Which is Better?
Project 3: Assess Learners

Mahmoud Shihab
mshihab6@gatech.edu

Abstract—In this project, the expectation is to explore the performance of different kinds of learners, mainly Decision Trees, Random Trees, and Bootstrapped Aggregated Trees (either DT or RT). The objective of this paper is to explore when a certain algorithm is better than others, at what conditions do they tend to overfit, and if bootstrapping helps alleviate overfitting. These questions by looking at the effect leaf size has on the output, and will be measured through the use of metrics like RMSE, R2, Correlation, MAE, and Maximum Error.

1 INTRODUCTION

This paper will look at the behavior and performance of three machine learning algorithms from an algorithmic family called Classification and Regression Trees (CARTs). To do so, three experiments will be conducted to understand Decision Trees (JR Quinlan), Random Trees (A Cutler), and Bootstrapped Aggregated Trees (also known as Bagging).

2 METHOD

All experiments will be analyzed through the lens of checking different metrics between In Sampling (predictions vs training labels) and Out of Sampling (predictions vs testing labels), and seeing how they both change. Since this report will be looking at trees, for each experiment, the change in metrics will be looked at over different leaf sizes.

1. The first experiment will take Decision trees and vary their leaf size to see the changes in RMSE metric.
2. The second exercise will look at bagging 20 Decision Trees, and just as before, check how the RMSE change with respect to their leaf sizes.
3. The last experiment will compare Decision Trees and Random Trees, seeing how their MAE, R2, and ME change with leaf size.

The Istanbul dataset will be used for the experiments. It includes the returns of multiple worldwide indexes for several days in history. The learners will predict what the return for the MSCI Emerging Markets (EM) index will be based on the other index returns. The split will be 60% training, 40% testing, randomly assigned.

3 DISCUSSION

An imporant topic in Machine Learning is Overfitting. Overfitting occurs when an algorithm model fits exactly against its training data (Awan, 2023, Overfitting). Normally, training and testing errors tend to follow each other somewhat, but according to Professor Tucker Balch, overfitting is categorized by a drop in training error and a rise in testing error.

This means that the model is considered the best in training, but will perform poorly with new unseen data. This is why important to understand what overfitting is. So one can avoid it.

Check for overfitting by plotting the training and testing errors vs. the model parameters, finding the deviation point. Other ways to avoid overfitting include having simpler models, collecting more data, and using ensemble learners (like bagging) (Awan, 2023, Overfitting)(IBM, 2023).

3.1 Experiment 1: Decision Tree vs Leaf Size

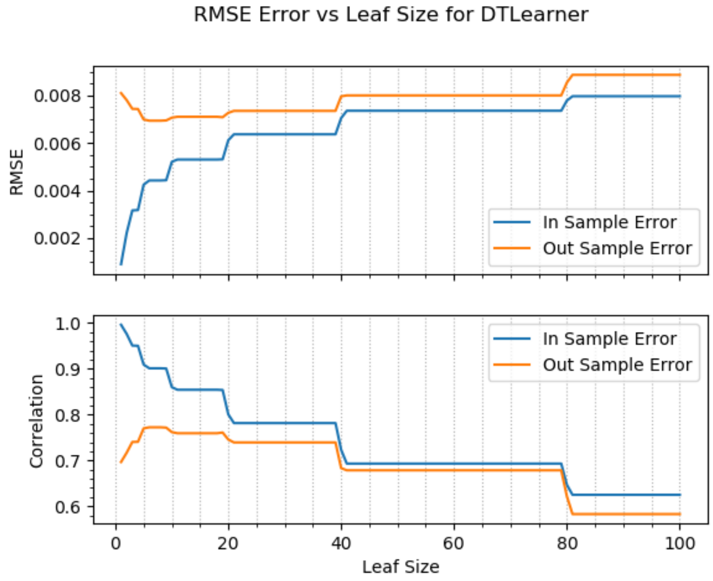


Figure 1—RMSE and Correlation of DT varying with leaf size

As shown in Figure 1, the Decision Tree performs perfectly In Sample with a leaf size of 1, boasting a RMSE[1] of 0.000907 and a correlation of 0.996046. Out of Sample, the RMSE is 0.008102 and the correlation is 0.696625. This wide deviation of In and Out is evidence of overfitting. In fact, it seems like overfitting occurs at around a leaf size of 5, where the RMSE for both in an out are 0.004249 and 0.006992 respectively. From there, as leaf size decreases, the In Sample RMSE will continue to drop and the Out of Sample will continue to rise.

