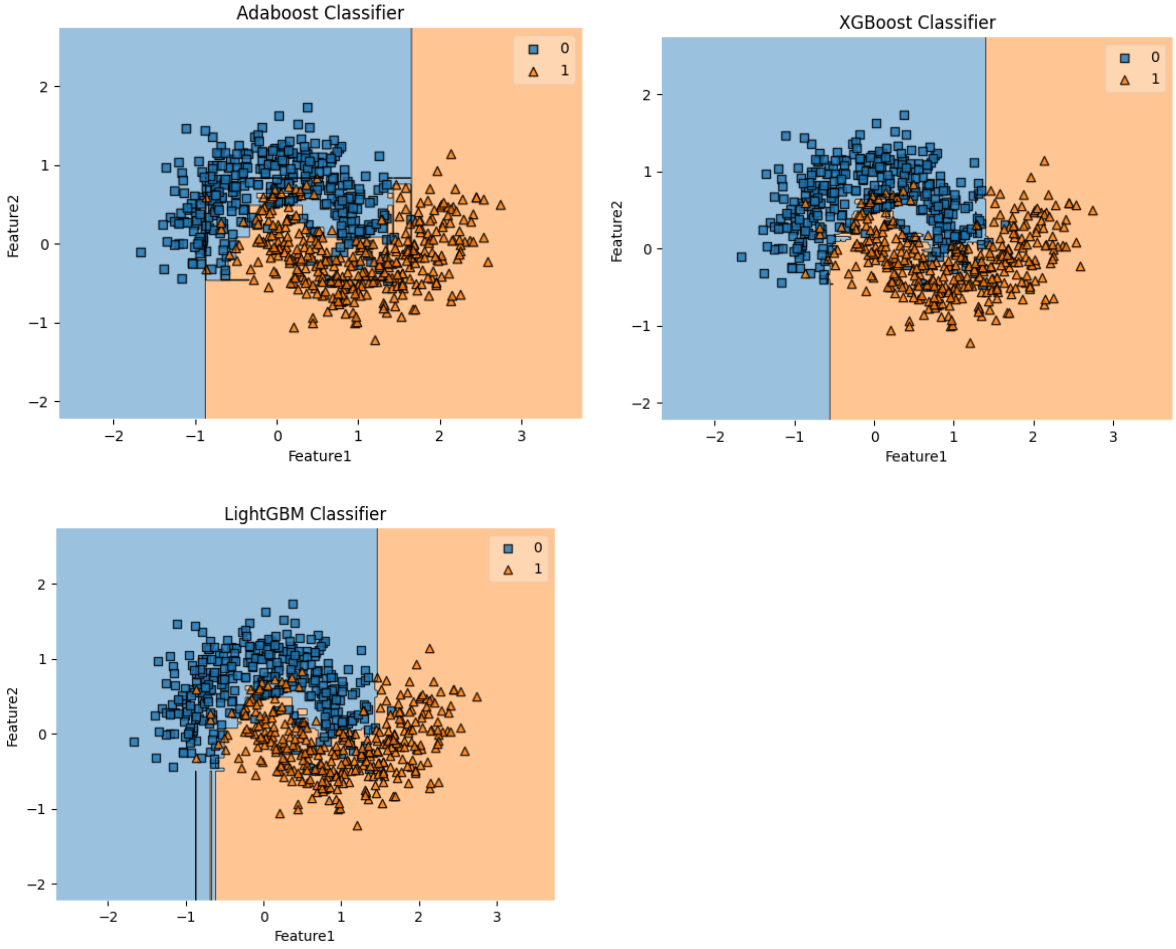
From above two graphs we can clearly see that in max_depth it increases accuracy for test datasets upto max_depth 4 after that it starts to overfitting the data and in case of num_leaves it increases the accuracy of the test data set upto num_leaves 5 after that it starts to overfitting the data

So as a conclusion we can say that best max_depth = 4 and best num_leaves = 5

6. Report which parameters can be used for better accuracy and also which parameter can be used for avoiding overfitting.

The parameters that can be used to get the better accuracy and also avoid overfitting can be max_depth, num_leaves, min_sample_split, learning rate their fine tuning can help to increase the accuracy and also to avoid overfitting we have check if the accuracy on the training set should not increase too much.

7. Plot the decision boundaries for all the 3 models and compare their performance.



Accuracy on test set LGBM : 0.9233333333333333

Accuracy of Adaboost on test set : 0.9233333333333333
Accuracy Xgboost on test set : 0.9133333333333333

Almost all the classifiers perform similarly, just AdaBoost and LGBM perform slightly

well because of the tuning. So,these two differentiate decision boundaries Slightly better.

QUESTION 3

Train a Bayes classification model on the above dataset, (using sklearn)(tune the hyperparameters accordingly)

We have selected Gaussian Naive Bayes because the dataset is Gaussian or normal and Continuous.

Here are the accuracies for the same.
Accuracy on training set : 0.85

Accuracy on test set : 0.8533333333333334

From all the above trained models, choose any 3 models of your choice (which are giving good accuracy). Group them along with the trained Bayes Classification model, in a

VotingClassifer from sklearn. Train the VotingClassfier again. And compare its performance with the models which were individually trained. [10 marks]

We have selected the AdaBoost, XGBoost, LightGBM and Gaussain Naive Bayes and have grouped them together to make a voting classifier using sklearn.ensemble.

Here is the accuracy for the Voting Classifier.
Accuracy on training set : 0.95

Accuracy on test set : 0.9166666666666666

We have used ‘hard’ as a rule for the final output which gives the answer as the majority class.The comparison between the voting classifier and the all the other models is that the accuracy of the AdaBoost model , XGBoost model,LGBM model and voting classifier is almost the same there is no such big difference between them on the testing dataset while the Gaussian model performs very low in comparison to all the other models. Coming on the training set part, the XGBoost model overfit the training dataset that's why its accuracy on training dataset is very high. However, the voting classifier does not have that gaussian model's effect that much on it because it is following the majority rule so its accuracy is close to all the other models.