Learning Method Overfitting Assessment

Author Name
sspickard3@gatech.edu

Abstract*—This report investigates overfitting in machine learning using decision tree-based learners on stock return data. We evaluate how leaf size impacts overfitting and test whether bagging can reduce it. We also compare performance trade-offs between decision trees and random trees using alternate metrics. Bagging reduces overfitting and decision trees generally outperform random trees in predictive accuracy but take longer to train.

1 INTRODUCTION

Overfitting occurs when a model fits the data too closely, capturing noise instead of the underlying signal[1]. This leads to poor generalization and degraded test performance. In this study we explore how decision tree-based learners behave under varying tree complexity (leaf size), and how ensemble techniques such as bagging influence overfitting. Our hypothesis are: (1) smaller leaf size increases overfitting in decision trees, and (2) our bagging reduces overfitting across learners. We also compare decision trees and random trees using alternative performance metrics to assess trade-offs between accuracy and efficiency.

2 METHODS

We evaluated overfitting behavior using three learner implementations. The first was a standard Decision Tree Learner, which recursively split on the feature most correlated with the response variable and then used the median value of that feature as the split point. The second was a Bagged Decision Tree Learner, an ensemble method that trains multiple Decision Tree instances on different bootstrap samples of the training data and averages their predictions. Ten bags were used, each trained on 60% of the training data sampled with replacement. The third was a Random Tree Learner, which randomly selected a feature for each split while still using the median value of that feature as the threshold. All learners used a leaf size as a hyperparameter to control the minimum number of data points allowed in a leaf size.

Our experiments used the Istanbul.csv dataset, which includes daily returns of major global stock indices from Jan 5, 2009 and Feb 22, 2011. The target variable was the MSCI Emerging Markets Index (EM), and the predictor variables included daily returns from indices such as S&P 500, DAX, FTSE 100, and others. The date column was excluded from modeling.

Table 1—Variables included in the

Name	Description
date	Date of the observation, excluded from the analysis
ISE-TL	Return of the Istanbul Stock Exchange (ISE) in Turkish Lira
ISE-USD	Return of the ISE in USD
SP	Return of the S&P 500 (USA)
DAX	Return of the DAX Index (Germany)
FTSE	Return of the FTSE 100 Index (UK)
NIKKEI	Return of the Nikkei 225 Index (Japan)
BOVESPA	Return of the BOVESPA Index (Brazil)
EU	Return of the Euro Stoxx 50 (Europe)
EM	Return of the MSCI EM Index (Emerging Markets)

The dataset contained 536 observations. We randomly split the data into 60% for training and 40% for out of sample testing. Although temporal ordering is typically preserved in financial time series, our primary objective was to analyze overfitting behavior, not predictive validity over time, and thus random split was acceptable for these experiments.

3 DISCUSSION

This section presents the results of three experiments designed to evaluate overfitting behavior in decision tree–based learners. Experiment 1 investigates the effect of leaf size on overfitting in a single decision tree. Experiment 2 evaluates whether bagging mitigates overfitting in decision trees. Experiment 3 compares the performance of decision trees and random trees using non-traditional evaluation metrics. For each experiment, we provide visualizations and analysis of key trends and performance trade-offs.

3.1 Experiment 1

The goal of Experiment 1 was to investigate overfitting behavior in a decision tree learner without the use of bagging. Our hypothesis was that smaller leaf sizes would lead to overfitting, as the tree would become too complex and tailor itself to noise in the training data.

We varied the minimum leaf size from 1 to 20 and evaluated performance using in-sample and out-of-sample root mean squared error (RMSE) and correlation. As shown in Figure 1, when the leaf size is set to 1, the in-sample correlation for i is close to 1.0, indicating that the model fits the training data extremely well. However, the out-of-sample is lower (around 0.7), and the out-of-sample RMSE is high, suggesting poor generalization.

As the leaf size increases, the model becomes simpler, learning to a reduction in overfitting. Both in-sample and out-of-sample metrics begin to converge and the out-of-sample performance flattens around leaf size of 5. This suggests that smaller leaf sizes produce overly complex models that do not generalize well, while larger leaf sizes encourage simpler models that better capture the underlying signal.

