Pattern Recognition and Machine Learning
2022 Winter Semester

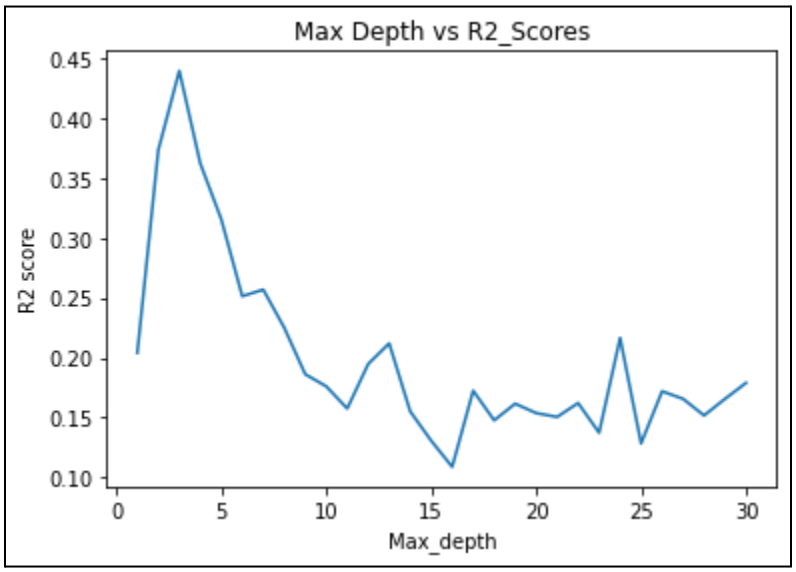
Report - Lab Assignment - 3

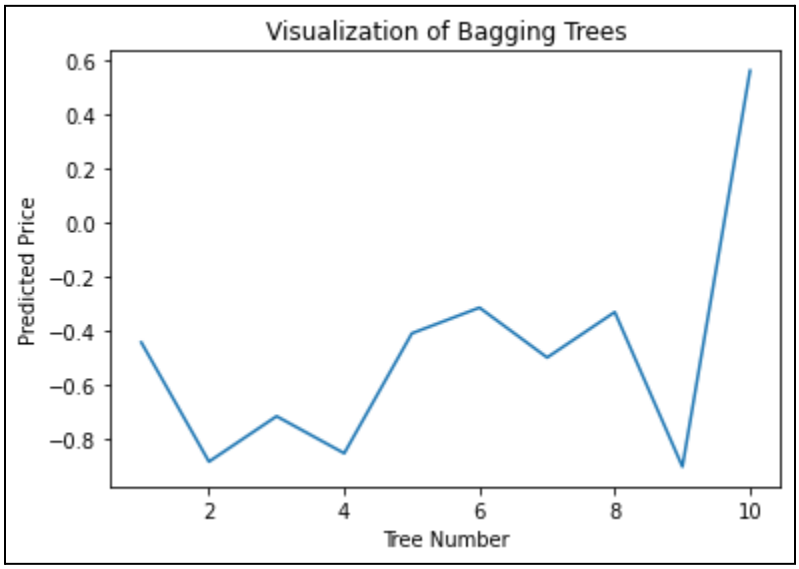
Question - 1

- Firstly, I Imported the necessary data manipulation libraries pandas and numpy.
- Then I created a DataFrame from the given Housing.csv file.
- Then, I created a visual plot of the data using the seaborn library.
- Then, I checked the data for null values and dropped an entire row if found any.
- After that, I encoded the features with object dtypes and updated the current DataFrame df.
- Afterwards, I splitted the data into Training and Testing set in a 70:30 ratio.
- Then, I used a DecisionTreeRegressor Class of sklearn.tree module to predict the labels and found out the accuracy as 47.37%.
- Then, after performing 5 fold cross-validation, I calculated the r2_scores at different max_depths and created a plot of max_depth vs the scores obtained corresponding to them.
- I found out the best max_depth = 3 for which the r2_score was maximum.
- Then, I created a bagging function with n_estimators = 10, that means it would create 10 different trees.
- In the bagging function, all the 10 different trees made different predictions and then I printed out the R2 Scores of all the different trees separately.
- Also, I plotted the graph of the tree number and its corresponding R2_score.
- After that, the average R2_score of all the different trees came out to be -0.47.
- Then, I created a function which combined all the trees into one using their weighted averages and I finally got the final predictions of the combined tree.
- The final R2_score of the combined trees came out to be 0.039.
- When I increased and decreased the max_depth by 2 units of earlier, I found out the R2 Scores as -0.002 and 0.0566 respectively.
- Then, I used the RandomForestRegressor Class of sklearn.ensemble module to make the label predictions and found out the following:
    - ❖ Mean Squared Error - 1276332056747.2378
    - ❖ Mean Absolute Error - 829701.5955
    - ❖ R2 Score - 0.63205

- Then, I used the AdaBoostRegressor Class of sklearn.ensemble module to make the label predictions and found out the following:
    - ❖ Mean Squared Error - 1629098192592.5369
    - ❖ Mean Absolute Error - 974926.992
    - ❖ R2 Score - 0.5303

Some Plots:

- 

- 

Question - 2

- Firstly, I Imported the necessary data manipulation libraries pandas and numpy.
- Also, I installed the xgboost and lightGBM libraries using
    - !pip install xgboost
    - !pip install lightgbm
- Then, I created a DataFrame df2 from the data “Breast_cancer_data.csv” and visualized the data using seaborn library.
- Then, I checked for the null values in the Dataframe and dropped the corresponding row if I found any null.
- Afterwards, I splitted the data into Training and Testing set in a 70:30 ratio.
- Then, I used a DecisionTreeClassifier Class of sklearn.tree module to predict the labels and found out the accuracy as 92.3976 %.
- Then, after performing 5 fold cross-validation, I calculated the r2_scores at different max_depths and created a plot of max_depth vs the scores obtained corresponding to them.
- I found out the best max_depth = 4 for which the r2_score was maximum.
- Then, I implemented xgboost classifier with subsample = 0.7 and max_depth = 4 and predicted the labels keeping n_estimators = 10 and found out the accuracy score on the training data as 0.8919 and 0.9356 on the testing set.
- Similarly, I implemented the LightGBM classifier with max_depth = 3 and predicted the models for different values of num_leaves from 2 to 14.
- I found out that at fixed max_depth = 3 and num_leaves = 8, the model started showing overfitting and accuracy score didn’t change further.
- Then, I fixed num_leaves = 8 and checked the accuracy for different values of max_depth from 2 to 14 and found out that at a maximum depth of 6 and for num_leaves = 8, the model started showing overfitting.
- Also, the parameters such as min_data_in_leaf, max_depth and num_leaves with proper tuning can give us a model with better accuracy and can avoid overfitting to a great extent.