PRML LAB 3

Harsh Sharma (B20CS017)

Question 1

Preprocessing and data splitting

- In preprocessing , all the rows containing nan values have been removed .
- For categorical encoding features having dtype equal to object are encoded using label encoder .
- For normalization of data MinMaxScaler is used .
- Data is splitted into 80:20 ratio in which 80 is training and 20 is testing .

Decision Tree Regressor

a) Cost Function

- Cost is calculated by mse that is mean squared error .
- Information gain is calculated by subtracting the weighted sum of the mse of the children after splitting from the parent gini.

b) Best_Splitting

- In the best_split() function the dataset is splitted into children that are left and right subtree and parent node .
- A threshold value is calculated on the basis of maximum information gain .
- On the basis threshold value left and right subtree are created.

c) Training

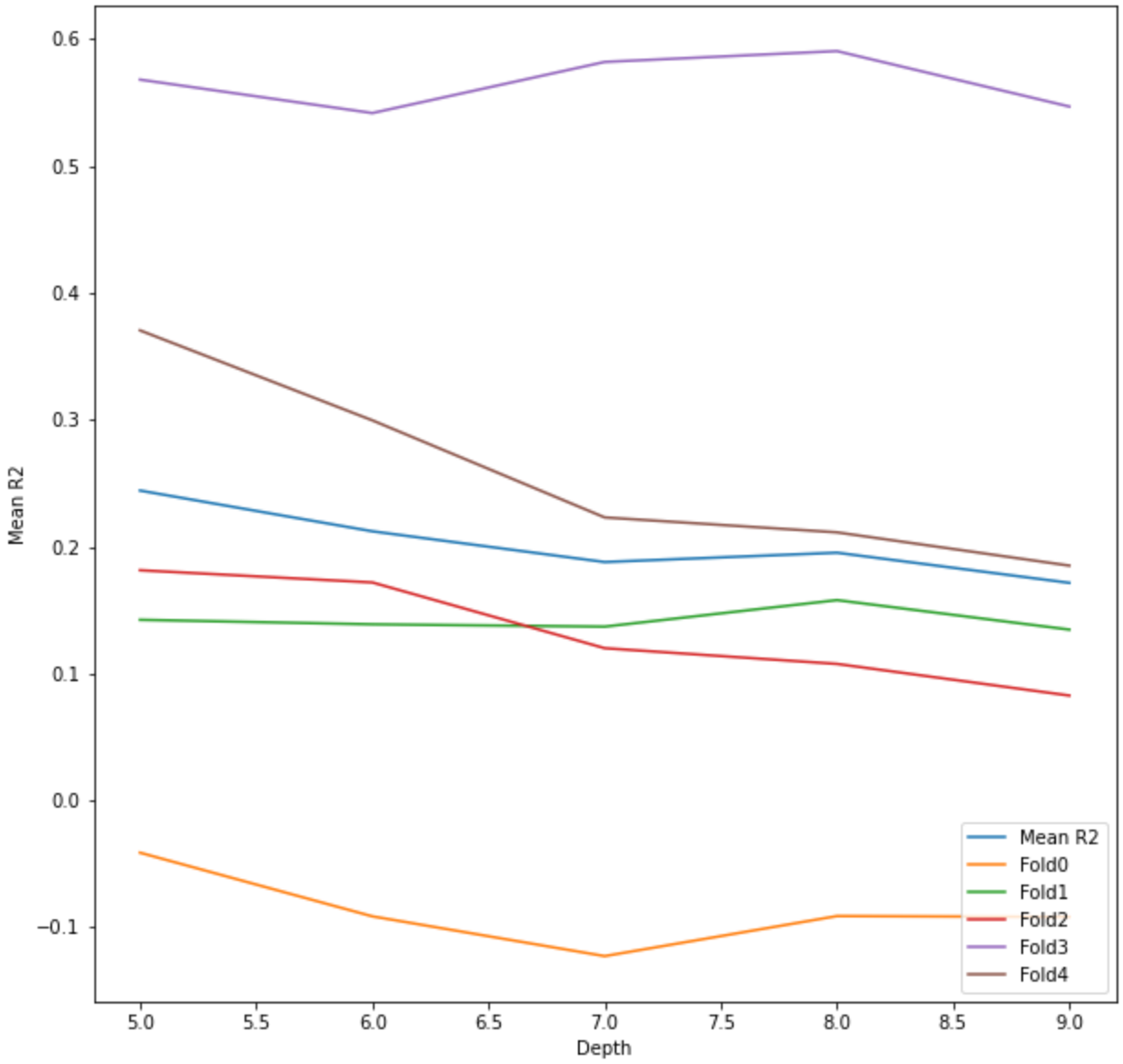
- This is recursively done by calling best_split() function on a training dataset of size 80 percent of actual dataset until regression is completed, that is the mse becomes zero or maximum depth is reached or min samples split is reached to create a decision tree .
- When max depth and min samples reach a certain value or there is no information gain, the tree stops growing and the node becomes a leaf node, with the value of the leaf node determined by average value .

d) Accuracy

- Predicted values are compared to the actual output of testing data to obtain accuracy .
- The R2 squared score of the model obtained is equal to 0.5235649367 .

