Project 3:
Assess Learners

Wai Kay Kok
wkok6@gatech.edu

Abstract—In this project, 4 types of Classification and Regression Trees (CARTs) are built to predict emerging market stock returns from other indices. They are the classic decision tree (DT Learner), random tree (RT Learner), Bag learner and Insane Learner. The effect of leaf size and bagging on overfitting are studied using the DT Learner and Bag Learner. DT Learner and RT Learner are compared on their prediction performance and training speed.

1 INTRODUCTION

Classification and Regression Trees (CARTs) is a family of tree-based machine learning algorithms for predicting discrete categorical or continuous numerical values. Within this family, there are various forms of algorithms that vary in complexity from basic decision tree to random forest that combines sampling technique bootstrapping and ensemble approach. CARTs have several advantages (James et al., 2023). Firstly, they are non-parametric methods. They do not require data to assume a normal distribution. They work with data of any distribution. Secondly, they are capable of modelling non-linear relationships, thus making them suitable for data where the relationship between features and response variables are non-linear. Thirdly, they are robust to outliers. Removal of outliers is not necessary with CARTs. Lastly, CARTs are easy to understand and interpretable. Despite these strengths, CARTs do have their weaknesses (James et al., 2023). Although they are good at modelling non-linear relationships, they may struggle when such relationships become too complex. They are also vulnerable to overfitting to the training data, leading to poor prediction performance on unseen data, especially for trees that are too large. On the other hand, small trees could lead to underfitting, leading to poor predictive power. Hence, an optimal balance needs to be found to achieve optimal performance. This can be attained through hyperparameters tunning and techniques, such as bagging, boosting and ensemble approach. In this project, 4 different types of CARTs will be implemented and evaluated on their performance in predicting emerging market stock returns, with respect to overfitting. The Istanbul dataset is used. The 4 types are classic decision tree (DT Learner), random tree (RT Learner), decision tree with bagging (Bag Learner) and an ensemble learner comprising 20 Bag Learners (Insane Learner). Through these 4 types of CARTs, the role of leaf size, bagging technique and ensemble approach in mitigating overfitting will be demonstrated. The hypothesis to be tested is that leaf size tuning, bagging and ensemble approach minimize overfitting. The null hypothesis is that leaf size tuning, bagging and ensemble approach have no effect on overfitting.

2 METHODS

Classic decision tree (DT Learner), random tree (RT Learner), learner incorporating bootstrapping and ensemble approach (Bag Learner) and learner comprising an ensemble of Bag Learner (Insane Learner) are constructed as described in CS7646 Canvas Project 3 instruction. The base decision tree, on which all the other learners are built, are constructed according to JR Quinlan method (Quinlan, 1986).

2.1 Experiment 1

To determine the effect of leaf size on overfitting of DT Learner, the learner is trained to make predictions on both the In-Sample (training set) and Out-Sample (test set) at various leaf sizes ranging from 1 to 50. This is repeated for 10 independent iterations and the average RMSE for each leaf size is computed. In each iteration, a training set is randomly split from the dataset. The results of both In-Sample and Out-Sample are plotted and compared.

2.2 Experiment 2

To determine the effect of bagging on overfitting of DT Learner, the Bag Learner is trained to make predictions on both the In-Sample and Out-Sample at various leaf sizes ranging from 1 to 50. Bag number is set to 20. This is repeated for 10 independent iterations and the average RMSE for each leaf size is computed. In each iteration, a training set is randomly split from the dataset. Bootstrap samples of the same size are obtained from the training set. The results of both In-Sample and Out-Sample are plotted and compared. The Out-Sample results from both experiment 1 and 2 are also plotted to determine which has better prediction performance.

2.3 Experiment 3

To determine the pros and cons of DT Learner and RT Learner, they are compared in terms of their prediction performance and training speed using the mean absolute error (MAE) and time to complete training metrics. Both learners are trained to make predictions on only the Out-Sample at various leaf sizes ranging from 1 to 50. This is repeated for 10 independent iterations and the average MAE for each leaf size is computed. In each iteration, a training set is randomly split from the dataset. For training time, the time taken for training to complete for each learner at each leaf size is computed by taking the difference between end and start time. The average time is computed from 10 independent iterations. The results are plotted and compared.

