Project 3: Assess Learners

Nathan Riojas
nathanriojas@gatech.edu

***Abstract*—Tree learners can vary in their implementation and behavior. In this paper, decision trees of varying leaf size are investigated to study overfitting, and bagging is performed to assess its effect on overfitting reduction. Further investigation is also done to compare decision tree and random tree algorithms based on performance and accuracy.**

1 INTRODUCTION

This paper investigates the behavior of decision trees, random trees, and ensemble learners over the course of three experiments with the Istanbul.csv dataset.

1.1 Note on Regression

While tree learners are often used to solve classification problems, the implementations here solve regression problems. As such, the mean of training y values is used when data is aggregated to create leaves. However, were this a classification problem, the approach would call for the use of the mode instead of the mean.

1.2 Learner Summary

Decision trees were implemented using an algorithm developed by JR Quinlan which recursively builds a tree given training data. This type of tree creates nodes by selecting the median of the feature having the highest correlation with the training y values, and continuously does this until the recursive splits arrive at a leaf node. The built tree could then be queried to give the expected result for an instance or several instances of testing data.

Random trees were built in a similar manner as decision trees, except the feature chosen when selecting the median to create a node to further branch the tree was randomly selected, not requiring the correlation calculations of each column with the training y data. This resulted in the tree being built differently than it was for a decision tree. Querying random trees was the same process as it was for decision trees.

Ensemble learners were implemented using boostrap aggregating or bagging. These learners were given an input number of bags or instances to make of a specific learner. In this investigation, the size of the data the bag learner passed to a specified bag (commonly referred to as n') was equal to the training data size n.

The data passed to the learner within the bag however was created by randomly sampling the input training data with replacement. Thus, when querying the bag learner, it queries the multiple learners or bags inside it, gathers the predicted values of each, and then returns the mean of said values.

1.3 Insane Learner Quick Note

This was a learner that was not studied in further detail but is worth discussing briefly. The insane learner was a bagging learner that implemented 20 bag learners each having 20 bags of linear regression learners.

1.4 Leaf Size and Recursion

Decision tree and random tree learners were further characterized in this study by the size of leaf specified. Leaf size defines the point at which the remaining data is aggregated into a leaf. Thus, the training y values are averaged to produce the value the built tree would return when queried.

It is important to note, in the actual implementation of these tree classes, when recursively creating the tree, leaf size is what is used as the base case to terminate recursion.

Two additional cases were considered. The first being the case in which all y values of a dataset were the same, in which case further recursion was not necessary so a leaf was made and returned. The second was the case in which the median of the feature chosen to split did not adequately split the data. For decision trees, the next best correlated column was used instead, or if no columns could produce valid splits, a leaf was created and returned. For random trees, a leaf was created and returned as well.

1.5 Overfitting Definition

Before detailing the experiments, overfitting must be understood, as this is how decision trees and bag learners were studied. Overfitting is the behavior seen

when in sample error decreases as out of sample error increases. This behavior is not desired so determining regions in models where this is present helps in knowing when and how to apply these learners.

1.6 Hypothesis

With regards to decision trees, it is to be expected that overfitting will be inversely proportional to leaf size. This is because the lower the leaf size, the more likely the leaf aggregation values will tend to match the training y values in a one-to-one manner. However, it is expected that bagging will be able to aid in reducing overfitting with respect to decisions trees as leaf size varies.

In comparing random trees and decision trees, the expectation is that decision trees will be more accurate and consistent than random trees. The logic is simply that decision trees carefully select the nodes they create to make the next branch or node in the tree using correlation, so this should produce less error in the tree built. However, random trees exist for a reason, so it is expected that they will run faster than decision trees to some degree.

2 METHODS

Three experiments were done to study these learners. All investigations were done comparing a specific metric against the learner in question for different leaf sizes.

2.1 Experiment 1

In the first experiment, 100 decision trees were trained and tested. The same data was used to train each of these trees, however, each tree varied in its leaf size (from 1 to 100). The metric studied here was the root mean square error or RMSE, which was calculated using the formula below.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

This was done for both in sample and out of sample data by querying the trained tree with training and testing data, and subsequently measuring the RMSE of the output against the expected training and testing results, respectively.

Since RMSE is a metric that is useful for determining the overall goodness of fit of a model, to better understand the accuracy of decision trees, the 100 trees of varying leaf size were plotted against their RSME. This provided a better understanding of how leaf size affected accuracy, in sample and out of sample. From this plot, the region of overfitting was able to be determined.

