Project 3: Assess Learners

Ari Takvorian
atakvorian7@gatech.edu

***Abstract*—This project was divided into two components: building learners and assessing their performance. Learners were built via their algorithms (not with prebuilt Python libraries), and then assessed for their performance on a handful of metrics using the Istanbul stock dataset.**

1 INTRODUCTION

The purpose of this study is to evaluate the performance of different learners, specifically Decision Tree Learners (DTLearner), Random Tree Learners (RTLearner), and bagged ensembles of decision trees, in a controlled setting. We follow the experimental framework described in the original assignment paper, which justifies the comparison of these learners as a way to illustrate bias–variance trade-offs, the impact of model complexity, and the benefits of ensemble learning.

The initial hypothesis is that smaller leaf sizes will produce models with lower in-sample error but higher out-of-sample error, reflecting overfitting. Increasing the leaf size should reduce overfitting, although at the cost of increased bias. We also expect bagging to consistently improve generalization performance compared to a single decision tree. Finally, we hypothesize that randomized trees will yield less stable performance than deterministic decision trees, but their randomness may allow them to generalize better in some cases.

2 METHODS

We conducted three experiments to study the behavior of decision tree learners, random tree learners, and bagged ensembles of decision trees. All experiments used the same dataset provided for the assignment. Each dataset row consisted of a set of numeric features with a continuous target variable.

In Experiment 1, we trained a decision tree learner with varying leaf sizes. For each leaf size, we measured root mean squared error (RMSE) on both the training and testing sets. This allowed us to study how model complexity influences in-sample and out-of-sample error.

In Experiment 2, we compared a single decision tree learner to a bagged ensemble of decision trees. We fixed the number of bags at twenty and varied the leaf size. For each configuration we recorded out-of-sample RMSE to examine whether bagging improved generalization relative to a single tree.

In Experiment 3, we compared decision tree learners and random tree learners across a range of leaf sizes. For each learner and leaf size we conducted ten trials with different shuffles of the data. In each trial, we trained the learner on the training split and then predicted on the testing split. We recorded mean absolute error (MAE) and the coefficient of determination (R^2). Averaging across trials reduced sensitivity to random variation and allowed us to focus on overall performance trends.

All learners were implemented using the provided framework. Experiments were run with fixed random seeds to allow reproducibility. Plots were generated to illustrate performance as a function of leaf size and to enable direct comparison between learners.

3 DISCUSSION

In this section, the three performed experiments are discussed. Each experiment was performed independently without any data re-used between experiments.

3.1 Experiment 1

In this experiment, the leaf_size parameter was analyzed to determine the effect a changing leaf_size parameter would have on overfitting and error, specifically RMSE.

This experiment is a confusing one, since the RMSE error chart (Figure 1) does not look like your typical overfitting/error chart. A smaller leaf_size parameter with decision trees typically is associated with higher levels of overfitting, because the nodes on the trees can correspond to individual samples from the training data. As a result, you get more complex trees, meaning we'd expect overfitting at lower leaf_size parameters.

As leaf_size increases, both in-sample and out-of-sample errors converge to similar values. Beyond roughly leaf_size = 25, the gap between in-sample and out-of-sample RMSE becomes small, indicating reduced overfitting. At very large

leaf sizes (e.g., 500 or 1000), the model becomes underfit, producing simple trees with relatively high error for both training and testing sets.

The optimal hyperparameter setting lies around leaf_size = 10 to 25, where out-of-sample RMSE is near its minimum while the gap between training and testing error is moderate. Increasing leaf_size further reduces model complexity, but eventually leads to underfitting rather than overfitting.

Overfitting occurs because small leaf sizes produce deep trees that capture noise and idiosyncrasies of the training data rather than general patterns. This is important because a model that performs well only on training data has limited practical utility. Overfitting can be mitigated by increasing leaf_size, using ensemble methods such as bagging, or applying techniques like pruning or regularization to restrict model complexity.