3 DISCUSSION

3.1 Experiment 1

Figure 1 shows that the RMSE of predictions for In-Sample and Out-Sample diverge significantly when leaf size decreases from 10 to 1. This disparity is an indication of overfitting the DT Learner to the training data. Overfitting is a phenomenon in machining learning where the algorithm is very good in predicting the data on which it is trained, but performs badly in predicting data it has not

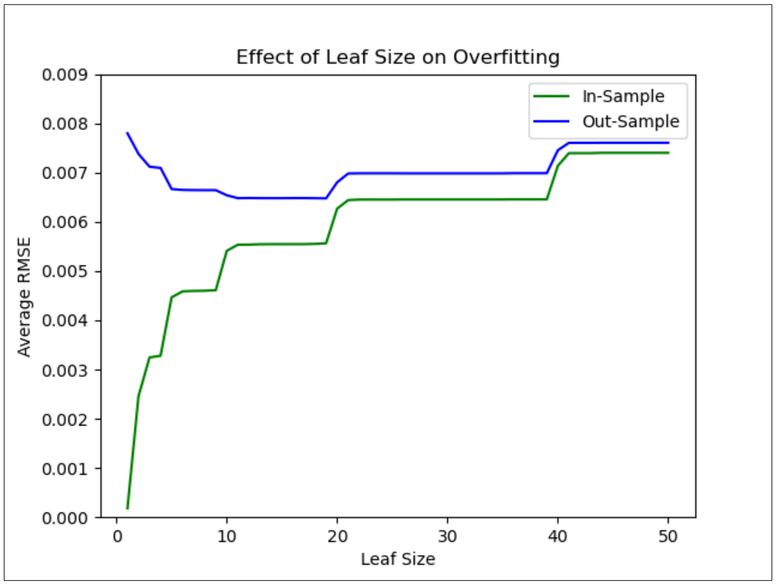

Figure 1— Effect of leaf size on overfitting of DT Learner indicated by the change of In-Sample (train) and Out-Sample (test) RMSE as a function of leaf size. The average RMSE are the results of 10 independent iterations of train-predict cycles over varying leaf size of 1-50.

seen before. It occurs when the algorithm over-learns the training data, picking up random noises beyond the general patterns in the data. As a result, it is unable to generalize what it has seen in training to the unseen dataset. This often happens when the model becomes too complex and has a high degree of freedom. In the case of DT Learner, complexity and degree of freedom increase with decreasing leaf size (figure 1). Such a model has high variance. Hence, to mitigate overfitting, leaf size can be increased from 1 to 10. Leaf size 10 is the optimal, below which overfitting will start. Leaf size can be increased without affecting RMSE until about 20, beyond which RMSE start rising again, indicating deterioration of prediction performance due to underfitting. When the leaf size is more than 20, the DT Learner becomes too simple to pick up pattern in the training dataset. Not enough features are utilized to split nodes. Such a model has high bias. Taken together, experiment 1's results indicate that leaf size is a critical hyperparameter for a decision tree algorithm. It is inversely related to overfitting.

3.2 Experiment 2

Overfitting can also be mitigated by employing techniques, such as bagging. Figure 2 shows that bagging reduces overfitting in DT Learner. The results are obtained from combining 20 DT Learners trained with different bootstrap samples. The gap between the RMSE of In-Sample and Out-Sample from leaf size 1 to 10 is significantly smaller, compared to the gap in figure 1 (the axes of both figures are drawn to the same scale for comparison). The Out-Sample curve from leaf size 1 to 10 is also relatively flat compared to the one in figure 1, suggesting that overfitting can be minimized by bagging when leaf size less than 10 is used. This is achievable because the bootstrapping and ensemble approach reduces the overall variance of the model. Bagging, however, cannot eliminate overfitting, as indicated by the significantly lower In-Sample RMSE below leaf size 10. Furthermore, increasing leaf size beyond 20 leads to deterioration of performance as shown by the rising RMSE. It is because bagging reduces the variance, but not the bias, of the model. Hence, bagging solves overfitting, but not underfitting issue. It is interesting to note that the Out-Sample RMSE of DT Leaner with bagging is below the Out-Sample of DT Learner without bagging over the entire range of leaf sizes studied here (Figure 3). This indicates that bagging improves the prediction performance of the classic DT Learner. Taken together, the experiment demonstrates that bagging mitigates overfitting and improves prediction accuracy.