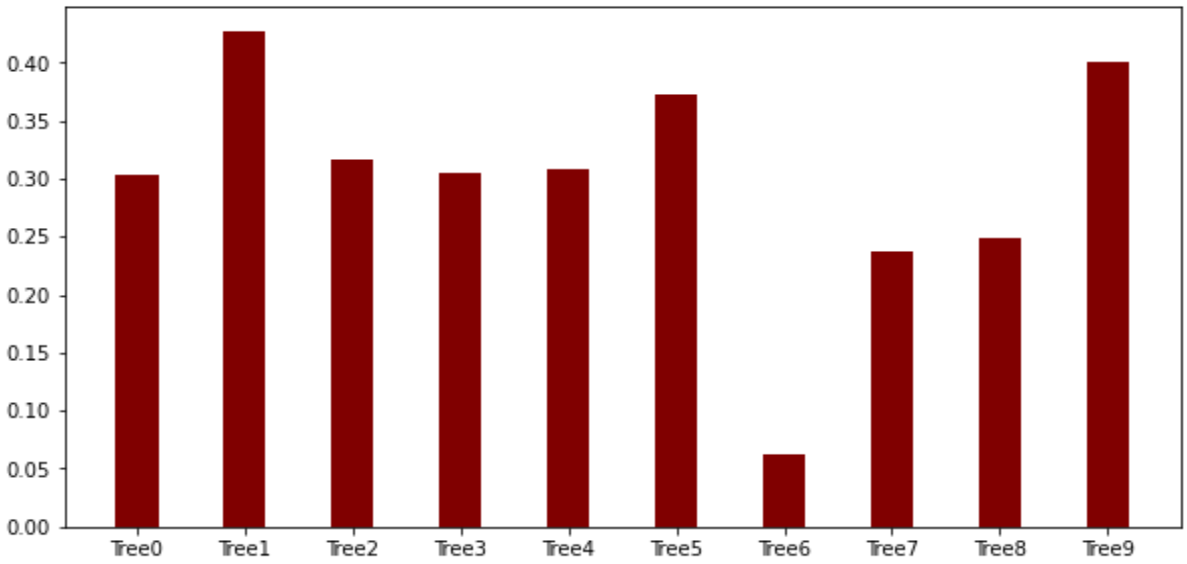
Fivefold Cross Validation

- The training data is divided into five distinct datasets, each of which is exclusive and exhaustive.
- The DecisionTreeRegressor is trained on four partitions and then tested on the fifth.
- This is done five times more until each partition has served as testing data at least once.
- The optimal value of max depth for which the model offers the best mean R-squared score is determined using this technique.
- For the housing dataset the optimal depth that was obtained was 8 with R2 score equal to 0.5904181740932697 .
- The mean R-squared score, as well as the R-squared scores for each of the five-fold validation folds, are shown in the graph below.



BAGGING

Bagging creates 10 datasets by randomly picking training data and replacing it. Each of these datasets is used to train the model, which results in the creation of ten decision tree regressors. The testing data is used to validate the models.The average R2 score of all 10 trees obtained is equal to 0.2980620555436034 . The outcome of the 10 trees is shown in the diagram below.

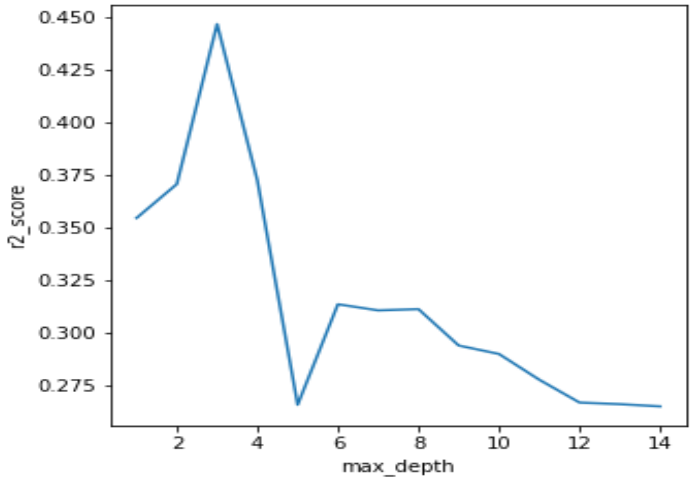


BAGGING

The trees generated via bootstrap aggregation are then utilized to build one model in the ensemble of trees. This model's prediction is made by running the data through all of the trees and averaging their predictions.The model is evaluated using an R-squared score and it was found to be 0.6299390145303544 .