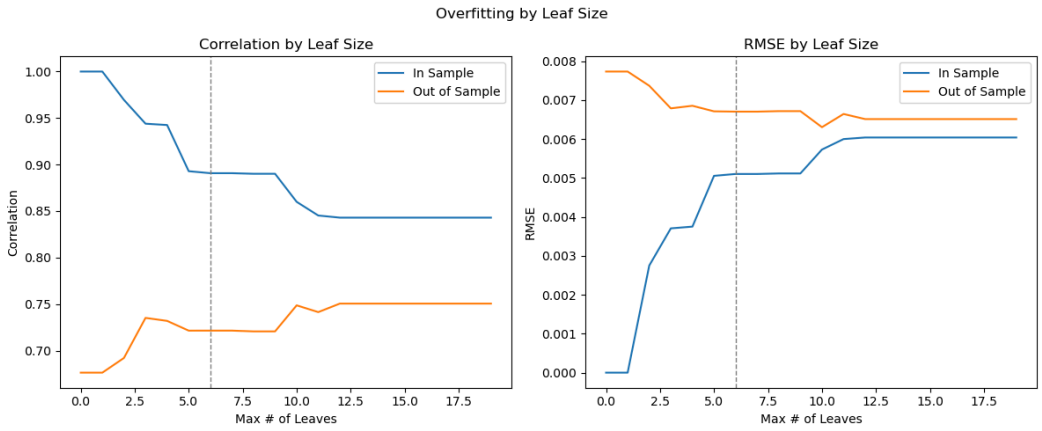

Figure 1—Correlation and RMSE as Leaf Size varies from 1 to 20 in an unbagged decision tree model.

This has important real-world implications. If a user relied solely on in-sample performance when predicting future returns of the Emerging Market Index, they might deploy an overly confident model that performs poorly in production.

This highlights the importance of using test data and diagnostic plots to select appropriate model complexity.

Overfitting can be mitigated through methods such as cross-validation, early stopping, and ensemble learning techniques like bagging [2] which are explored in the next experiment.

3.2 Experiment 2

In Experiment 2, we examined whether bagging can reduce or eliminate overfitting in decision trees. Bagging, or bootstrap aggregating, is an ensemble technique that trains multiple models on random samples (with replacement) of the training data and averages their predictions. This reduces model variance and can improve generalization by smoothing out overfitting-prone behavior from individual learners.

We repeated the procedure from Experiment 1 using a Bagged Decision tree Learner composed of 10 bags, each trained on 60% of the training data. We again varried the leaf size from 1 to 20 and evaluated in-sample and out-of-sample correlation and RMSE. Results are shown in Figure 2.

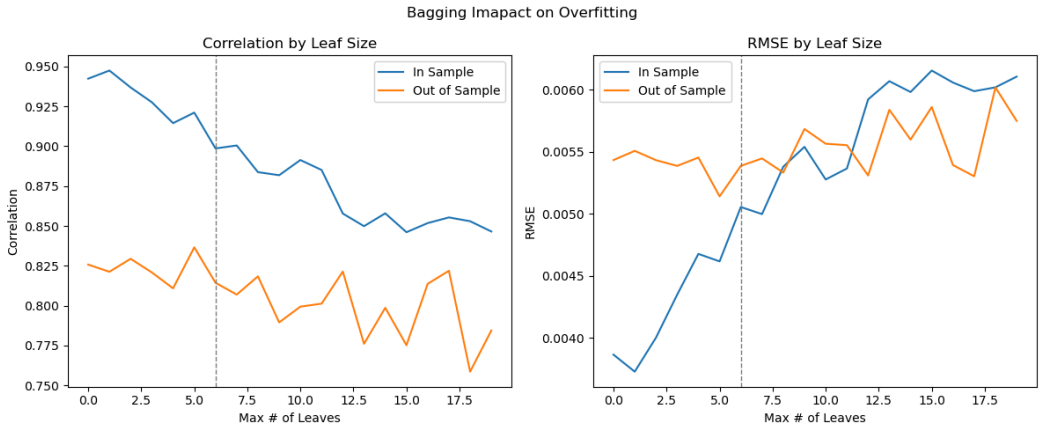

Figure 2—Correlation and RMSE as Leaf Size varies from 1 to 20 in a bagged decision tree model.