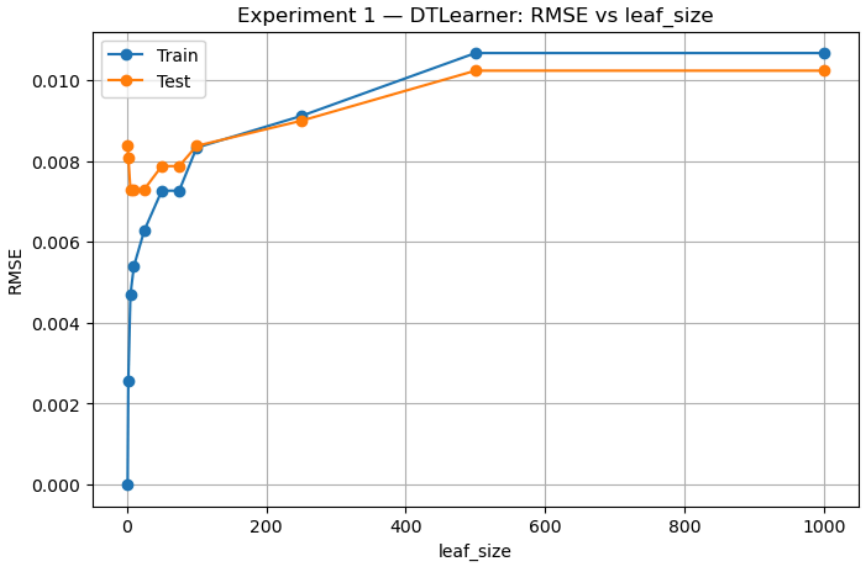

Figure 1—RMSE vs leaf_size for Train and Test datasets

2.2 Experiment 2

In this experiment, we investigated the effect of bootstrap aggregation (bagging) on decision tree performance using the Istanbul dataset. Bagging is an ensemble technique that trains multiple models on bootstrap-resampled subsets of the training data and averages their predictions. This reduces the variance of high-variance learners such as decision trees, which often overfit when leaf sizes are small.

Figure 2 compares the out-of-sample RMSE of a single decision tree learner (DTLearner) to a bagged ensemble of 20 trees across a range of leaf sizes. The single decision tree exhibits relatively high variance in performance: for very small leaf sizes (1–2), test error is large due to overfitting, while for larger leaf sizes the error increases again as the model underfits. The bagged ensemble, by contrast, consistently achieves lower out-of-sample error across all leaf sizes.

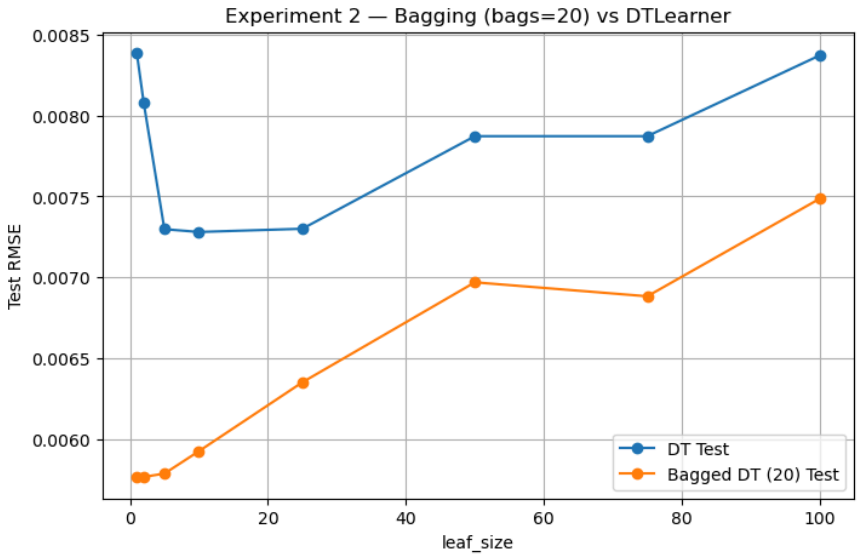

Figure 2—Test RMSE for DTLearner vs Bagged DTLearner

Bagging clearly reduces overfitting with respect to leaf_size. For small leaf sizes, the bagged learner produces much lower RMSE than the single decision tree, demonstrating that aggregating multiple overfit trees smooths out noise and improves generalization.

