Project 3: Assess Learners

Samuel Wagner
swagner38@gatech.edu

Abstract — This study explores decision tree models, focusing on overfitting, bagging efficacy, and classic versus random trees. Hypotheses suggest overfitting susceptibility with suboptimal leaf_size. Bagging mitigates overfitting, while random trees offer a less computationally expensive alternative. Experiments with Istanbul.csv validate these hypotheses, optimizing decision tree models.

INTRODUCTION

This report examines decision tree models, focusing on overfitting, bagging efficacy, and the comparison between classic and random trees. Despite their interpretability, concerns about overfitting persist, especially when parameters like leaf_size are suboptimal. Bagging techniques may alleviate overfitting by aggregating predictions. Random tree models, with randomized feature selection, may offer a less computationally expensive alternative. Through experiments, this study aims to validate these hypotheses and optimize decision tree models for diverse applications.

METHODS

In three experiments, we investigated decision tree models using the Istanbul.csv dataset with DTLearner and RTLearner. Experiment 1 examined overfitting concerning leaf_size variations, while Experiment 2 explored the effect of bagging on overfitting by adjusting bag counts alongside leaf_size. Through RMSE analysis and charts, we assessed the impact of these parameters on model performance. In Experiment 3, we quantitatively compared classic decision trees with random trees, introducing new metrics for evaluation. The experiments aimed to understand the trade-offs and performance disparities between different decision

tree approaches, providing insights into overfitting mitigation strategies and the comparative advantages of classic versus random tree learners.

DISCUSSION

EXPERIMENT #1: OVERFITTING

Leaf size, or the minimum number of samples required to make a leaf node, can impact the occurrence of overfitting in decision-tree models. Setting a smaller leaf size may increase the likelihood of overfitting by allowing the tree to capture noise and anomalies in the data. Conversely, a larger leaf size can help mitigate overfitting by encouraging the model to make more generalizable splits based on larger subsets of the data. As is shown in Figure 1 below, the results of this experiment confirm this pattern. As leaf size decreases from 10, the in-sample error decreases while the out-of-sample error increases, indicating overfitting. However, as leaf size increases from 10, the in-sample and out-of-sample errors both increases, indicating that the overfitting subsides.

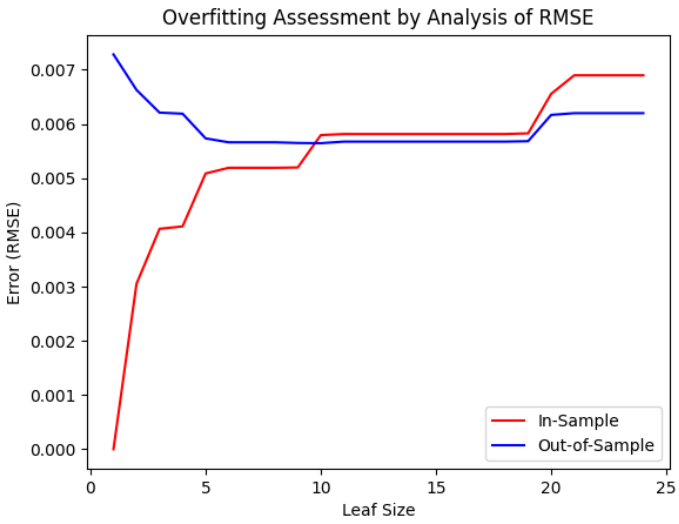

Figure 1 - Comparison of in-sample and out-of-sample errors to assess overfitting

However, leaf size alone does not entirely determine whether overfitting occurs; it interacts with other parameters such as the maximum depth of the tree and the minimum number of samples required to make a split.

Addressing overfitting in decision-tree models often involves employing strategies like pruning, which involves removing unnecessary branches to simplify the model and prevent it from memorizing noise in the training data. Moreover, constraints on the tree's depth and the minimum number of samples required for splits can help control the model's complexity and mitigate overfitting tendencies. Additionally, ensemble techniques like bagging can effectively combat overfitting in decision-tree models. Bagging, short for bootstrap aggregating, involves training multiple decision trees on different subsets of the training data and aggregating their predictions. By introducing randomness and diversity into the ensemble, bagging reduces the risk of overfitting by preventing individual trees from memorizing the training data too closely. The combined predictions of multiple trees tend to be more robust and generalizable than those of a single tree, thereby enhancing the model's overall performance and reducing the likelihood of overfitting.