Compared to the unbagged learner, the bagged model exhibited much flatter performance curves across all leaf sizes. The gap between the in-sample and out-of sample correlation is narrower, and RMSE values are more stable for out of sample. While some overfitting is still present at very small leaf sizes, its

effects are muted relative to the single tree case. This suggests that bagging successfully reduces overfitting but does not entirely eliminate it.

Interestingly, there is no clear "elbow" point in the performance metrics, making it harder to visually identify an optimal leaf size. This smooth behavior implies that the ensemble buffers the model from sharp overfitting transitions.

In summary, bagging reduces overfitting by stabilizing model predictions across leaf sizes. It does not completely eliminate overfitting, especially at extreme values like leaf size = 1, but it clearly improves generalization performance compared to a single tree.

3.3 Experiment 3

In Experiment 3, we compared the performance of Decision Tree learners and Random Tree learners using two alternative evaluation metrics: training time and prediction accuracy measured by Mean Absolute Error (MAE) and R-Squared. This experiment used newly run data and did not reuse the results from Experiments 1 or 2.

As shown in Figure 3, both models demonstrate the expected overfitting at low at low-leaf size, with higher variance in performance across the leaf size range. The Decision Tree model shows a smaller MAE and higher R-Squared values and higher stability in out-of-sample predictions compared to the Random Tree. In contrast, the Random Tree model exhibits more erratic behavior, likely due to the randomness in its feature selection.

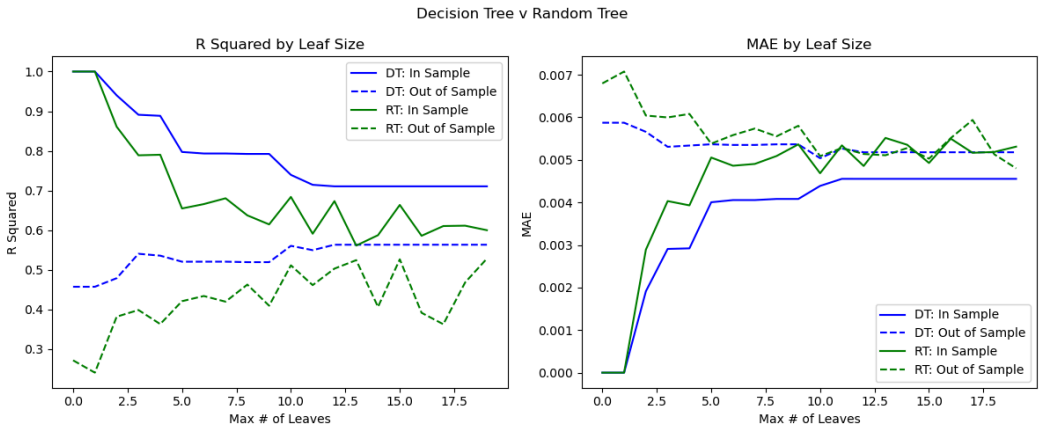

Figure 3—R-Squared and MAE as Leaf Size varies from 1 to 20 in a decision tree model compared to random tree model.

Figure 4 presents training and prediction times. Here, the random tree significantly outperforms the Decision tree in terms of speed, especially at lower leaf sizes. Training time for the Random Tree is several orders of magnitude lower at leaf size 1. Prediction times for both models are comparable.