2.2 Experiment 2

This experiment was virtually identical to experiment 1, with the caveat that instead of a single decision tree at each leaf size, 20 trees or bags were created. This was the application of a bagging learner. The default number of bags selected in this experiment was 20, however, this number was arbitrary and could be increased or decreased as desired.

As an example, in the instance of a leaf size of 50, a bag learner was created with the specification of making 20 bags or instances of decision trees. Each tree would then be initialized with the hyperparameter of leaf size set to 50.

The data each decision tree within the bagging learner was trained on varied however as this is how bagging works. The data given to the bag learner to train on was randomly sampled with replacement and then given to each bag instance of the decision tree to train on and develop its tree. The size of data randomly sampled was specified to be the same as the data input. The variation that occurred was a result of sampling with replacement, meaning duplicate values of data could occur. Hence, the 20 bags of decision trees were trained on different versions of the training data.

The same style of plot was generated to study bagging learners and overfitting: RMSE vs leaf size.

2.3 Experiment 3

This experiment diverged from the study of overfitting and instead aimed to compare decision trees and random trees. Two metrics were used to accomplish this. The first being the Mean Absolute Error (MAE) as calculated below.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Compared to RMSE, MAE more directly represents the cumulation of error. While mean square error tends to heavily penalize large prediction error, MAE treats all error the same. With this in mind, the MAE for expected y values versus predicted y values of decision tree and random tree models was calculated in sample and out of sample and plotted against leaf size.

The second metric used to compare decision trees and random trees was time, specifically training time. Perhaps the biggest disadvantage to using tree algorithms for machine learning, is the time it takes to train in comparison to say a linear regression model. Querying time is actually pretty fast, assuming the tree is able to remain relatively balanced. It is therefore of great interest to understand ways to improve the training time, especially if exceptionally large datasets are to be used.

In order obtain this metric, Python's time module was used to take the time before and after executing the training function (called add evidence in this paper's implementation) and subtract the two to get the total training time. This was plotted against leaf size for both decision trees and random trees.

3 DISCUSSION

Please note that on the graphs for figures 1-6, the *number of leaves* x axis label refers to leaf size as discussed earlier specified when making the specific tree, not the number of leaves in the tree.

3.1 Experiment 1

Upon plotting RMSE vs leaf size for decision tree learners, as seen in Figure 1, it is apparent that the initial hypothesis regarding overfitting was correct. Overfitting is inversely proportional to leaf size. Based on the graph, it can be seen that overfitting occurs for leaf sizes less than about 9. The highlighted region to the left of that in Figure 1 is the area for which overfitting occurs. Therefore, overfitting occurs from right to left starting at a leaf size of about 9.

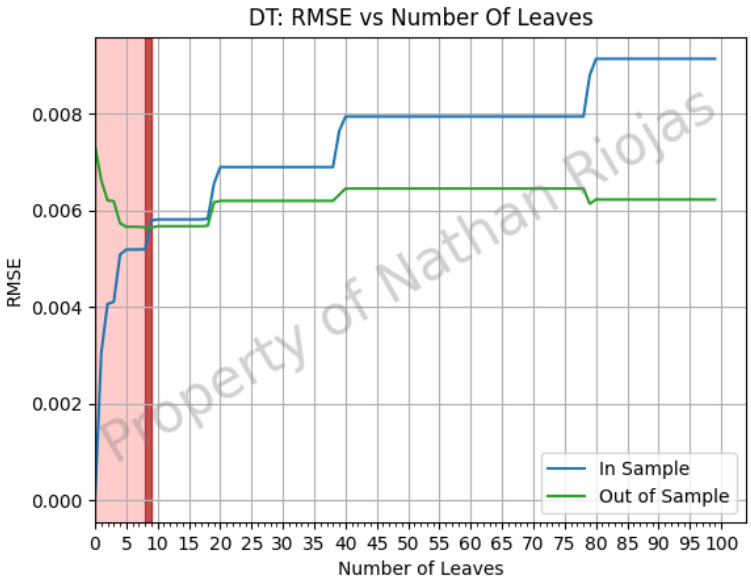

Figure 1 — Analysis of Decision Tree RMSE vs Leaf Size

3.2 Experiment 2

Interestingly, when plotting RMSE vs leaf size for the bag learners, as seen in Figure 2, it appears that the initial hypothesis that overfitting can be reduced using bagging was incorrect, or if correct, correct only to a marginal degree. Overfitting can be observed prior to the point of intersection of the in sample and out of sample lines. This appears to occur up to a leaf size of around 9, the same as was observed in Experiment 1 for decision tree learners. Prior to that leaf size, which is again highlighted in Figure 2, is the area for which overfitting occurs. Again, overfitting occurs from right to left starting at a leaf size of about 9.