DEPTH VS R2 SCORE

On changing depth close to optimal depth that we obtained in above part we observe an increase in the R-squared score as shown below.



.RandomForestRegressor

Random forest regressor is implemented using sk learn library .The mean squared error obtained is 0.008148866299111865 .

AdaBoostRegressor

AdaBoostRegressor is imported from the library and is implemented.The mse and r2 score obtained are 0.01237186920391076 and 0.44429709166009346 respectively .

Question 2

Preprocessing and data splitting

- In preprocessing , all the rows containing nan values have been removed .
- For categorical encoding features having dtype equal to object are encoded using label encoder .
- For normalization of data MinMaxScaler is used .
- Data is splitted into 80:20 ratio in which 80 is training and 20 is testing .

Decision Tree Classifier

a) Cost Function

- Cost function is calculated by using the gini index which is calculated by subtracting the sum of the squared probabilities of each class from one .
- It favors larger partitions.
- For a perfectly classified gini index would be zero .
- Information gain is calculated by subtracting the weighted sum of the gini index of the children after splitting from the parent gini.

b) Best_Splitting

- In the best_split() function the dataset is splitted into children that are left and right subtree and parent node .
- A threshold value is calculated on the basis of maximum information gain .
- On the basis threshold value left and right subtree are created.

c) Training

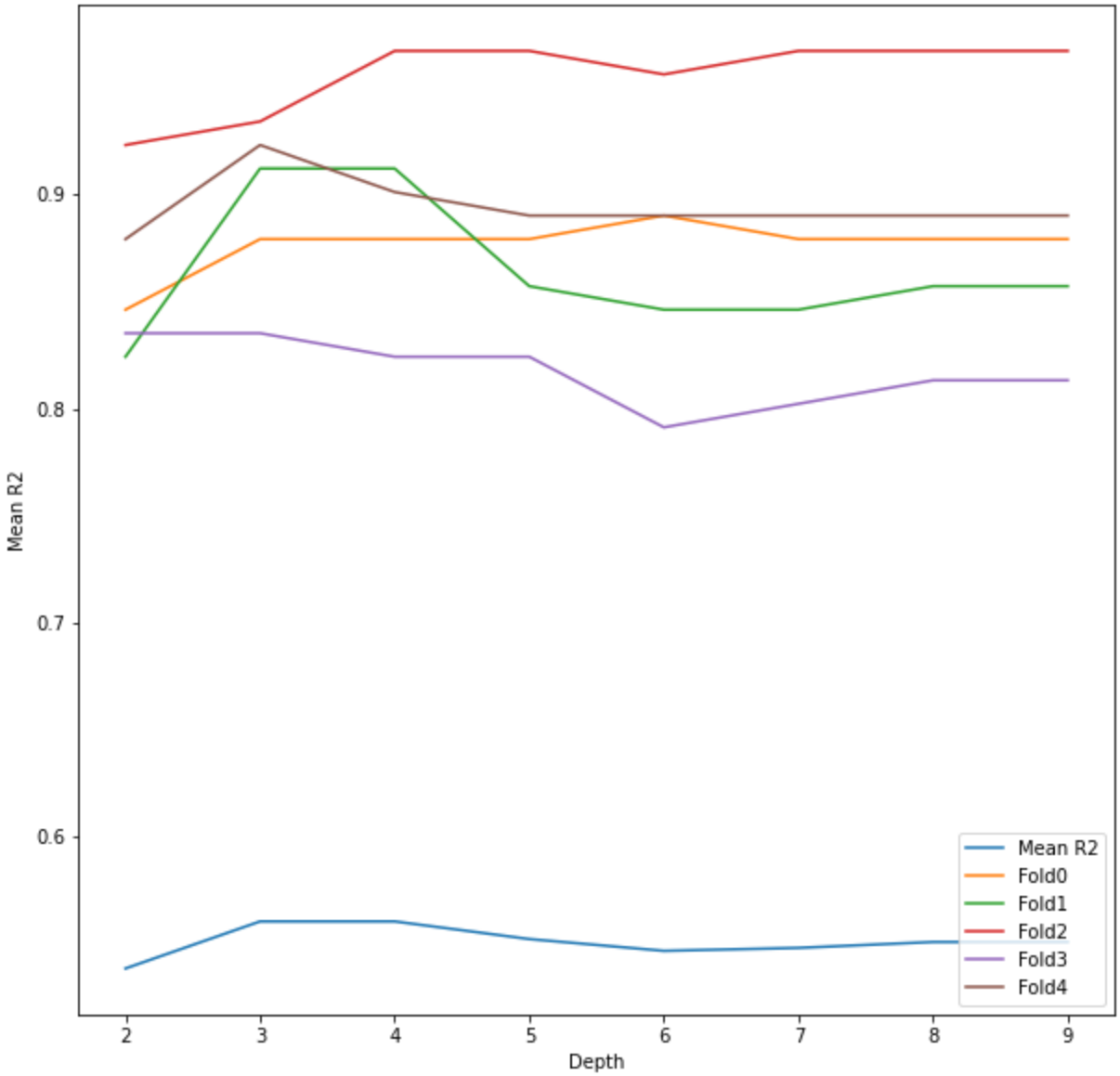
- This is recursively done by calling best_split() function on a training dataset of size 80 percent of actual dataset until classification is completed, that is the gini index becomes zero or maximum depth is reached or min samples split is reached to create a decision tree .
- When max depth and min samples reach a certain value or there is no information gain, the tree stops growing and the node becomes a leaf node, with the value of the leaf node determined by voting.

