CS7646 Project 3

Petros Venieris
pvenieris3@gatech.edu

1 ABSTRACT

This project investigates decision tree-based learning algorithms, including DTLearner, RTLearner, BagLearner, and InsaneLearner. DTLearner selects features based on correlation, while RTLearner chooses features randomly. Experiments explore overfitting, the effect of bagging, and model performance across various leaf sizes, using RMSE as a metric. Insights into model complexity, generalization, and ensemble learning are discussed.

2 INTRODUCTION

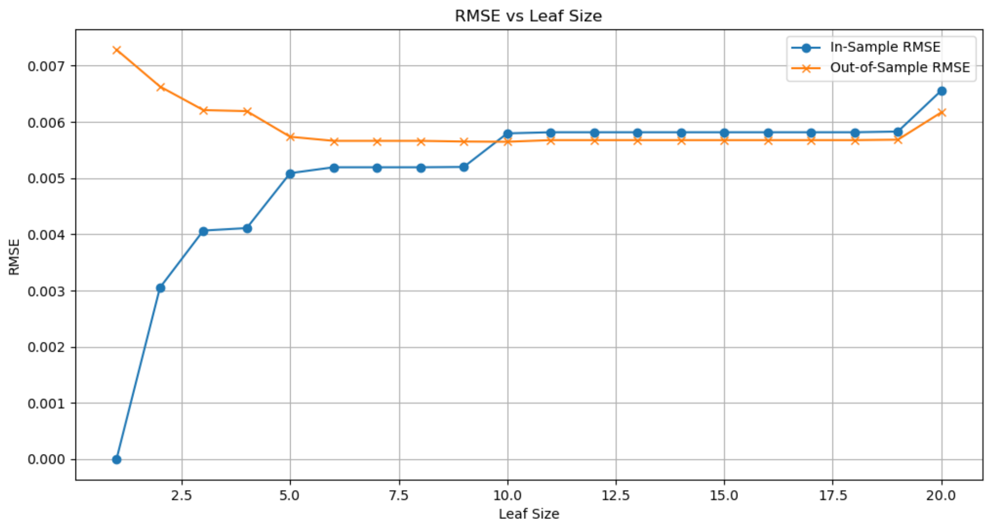
In this report, we investigate the performance of various decision tree-based learning algorithms, including DTLearner, RTLearner, BagLearner, and InsaneLearner, with a focus on their application in regression tasks. Decision trees, such as DTLearner, which selects features based on correlation with the target variable, and RTLearner, which introduces randomness by selecting features randomly, are foundational machine learning methods. We extend our study to ensemble techniques, specifically BagLearner and InsaneLearner, to evaluate how aggregating multiple learners can improve model performance and generalization. BagLearner builds on the idea of reducing variance by averaging predictions from several learners, while InsaneLearner pushes this concept further, using an extreme number of learners to explore the limits of bagging. Throughout our experiments, we utilize the testlearner script to automate data processing, assess performance using root mean squared error (RMSE), and analyze overfitting with respect to leaf size. The experiments are designed to assess whether ensemble learners, particularly BagLearner, can yield lower RMSE and more robust predictions than individual learners, confirming the hypothesis that bagging strategies enhance predictive performance and reduce error.

3 METHODS

In our investigation, we assess various decision tree algorithms, namely DTLearner, RTLearner, BagLearner, and InsaneLearner, utilizing the Istanbul dataset. The dataset underwent manual preprocessing and was divided into training and test subsets to ensure precise control over the data management process. DTLearner constructs decision trees by selecting the feature with the highest absolute correlation with the target variable, whereas RTLearner employs a stochastic feature selection approach. BagLearner combines multiple decision trees trained on resampled data to bolster model stability, while InsaneLearner amplifies this by incorporating a greater number of base models to explore the boundaries of ensemble methods. We utilized the test learner script for systematic evaluation, measuring model performance through root mean squared error (RMSE) on both training and testing datasets. To investigate overfitting, we plotted RMSE against leaf size when verbose was True ,allowing for detailed examination of tree construction and error diagnosis. This thorough analysis aims to elucidate the advantages and limitations of these algorithms. Let us now proceed with the experiments and review the outcomes.

4 EXPERIMENT 1

I Will start by showing the picture of RMSE of DTLearner versus leafe size , and I will talk about results later. We have to answer many questions regarding overfitting based on our metric.

