Project 3: CS7646

Srinadh Raja Nidadana
snidadana3@gatech.edu

GT ID: 903966341

Abstract:

This report investigates decision tree learners' overfitting tendencies, the impact of bagging on model generalization, and a comparative analysis between Decision Tree Learner (DTLearner) and Random Tree Learner (RTLearner). Three experiments were conducted using *Istanbul.csv* to analyze RMSE trends, assess bagging effectiveness, and compare learners based on Mean Absolute Error (MAE). Findings highlight overfitting patterns, the mitigation potential of bagging, and insights into learner performance differences.

Introduction:

Decision tree models, particularly DTLearner, often exhibit overfitting when hyper parameters such as leaf size are not carefully tuned. Overfitting results in poor generalization, affecting prediction reliability. This study explores overfitting, examines how bagging influences generalization, and compares DTLearner with RTLearner to identify advantages and drawbacks. The primary hypothesis is:

- Overfitting increases for small leaf sizes in DTLearner
- Bagging mitigates overfitting but does not eliminate it entirely
- RTLearner may demonstrate improved generalization compared to DTLearner

Experiment 1: Overfitting Analysis

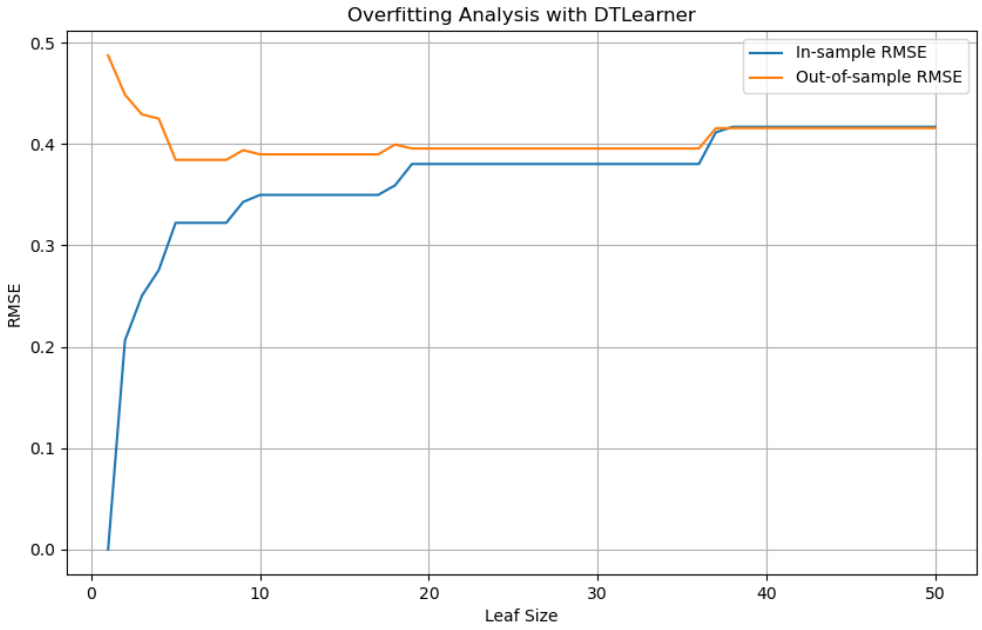
Question:

- Does overfitting occur with respect to leaf_size?
- For which values of leaf_size does overfitting occur? Indicate the starting point (a.k.a., optimal hyperparameter setting) and the direction of overfitting (e.g., the range over which overfitting occurs beginning with the optimal hyperparameter setting). Support your answer in the discussion or analysis. Use RMSE as your metric for assessing overfitting.
- Include a discussion of what overfitting is, why it occurs, why it is important, and how it is mitigated

Findings:

- The RMSE plot reveals that for small leaf_size values, in-sample RMSE is significantly lower than out-of-sample RMSE, indicating overfitting.
- Overfitting starts to diminish as leaf_size increases beyond 10, suggesting an optimal hyperparameter tuning range.
- Overfitting is evident for leaf_size < 10, where the in-sample error remains low while out-of-sample error stays high

Figure:



Experiment 1: Overfitting Analysis with DTLearner

Why Overfitting Occurs:

- Small leaf_size leads to complex trees that fit noise in training data
- Overfitting hinders generalization to new data.

How to Mitigate?

- Increasing leaf_size
- Using ensemble methods like bagging
- Pruning tree complexity

Experiment 2: Effect of Bagging on Overfitting

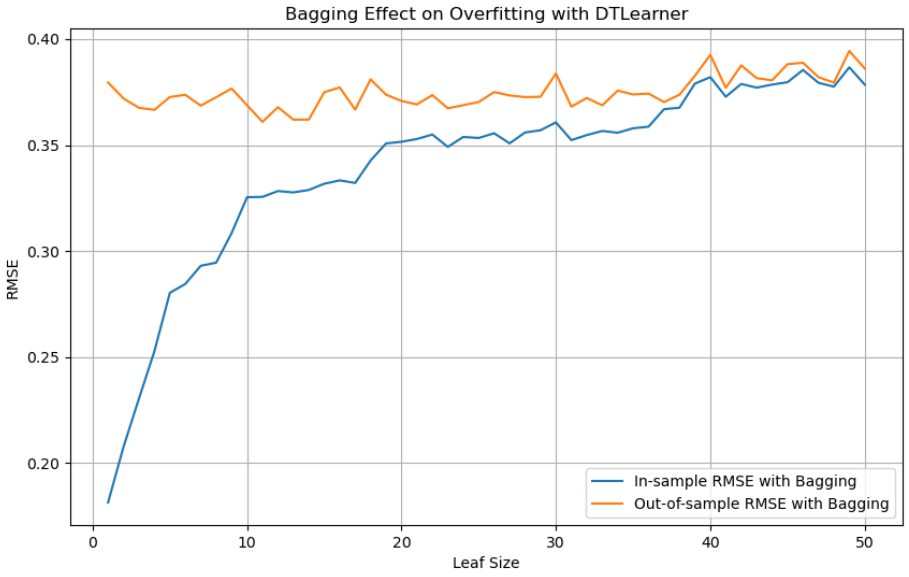
Question:

- Can bagging reduce overfitting with respect to leaf_size?
- Can bagging eliminate overfitting with respect to leaf_size?

Findings:

- Introducing bagging reduces overfitting by averaging multiple models, leading to more stable RMSE trends
- Bagging lowers RMSE variance but does not fully eliminate overfitting
- RMSE values for in-sample and out-of-sample converge better compared to the non-bagged model

Figure:



Bagging and its effect on overfitting using DTLearner

Key Observations:

- Overfitting remains noticeable for very low leaf_size
- Bagging enhances generalization without drastically altering optimal leaf_size

Conclusion:

- Bagging is effective but not a complete solution for overfitting
- Combining bagging with optimal leaf_size tuning yields the best performance

Experiment 3: Comparison of DTLearner and RTLearner

Question:

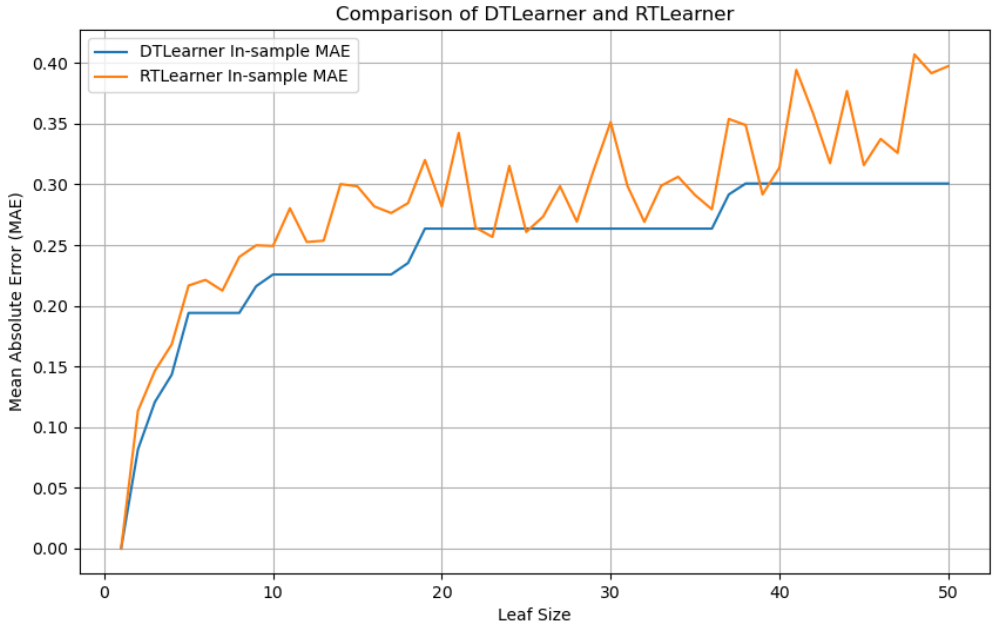
Provide at least two new quantitative measures in the comparison

- Using two similar measures that illustrate the same broader metric does not count as two separate measures. *(Note: Do not use two measures for the accuracy or use the same measurement for two different attributes – e.g., **time** to train and **time** to query are both considered a use of the "**time**" metric.)*
- Provide charts to support your conclusions

At a minimum, the following question(s) must be answered in the discussion.

- In which ways is one method better than the other?
- Which learner had better performance (based on your selected measures) and why do you think that was the case?
- Is one learner likely to always be superior to another (why or why not)?

Figure:



DTLearner Vs RTLearner

Findings:

- RTLearner exhibits higher variance in MAE compared to DTLearner
- DTLearner maintains relatively stable MAE trends, indicating better consistency
- RTLearner is more sensitive to leaf_size variations
- The standard deviation of errors is higher in RTLearner, suggesting less reliable predictions compared to DTLearner

Performance Analysis:

- DTLearner performs better in terms of predictive consistency
- RTLearner provides more variability, which can be useful in diverse datasets
- Neither method is universally superior; choice depends on use case

Summary:

This study highlights the following key insights:

- Overfitting in DTLearner occurs at small leaf_size values and can be controlled by increasing leaf_size
- Bagging helps in mitigating overfitting but does not completely remove it
- DTLearner and RTLearner offer trade-offs in terms of consistency and adaptability
- DTLearner shows more stable and predictable performance, while RTLearner offers higher adaptability but with increased prediction variance

In Future we can explore alternative ensemble strategies, such as boosting, and analyze additional performance metrics to refine model selection strategies.