Leaf Size	RMSE In	RMSE Out
3	0.003169	0.007440
4	0.003183	0.007433
5	0.004249	0.006992
6	0.004428	0.006941
7	0.004428	0.006941

Table 1—The RMSE values around the overfitting lea

CARTs leaf_size have the same effect on KNNs as n_neighbors. The smaller the value, the less the model generalizes, and will tend to match the data exactly. For this reason, most CARTs will tend to overfit with a leaf_size closer to 1.

Having a smaller leaf size means less data points in each node. In turn, this means that the tree will grow deeper and deeper. The deeper the tree, the more the splits, meaning each node has less points to learn from, effectively memorizing the data the closer to 1.

As the tree becomes less general, the error for the in sample (training) data will continue to decrease, as the tree matches the data more and more. However, it means the tree will struggle to understand the out of sample (test) data, and therefore the error will eventually start to rise (as seen above).

3.2 Experiment 2: Bagged Decision Tree vs Leaf Size

As stated in the discussion, ensemble models are another way to reduce overfitting. With bagging, a collection of weak learners are combined together using bootstrapped data, and the average answer is the one that is used. This helps

[1] RMSE, MAE, and ME are very dependent on the scale of the feature in question. For example, if the RMSE for a tree is 200, but the values of the datasets target is in 10,000s, then the model is performing very well. If another tree has the same RMSE value, but the datasets target is in the 1,000s, it is not performing as well. As such, non-normalized error metrics must be looked at with respect to their targets.

reduce the variance, and helping to increase accuracy and reduce overfitting. (James et al., 2023)

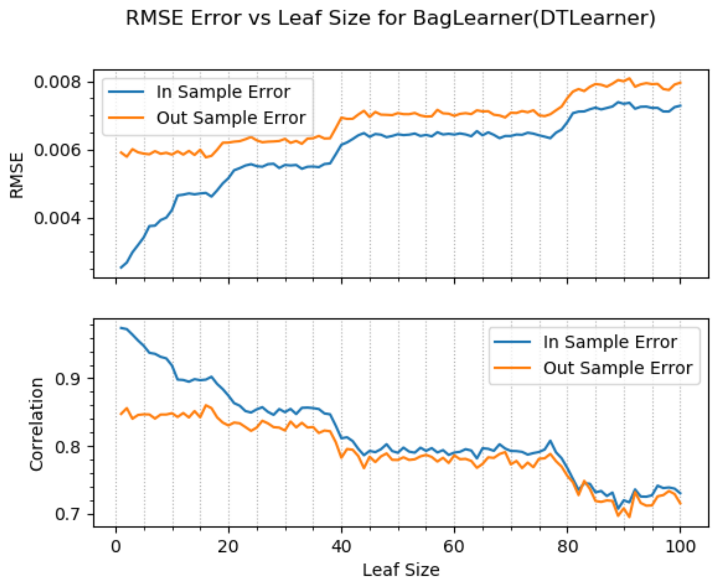


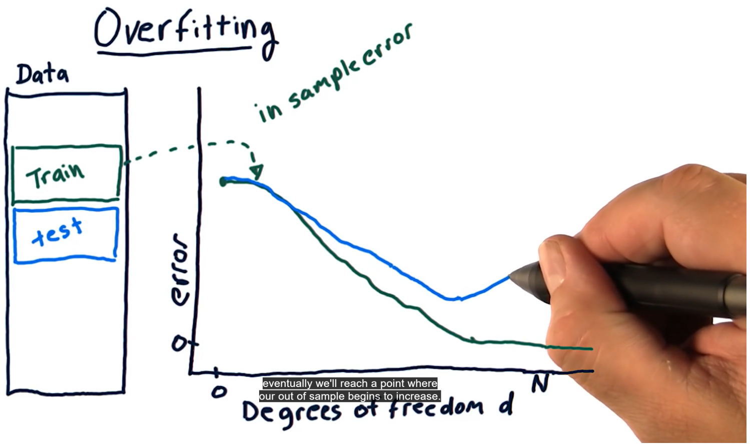
Figure 2—RMSE and Correlation of Bagged DT varying with leaf size

As seen in Figure 2, the testing error isn't increasing, but isn't decreasing either at the lower leaf sizes with an average error of ≈ 0.005888. Besides reducing the effects of overfitting, it also reduced the testing error when compared to just using Decision trees. The training error still moves as expected. More importantly, the trend between training and testing error seems to be more relevant than with just Decision Trees.