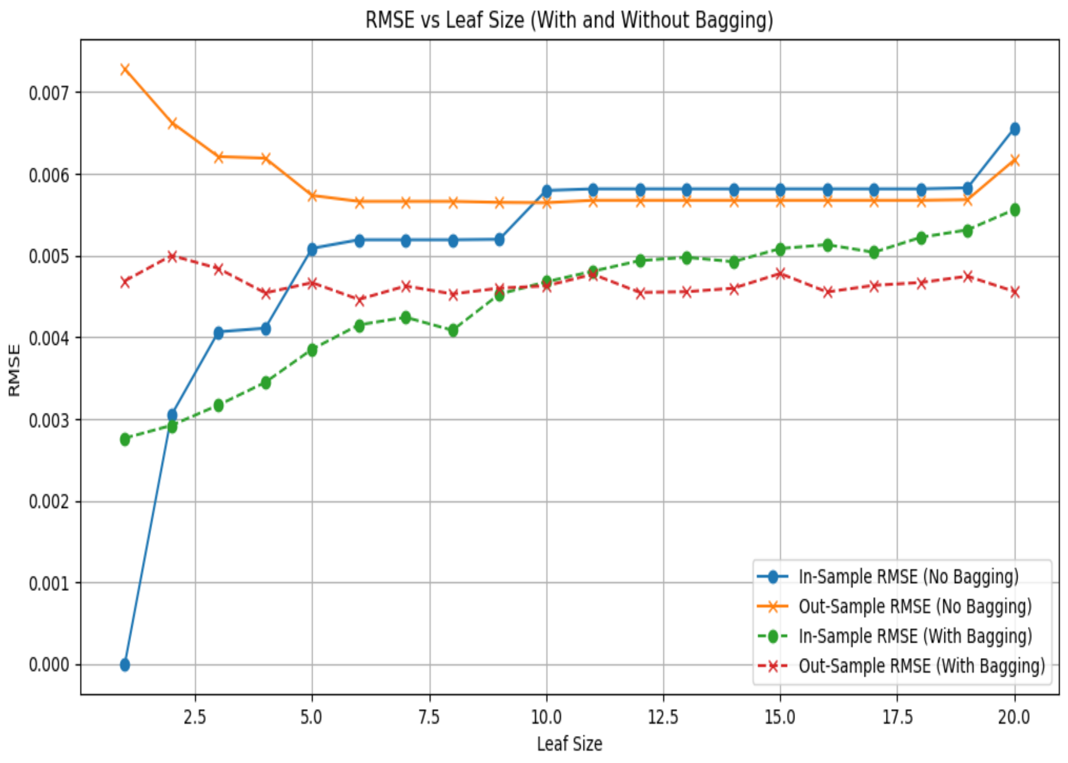


Let's start by explaining overfitting: Overfitting is characterized by a model that performs exceptionally well on the training data but fails to generalize to new, unseen data, which is often evident through poor performance on out-of-sample data. This occurs when the model becomes too complex, fitting not only the true patterns but also the noise in the training set. It is crucial to detect overfitting because it results in a model that does not reliably predict future data. The extent of overfitting can be assessed using root mean squared error (RMSE) for both in-sample and out-of-sample evaluations. A significant discrepancy, where in-sample RMSE is much lower than out-of-sample RMSE, indicates that the model has overfit the training data.

On this exact situation we can see that for big number of leafs , the RMSE for both in sample and out sample are close indicating a good model that neither overfit or underfit on the training data. From 10 leafs to 5 we see just a small change but still the numbers (0,006 to 0,005) are insignificant. **From leaf 5** and down we see a huge increase in out of sample error and a huge decrease in in sample error , this indicates us that our model start overfit to our training data and start lose value on out of sample data. Despite all of these results our overall RMSE is pretty small overall and indicate a good model but it is suggested to use leaf size >5 to avoid overfitting.

5 EXPERIMENT 2

Experiment 2 is asking us to use the bagging effect and check wherether this helps with overfitting , I will start again by showing a graph:

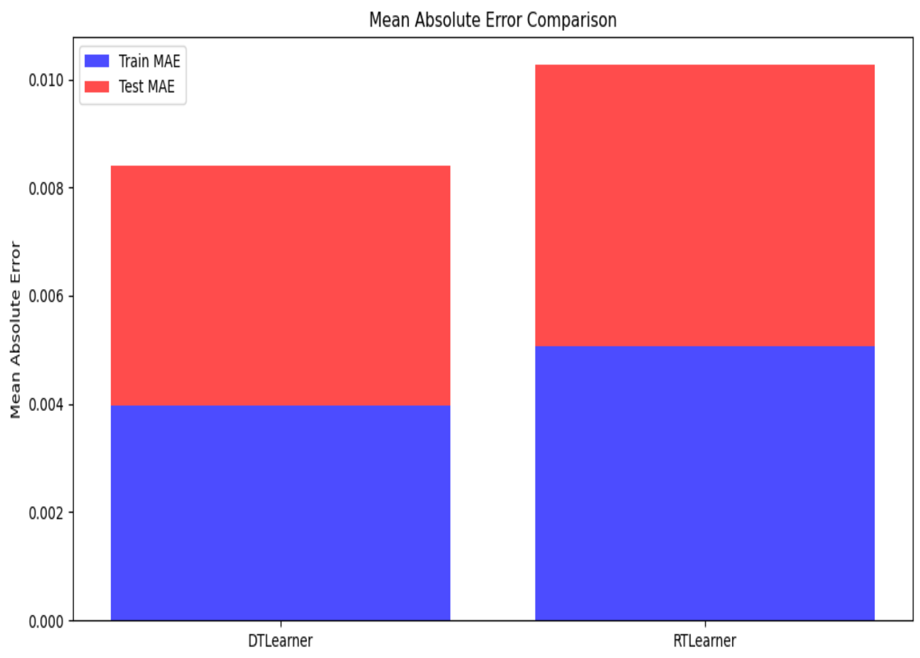


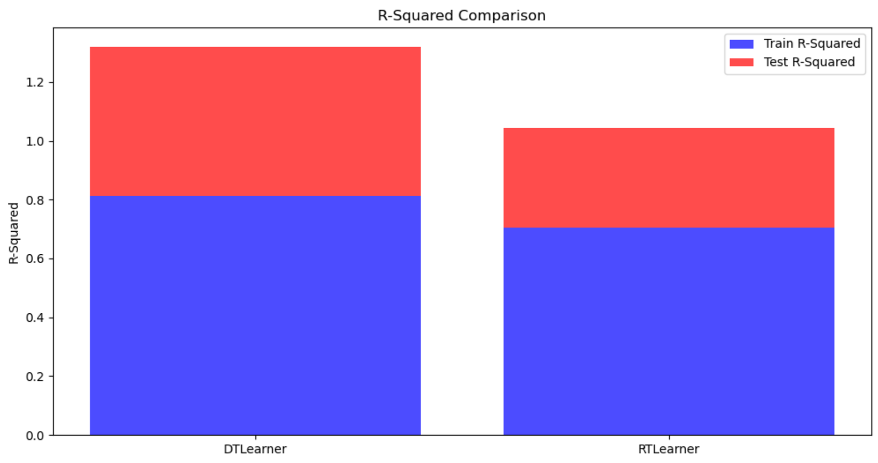
We can see our previous plot , in sample and out sample RMSE without bagging and then we can see them with bagging. The first noticeable difference and the most obvious one is that YES , bagging reduce overfitting. Out of sample RMSE is constant and steady from start to finish and in sample RMSE is closer to out of sample RMSE in a small range. No , it's not eliminating overfitting since we can see that after leaf 10 , we have a steady decrease in in sample RMSE until it reaches leaf size one while on the other hand Out of sample RMSE stays the same or almost the same around 0.005 RMSE. So even if the overfitting is pretty small(0,003 to 0,005) it still exist , but bagging make it so small it could be considered not important.

Also another useful insight is that the overall RMSE is always reduced with bagging(cause of the variance reduction).

6 EXPERIMENT 3

The third experiment is basically a comparitoin experiment , a DTLearner versus LTLearner experiment that want us to understand the difference between those 2 and which one perfom better here. I will comparing them using Rsquare and Mean absolute error metrics. I will start again with the plots:





We can see that DTLearner achieve both a lower MAE error and a higher R square on both train and test groups. That indicates that DTLearner might be making predictions that are generally closer to the true values on average, resulting in a lower MAE. DTLearner might also be capturing a significant portion of the variance in the target variable, which translates into a higher R-Squared value. This indicates that the model fits the data well in terms of variance explanation. These metrics show us that DTLearner results in more detailed and good predictions taking into account just some information about our data (highest absolute value correlation with Y) instead of doing random picks. So DTLearner will probably results in better results, but another metric that we haven't taking into consideration is time , I haven't data since the project ask us to don't take into consideration the time for out comparison but the best value of the RTLearner is that because it randomly choose without need for calculation it it faster than DTLearner , so in a dataset with millions of data we should probably see a big difference in time needed to run the algorithms. So the final answer on which Learner is the best , it depends on what our current goal is and what is needed , but DTLearner will probably make more accurate and trustworthy preddictions.

7 SUMMARY

This research explored decision tree learning using DTLearner and RTLearner, focusing on overfitting, the impact of bagging, and comparing traditional and random trees. In Experiment 1, we observed that smaller leaf sizes in DTLearner led to overfitting, while larger sizes mitigated it but sometimes caused underfitting. Experiment 2 demonstrated that bagging reduced overfitting by averaging multiple trees, although it didn't completely eliminate it, and allowed for more flexible leaf size choices. In Experiment 3, DTLearner outperformed RTLearner in terms of Mean Absolute Error (MAE) and R2, but RTLearner showed advantages in robustness and speed due to its random feature selection, making it less prone to overfitting but less accurate. Future investigations could explore hybrid models or advanced bagging strategies to further optimize performance.