d) Accuracy

- Predicted values are compared to the actual output of testing data to obtain accuracy .
- The accuracy of the model obtained is equal to 0.9210526315789473.

Fivefold Cross Validation

- The training data is divided into five distinct datasets, each of which is exclusive and exhaustive.
- The DecisionTreeClassifier is trained on four partitions and then tested on the fifth.
- This is done five times more until each partition has served as testing data at least once.
- The optimal value of max depth for which the model offers the best mean accuracy is determined using this technique.
- For the housing dataset the optimal depth that was obtained was 4 with accuracy equal to 0.967032967032967 .
- The mean accuracy , as well as the accuracy for each of the five-fold validation folds, are shown in the graph below.



XGBoost

XGBoost is imported from the library and is implemented with parameters subsample=0.7 and max_depth=4. It is trained on the training dataset and the accuracy is found on both testing data. Accuracy on Testing Data obtained is 0.9298245614035088 .

LightGBM

LightGBM is imported from the library and is implemented with parameters max_depth=3 and num_leaves was varied. The accuracy observed on changing the parameter num_leaves is given below.

0.956140350877193

0.9385964912280702

0.9298245614035088

0.9473684210526315

0.9385964912280702

0.9473684210526315

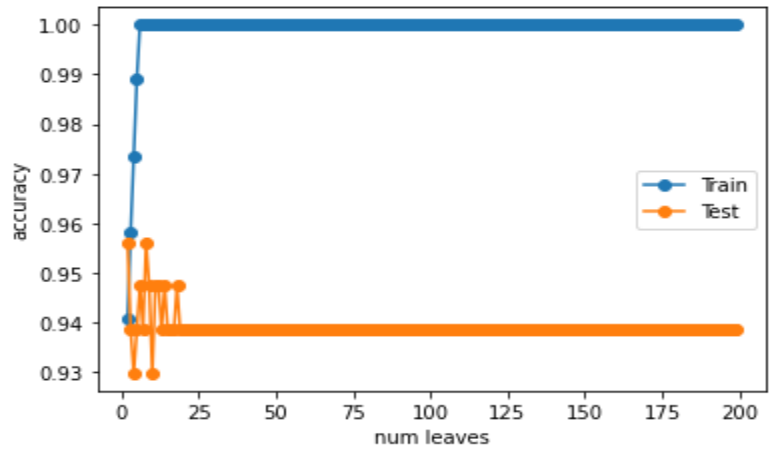
0.9385964912280702

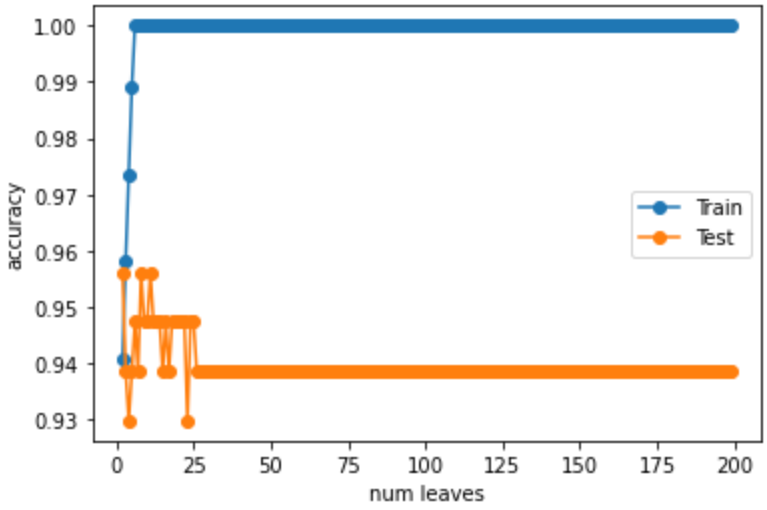
0.9385964912280702
0.9385964912280702

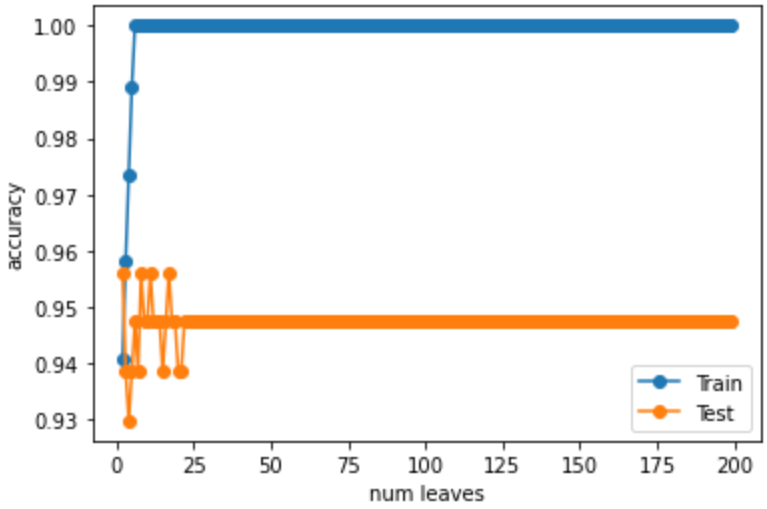