An argument can be made that around a leaf size of 10[2], since the testing error begins to deviate from the pattern of following the decreasing training error. But unlike in experiment 1, the testing error doesn't being to rise, which was a point made by Professor Tucker Balch. (See screenshot below)

This mitigation of Overfitting is a strong reason why bagging can help reduce the effect of overfitting. By exposing the constituent models to different parts of the dataset, the model is composed of several different trees, each with a lower variance. This allows the average across all trees to incorporate the strengths of each tree and reduce the effect of each trees error. (Awan, 2023, Bagging)

[2] A leaf size of 10 seems optimal as the gap between the two erros are small, and the trend between them seems to start at this point



eventually we'll reach a point where our out of sample begins to increase.

Figure 3—Screenshot of Overfitting Lecture provided by OMSCS ML4T Course

3.3 Experiment 3: Decision Tree & Random Trees vs Leaf Size

In this experiment, Decision Trees are compared against a modified version of A Cutlers Random Trees. Originally, Decision Trees select the best feature to split on using either Entropy, Correlation, or Gini Index. In Cutlers original design, Random Trees take a random feature and finds the average between two random points of that feature to use as the splitting value. The algorithm used in this experiment is almost identical to Cutlers original design, except that it used the median value of the whole column.

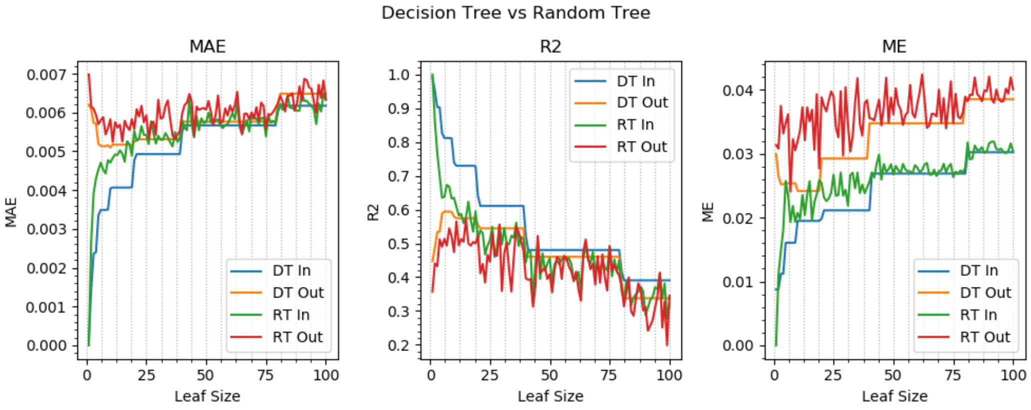


Figure 4—MAE, R^2, ME of DT & RT varying with leaf size

In Figure 4, the mean absolute error, R^2, and Maximum Error metrics are presented between Decision Trees and Random Trees with varying leaf sizes. On average, Decision Trees performed better than Random Trees. As shown in Table 2, the average errors are negative (meaning Random Tree errors > Decision Tree errors), and the R^2 was positive (meaning Decision Tree R^2 > Random Tree R^2).

	Average
MAE In	-0.000320
MAE Out	-0.000275
R2 In	0.074124
R2 Out	0.050251
ME In	-0.001718
ME Out	-0.004120

Table 2—The average difference between Decision Trees and Random Trees.

Another interesting realization is that in Random Trees, the testing MAE and ME values seem to run a little flatter. Unlike the decision tree values, which have a noticeable decline as leaf size decreases, the Random Tree values out of sample seem to move along and follow their mean values. Looking at the rolling averages for the metrics, it's clear that the Random Tree out of sample error metrics are a bit flat.