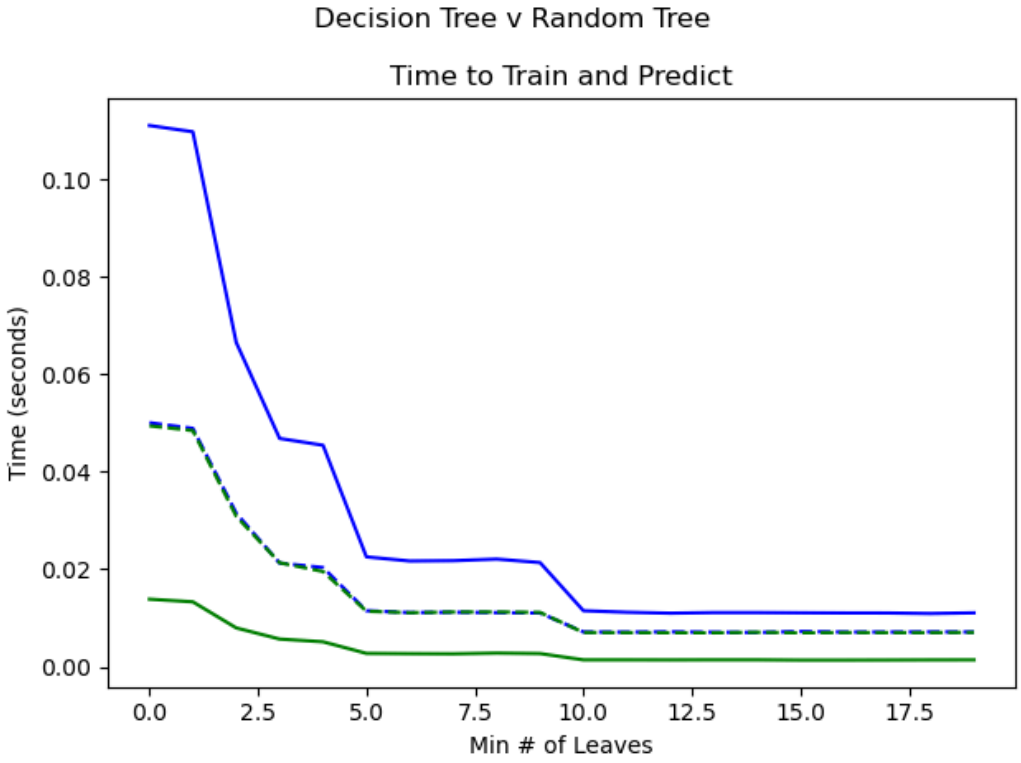

Figure 4—Training Time as Leaf Size varies from 1 to 20 in a decision tree model compared to random tree model.

These results suggest a trade-off between accuracy and training time. The Decision Tree offers more reliable and accurate predictions while the Random Tree trains much faster and may be preferable when computational resources or time are constrained.

Overall, we found that Decision Trees outperformed Random Trees in predictive accuracy, but Random Trees were superior in training speed. No model was universally better across all conditions. In contexts where interpretability, stability, and predictive power are prioritized, Decision Trees are the better

choice. However, in time-sensitive or resource constrained environments, Random Trees may offer a practical advantage.

4 SUMMARY

This report explored overfitting in decision-tree based learners and evaluated strategies to mitigate it. In Experiment 1, we observed that overfitting occurs in Decision Trees when the leaf size is small, starting when leaf size was one and tapered off when the leaf size was 5 or higher.

In Experiment 2,we tested whether bagging could reduce overfitting. The bagged decision trees exhibited more stable performance across all leaf-sizes, with reduced variance and smaller gaps between in-sample and out-of-sample metrics. While bagging did not eliminate overfitting completely, it clearly mitigated its effects and reduced sensitivity to the leaf size hyperparameter.

Experiment 3 compared decision trees with random trees using alternative metrics. Decision trees produced more accurate and consistent predictions, while random trees trained significantly faster. The decision tree was better suited for applications requiring accuracy and stability, whereas the random tree offered practical advantages in time-sensitive or resource-constrained environments.

Overall, we found that model complexity, ensemble methods, and leaner architecture impact overfitting and performance trade-offs. Ensemble learning like bagging can stabilize variance prone learners, while decision choices such as randomized splitting introduce robustness at the cost of precision. Selecting the right learner depends on the context: accuracy, generalization, and speed often trade off against each other.

Future work could explore the effects of cross validation, pruning, or using bagging with random trees. It would also be valuable to test these learners on different datasets that do not exhibit linear relationships to assess robustness under more challenging conditions.

4 REFERENCES

1. James, G., Witten, D., Hastie, T., Tibshirani, R., & Vollmer, S. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.
2. GeeksforGeeks. (2023, April 17). *How to avoid Overfitting in Machine Learning?* GeeksforGeeks.

https://www.geeksforgeeks.org/how-to-avoid-overfitting-in-machine-learning/