EXPERIMENT #2: BAGGING

Bagging, also known as Bootstrap Aggregating, is a potent technique in ensemble learning aimed at mitigating overfitting and enhancing the stability of machine learning models, particularly decision trees. In the realm of decision tree models, bagging functions by generating multiple subsets of the original dataset through random sampling with replacement. Each subset is then employed to train an independent decision tree model. Through this process, bagging enables each decision tree to grasp different facets of the data due to the variations introduced by random sampling.

The primary advantage of bagging, especially with decision trees, lies in its ability to alleviate (but not entirely eliminate) overfitting with respect to leaf size. Overfitting occurs when a model excessively learns the intricacies of the training data rather than capturing the underlying patterns or relationships. Decision trees are prone to overfitting, particularly when they become too deep and capture noise or outliers present in the training data.

By producing numerous decision trees from bootstrapped samples of the original dataset, bagging diminishes the variance of the model. While each tree may still

overfit to some extent on its data subset, the ensemble model, by combining the predictions of multiple trees, tends to generalize more effectively to unseen data. This is achieved through averaging or aggregating the predictions of individual trees, which helps alleviate the noise and errors inherent in individual models. Figure 2, below, shows this affect by comparing the in-sample and out-of-sample errors of decision tree models with bagging and without bagging. With the initial gap between each set of in-sample and out-of-sample errors representing the amount of overfitting that is occurring, the plot shows that when bagging is incorporated, the overfitting is reduced.

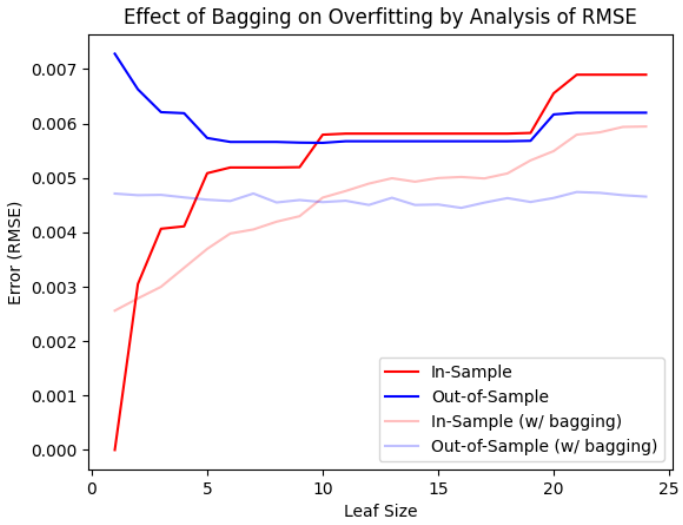

Figure 2 - Plot of the effect of bagging on overfitting

Additionally, bagging enhances model robustness by decreasing sensitivity to outliers and noise in the training data. Since each decision tree is trained on a different subset of the data, outliers and noise may exert less influence on the overall ensemble prediction. Consequently, this contributes to the creation of a more dependable and stable model that performs well on new, unseen data.

EXPERIMENT #3: DECISION TREES VS RANDOM TREES

A random tree model can potentially be more accurate than a normal decision tree due to its inherent ability to handle feature randomness during the splitting process. While a single decision tree tends to split nodes based on the most

discriminative features at each step, it might overlook relevant features that are less dominant in the dataset. In contrast, a random tree model introduces randomness in the feature selection process, even without bagging, by considering only a random subset of features at each split. This randomness encourages diversity among the trees in the model and allows for the exploration of different feature combinations, thereby capturing more nuanced patterns and reducing the risk of overfitting to specific features present in the dataset. By promoting variability in the decision-making process, random tree models without bagging can enhance the model's ability to generalize to unseen data and potentially improve predictive accuracy compared to a classic decision tree. This phenomenon is illustrated in Figure 3 below, especially at the smaller leaf sizes.