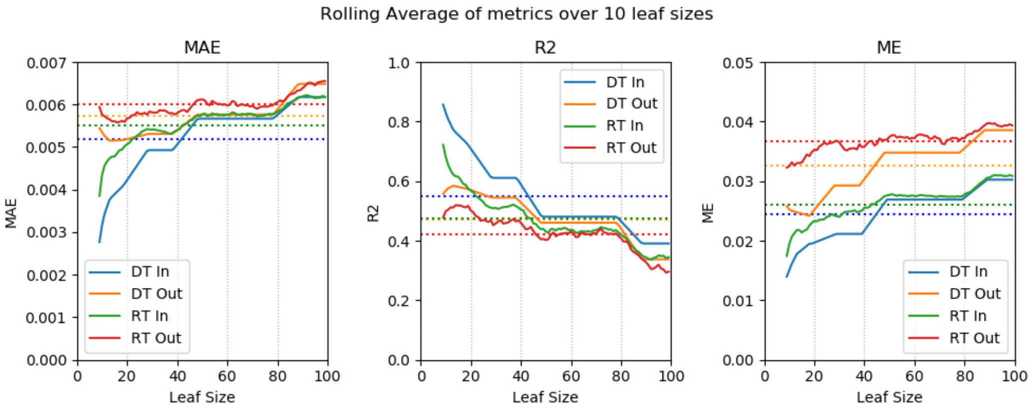
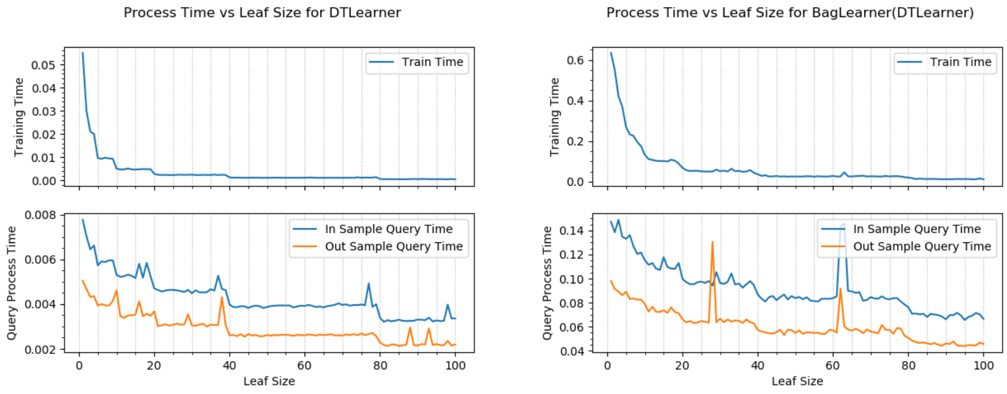


Figure 5—Rolling Average of MAE, R^2, ME of DT & RT varying with leaf size

All of these factors show that Decision Trees out perform Random Trees, and this is probably due to the random splitting. The splitting of Decision Trees is based on the entropy/gini index/correlation of the column with target variable. Selecting the column at random may select a less than optimal column to split on. This inefficiency seems to mean less and less when the leaf size is high, but as shown by Decision Trees and Bagging Learners, that leads to higher and higher errors in general. While both trees will eventually overfit, the increase performance for Decision Trees and basis for splitting shows that they are superior to Random Trees, despite their algorithms being almost identical.

4 SUMMARY

CARTs are very powerful tools that are used widely in the field. Their simplicity makes them interpretable models. Their structure makes finding feature importance easy. They are quick to train and even quicker to query. They are a bit prone to overfitting at lower leaf sizes, but as demonstrated in this report[3], it's detectable and avoidable, either by increasing the leaf size (pruning), or bagging the trees.



(a) Decision Trees

(b) Bagged Trees

Figure 6—Process Time Metric for Experiment 1 and 2

A potential way forward for this investigation would be to look at boosting and it's effect on the metrics/results vs bagging. It seems like bagging would be more effective (in general, but also) in tacking the overfitting than boosting, while boosting seems to help improve model performance (Banerjee, 2020). Another point of investigation would be max_depth and other forms of pruning, and seeing how these values affect overfitting (Ravindran, 2023).

[3] Exercise 1 demonstrated that Decision Trees can overfit at lower leaf sizes, and detecting it is possible. Exercise 2 showed how bagging can help reduce overfitting and improve a models performance by combining weak learners strengths and eliminating their weaknesses.

5 REFERENCES

[1] Awan, Abid Ali (Nov. 2023a). *What is Bagging in Machine Learning? A Guide With Examples?* URL: https://www.datacamp.com/blog/what-is-overfitting.

[2] Awan, Abid Ali (Aug. 2023b). *What is Overfitting?* URL: https://www.datacamp.com/blog/what-is-overfitting.

[3] Banerjee, Prashant (June 2020). *Bagging vs Boosting*. URL: https://www.kaggle.com/code/prashant111/bagging-vs-boosting.

[4] IBM (2023). *What is Overfitting?* URL: https://www.ibm.com/topics/overfitting.

[5] James, Gareth, Witten, Daniela, Hastie, Trevor, Tibshirani, Robert, and Taylor, Jonathan (July 2023). *An Introduction to Statistical Learning*. Springer. ISBN: 9783031387463. URL: https://hastie.su.domains/ISLP/ISLP_website.pdf.download.html.

[6] Ravindran, Rishika (Jan. 2023). *Overfitting and Pruning in Decision Trees — Improving Model's Accuracy*. URL: https://medium.com/nerd-for-tech/overfitting-and-pruning-in-decision-trees-improving-models-accuracy-fdbe9ecd1160.