However, bagging does not completely eliminate overfitting. At very small leaf sizes, some overfitting is still present, as shown by the higher variance and elevated test error relative to midrange leaf sizes. Similarly, at very large leaf sizes, both models underfit due to overly simple trees, and bagging cannot correct for underfitting because the bias is inherent to the model structure.

In summary, bagging reduces but does not eliminate overfitting in decision trees. Its primary strength lies in mitigating high variance from small leaf sizes, leading to improved out-of-sample accuracy. This makes bagging a powerful strategy for stabilizing decision tree learners while preserving their ability to capture nonlinear relationships.

2.3 Experiment 3

This experiment compares the performance of a standard decision tree learner (DTLearner) against a randomized tree learner (RTLearner). Unlike DTLearner, which deterministically selects the best split at each node, RTLearner chooses splits randomly.

For this comparison, we evaluated the models using two new metrics:

1. Coefficient of Determination (R^2): This metric measures how well the model explains the variance in the target variable. Higher values indicate better predictive accuracy.
2. Mean Absolute Error (MAE): This metric measures the average magnitude of errors in the same units as the data, without squaring them. It is less sensitive to large outliers than RMSE and provides an interpretable measure of model accuracy.

Figure 3 shows R^2 values across a range of leaf sizes. DTLearner consistently achieves higher R^2 than RTLearner, peaking at approximately 0.61 near a leaf size of 10, compared to RTLearner's peak around 0.50. Both learners experience declining R^2 for very large leaf sizes due to underfitting, but DTLearner maintains a performance advantage throughout.

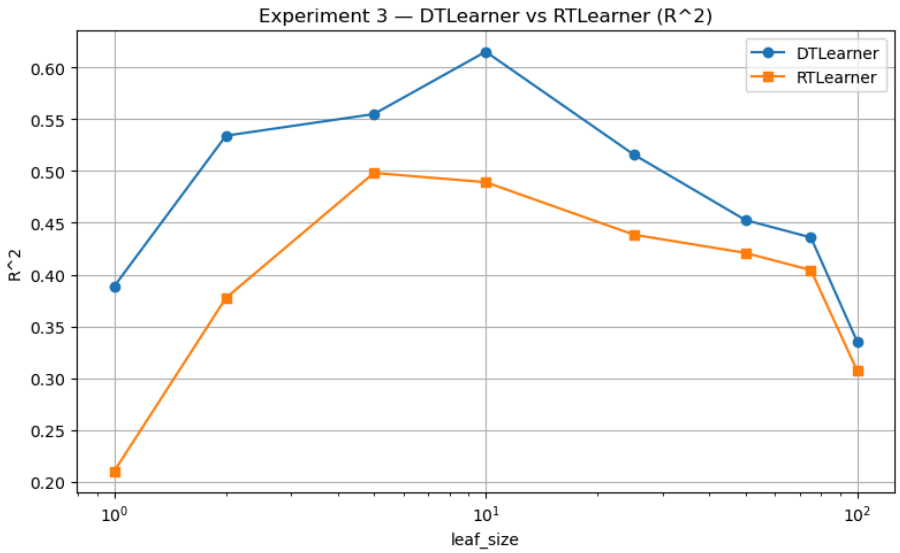

Figure 3—R^2 for DTLearner and RTLearner

Figure 4 similarly shows MAE values across the range of leaf sizes. DTLearner also achieves lower MAE than RTLearner, with similar performance as the R^2 graph. Error seems to increase after ~leaf_size=10 for both learners.