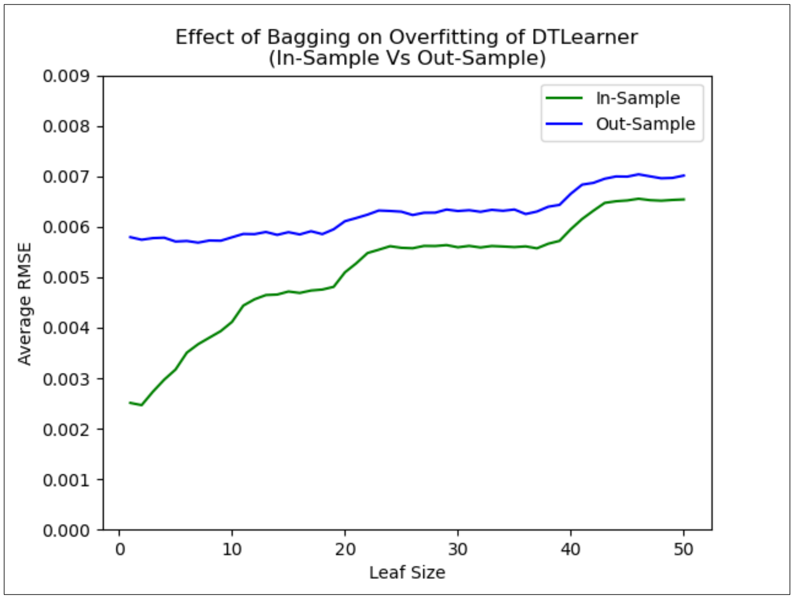

Figure 2 — Effect of bagging (20 bags) on overfitting of Decision Tree learner indicated by the change of In-Sample (train) and Out-Sample (test) RMSE as a function of leaf size. The average RMSE are the results of 10 independent iterations of train-predict cycles over varying leaf size of 1-50.

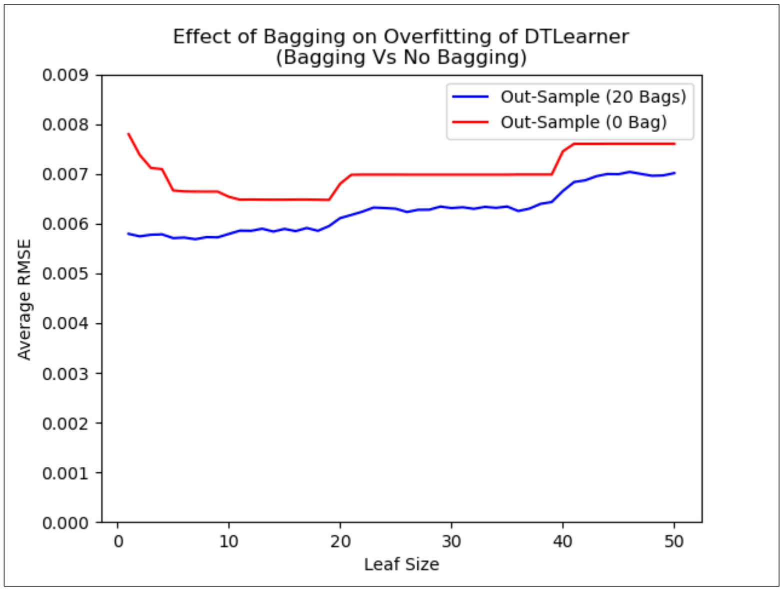

Figure 3— Bagging improves prediction accuracy compared to classic Decision Tree without bagging. The average RMSE are the results of 10 independent iterations of train-predict cycles over varying leaf size of 1-50.

3.3 Experiment 3

In a DT Learner, features selected to split the dataset are based on their correlation with the Y variable. A variant of it that selects features randomly can be built. In this experiment, the strengths and weaknesses of the 2 learners are evaluated and compared over a range of leaf sizes 1 to 50. Figure 4 compares their prediction performance based on the mean absolute error (MAE) metric. The reason for this choice is that it penalizes all errors equally. It is, thus, more robust to outliers. Since prediction performance is assessed, only Out-Sample are compared.