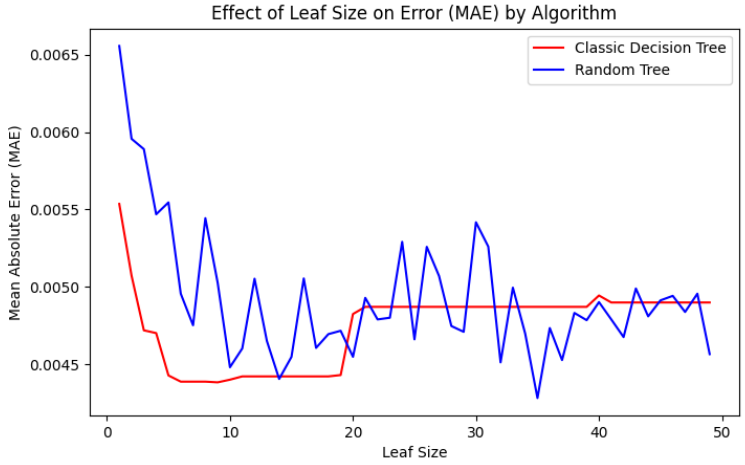

Figure 3 - The effect of leaf size on MAE by tree algorithm

Additionally, a random tree model without bagging can be less computationally expensive than a normal decision tree due to its reduced complexity and exploration of a smaller subset of features at each split. In a traditional decision tree, each node considers all available features to determine the optimal split, which can become computationally intensive, especially with large datasets or high-dimensional feature spaces. However, in a random tree model without bagging, the feature selection process is constrained to a random subset of features at each node, significantly reducing the computational burden. By limiting the number of features considered during splitting, the model reduces the search

space and computational overhead associated with evaluating potential splits, making the training process more efficient. Consequently, random tree models without bagging offer a computationally lightweight alternative to traditional decision trees while still maintaining the capacity to capture complex relationships within the data. This phenomenon is illustrated in Figure 4 below, especially at the smaller leaf sizes.

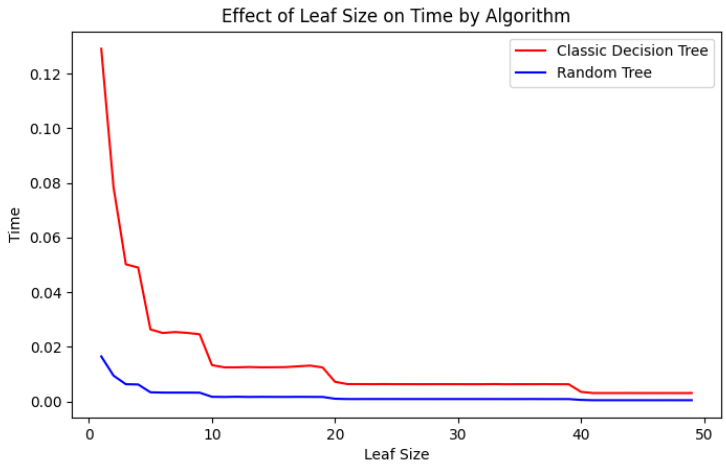

Figure 4 - The effect of leaf size on computational time by tree algorithm

It's improbable that one type of decision tree model will consistently outperform another across all datasets and contexts. Decision tree models, including basic decision trees, random forests, and gradient boosting machines, each offer unique advantages and trade-offs. For instance, while basic decision trees provide interpretability and simplicity, they may struggle with capturing complex relationships in the data compared to ensemble methods like random forests or boosting techniques. However, random forests and boosting methods tend to be more robust against overfitting and often yield higher predictive accuracy, particularly in high-dimensional or noisy datasets. The effectiveness of each decision tree model hinges on factors like dataset size, feature complexity, and the presence of interactions or outliers. Consequently, the superiority of a decision tree model is contingent on the specific characteristics of the dataset and the requirements of the task at hand.

SUMMARY

The report presents findings from experiments on decision-tree models, focusing on overfitting, bagging, and the comparison between decision trees and random trees. Overfitting occurs when models become overly complex, capturing noise instead of patterns; smaller leaf sizes exacerbate overfitting, while techniques like pruning and bagging mitigate it. Bagging reduces overfitting by aggregating predictions from multiple trees, enhancing stability and generalization. Random trees, with randomized feature selection, offer a less computationally expensive alternative to decision trees, capturing nuanced patterns and reducing overfitting risks. The experiments underscore the importance of addressing overfitting and highlight the effectiveness of ensemble methods like bagging and random trees in improving model robustness and efficiency, suggesting future investigations into more advanced ensemble techniques and hyperparameter exploration to further enhance model performance across various domains.