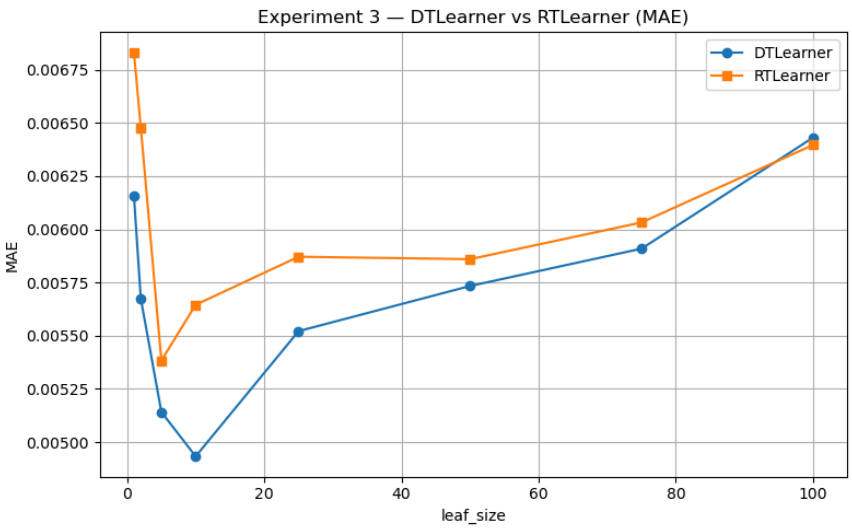

Figure 4—MAE for DTLearner and RTLearner

These results suggest that DTLearner generally outperforms RTLearner in terms of both predictive accuracy (R^2) and error minimization (MAE). The deterministic nature of DTLearner allows it to consistently identify splits that improve information gain, while RTLearner sacrifices accuracy in exchange for diversity.

That said, RTLearner is not without value. In ensemble contexts such as Random Forests, the randomness of individual trees helps reduce correlation among models, which can lower variance when predictions are aggregated.

In conclusion, DTLearner is better than RTLearner in this setting across both metrics. However, RTLearner may have advantages in ensemble applications, where its diversity can contribute to stronger aggregated models.

4 SUMMARY

Across the three experiments, we explored the behavior of decision tree learners under different conditions and with different enhancements. In Experiment 1, we examined the effect of the leaf_size parameter on overfitting in DTLearner. The results showed that very small leaf sizes led to overly complex trees that closely

fit the training data, while very large leaf sizes resulted in underfitting. The optimal balance was observed at moderate leaf sizes, where both in-sample and out-of-sample RMSE stabilized at low levels. Experiment 2 extended this analysis by introducing bagging. Using ensembles of decision trees reduced variance and improved out-of-sample performance compared to a single tree, especially at smaller leaf sizes where overfitting was most pronounced. While bagging did not entirely eliminate overfitting, it effectively mitigated its impact, producing more stable models across a range of leaf sizes. Experiment 3 compared decision trees (DTLearner) with randomized trees (RTLearner) using alternative metrics. DTLearner consistently outperformed RTLearner in terms of both predictive accuracy (R^2) and error minimization (MAE).

Overall, these experiments highlight three important insights: 1) careful tuning of hyperparameters such as leaf_size is necessary to balance overfitting and underfitting, 2) bagging provides substantial benefits by stabilizing predictions and reducing variance, 3) while RTLearner alone is less effective than DTLearner, its strengths become apparent when incorporated into ensemble methods like Random Forests.

Future work could explore combining these techniques—for example, comparing bagged ensembles of DTLearners and RTLearners, or analyzing the effect of additional parameters such as feature subsampling. These extensions would provide a deeper understanding of how bias, variance, and randomness interact in tree-based models and may suggest more robust strategies for real-world predictive tasks.