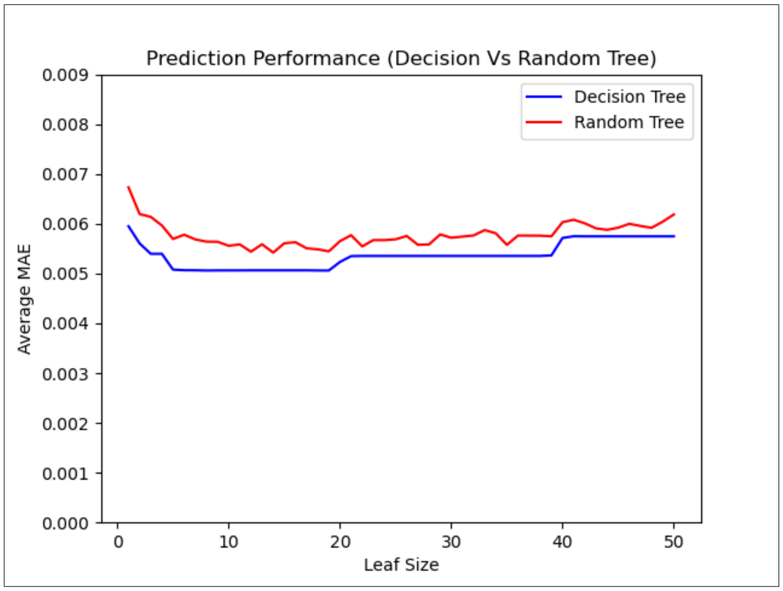

Figure 4— Comparison of prediction accuracy for Out-Sample between Decision Tree and Random Tree learners. The average MAE are the results of 10 independent iterations of train-predict cycles over varying leaf size of 1-50.

Figure 4 shows that the DT Learner curve is below the RT Learner over the entire range of leaf sizes studied. This suggests that predictions made by DT Learner are closer to the actual values than those made by RT Learner. Hence DT Learner performs better than RT Learner in prediction. This is expected, given that DT Leaner selects features based on their correlation to the Y variable, while RT

Learner randomly selects features which may not have any predictive relationship to the Y variable.

Although DT Learner performs better than RT Learner in prediction, the former takes more time than the latter to complete training (Figure 5). The difference grows larger with decreasing leaf size. This is expected, given the longer computational time needed to calculate and compare the correlations of the features. Hence, in applications where time or speed is critical, RT Learner may be the preferred choice.

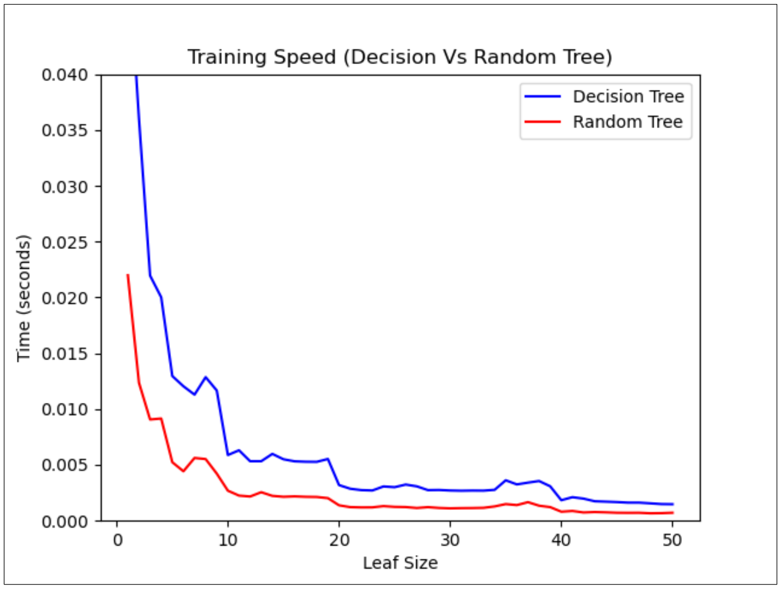

Figure 5— Comparison of time taken to complete training between Decision Tree and Random Tree learners. The time are the average results of 10 independent iterations of training cycles over varying leaf size of 1-50.

4 SUMMARY

Results from the 3 experiments have demonstrated that leaf size is a critical hyperparameter that influences overfitting and underfitting in CARTs algorithms. Bagging is an effective technique to mitigate overfitting and improve prediction performance. Classic DT Learner can predict more accurately but takes longer time to train than RT Learner.

5 REFERENCES

1. James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.
2. Quinlan, J.R. (1986). Induction of Decision Tree. *Machine Learning* (1), 81-106.