Analyzing the relation between max_depth and num_leaves

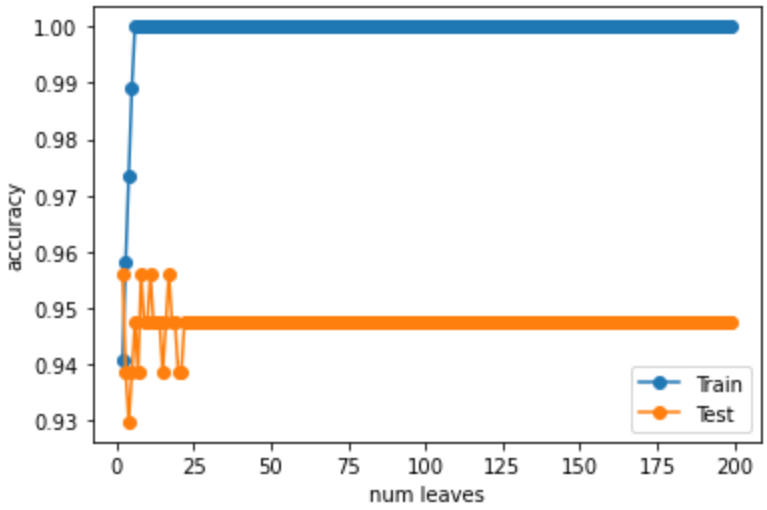
To examine these parameters and their relationships, max depth will be varied from 2 to 35, and num leaves will be modified from 2 to 200 for each max depth, with the accuracy of testing and training shown.

For some max_depths the graphs indicate overfitting of data, as we can see that as num leaves grow (X-axis of the graph), the accuracy (Y-axis) increases for training data but increases and subsequently drops for testing data. This indicates that the model has been overfitted to the training data and so performs badly on the testing data. While for some max_depth the graphs indicate no overfitting of data as we can see that in these graphs the model trains and testing and training accuracy increases together and then reaches a saturation level.




1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
accuracy
0
25
50
75
100
125
150
175
200
num leaves
Train
Test


1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
accuracy
0
25
50
75
100
125
150
175
200
num leaves
Train
Test


1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
accuracy
0
25
50
75
100
125
150
175
200
num leaves
Train
Test

Parameters can be used for better accuracy

a) **N_estimators** :- When n_estimators were varied and accuracy was observed then it can be concluded that accuracy increases with increased n_estimators and then becomes almost constant.

b) **Learning Rate: -** Another parameter named learning rate was varied and accuracy was observed.On increasing learning rate accuracy decreases .