With a fixed bag size of 20, it does not seem that bag learners can eliminate bagging nor can they reduce it. This could possibly be attributed to the Quinlan algorithm used to design the decision trees. It is possible that the consistency yielded when selecting columns based on correlation negates the effect bagging could have on reducing overfitting.

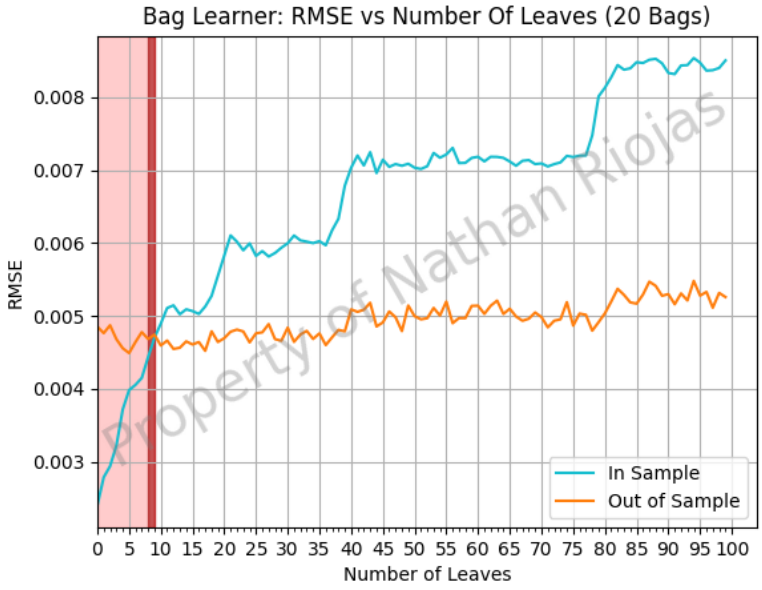

Figure 2 — Analysis of Bag learner RMSE vs Leaf Size

3.3 Experiment 3

Figure 3, Figure 4, and Figure 5 illustrate the comparison of the MAE of decision trees and random trees in sample and out of sample versus the leaf size. As expected, the random trees produce a highly scattered error trend, though it does seem to follow the same trend as decision trees, albeit generally with higher error. The conclusion that can be drawn here is that decision trees are superior in terms of consistency and tend to have less error. This in large part can be attributed to the column selection method which creates a generally more accurate tree.

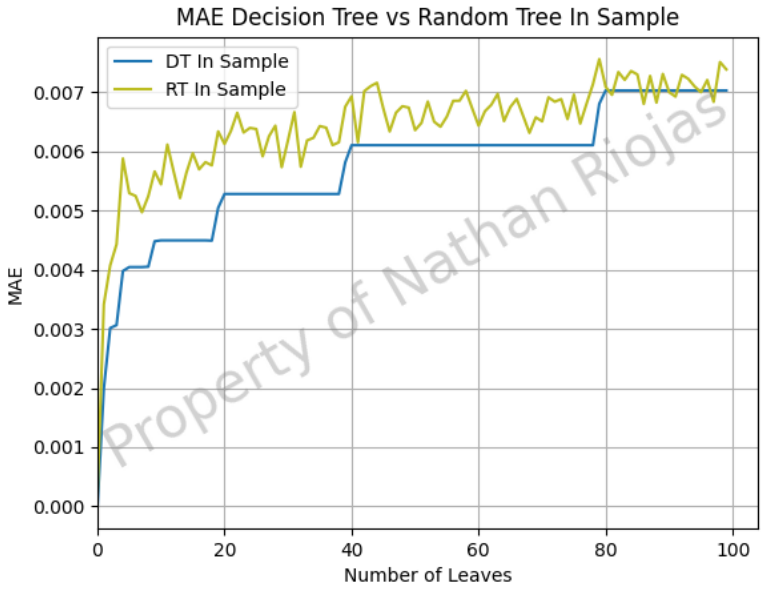

Figure 3—Comparison of Decision Trees and Random Trees MAE vs leaf size in sample

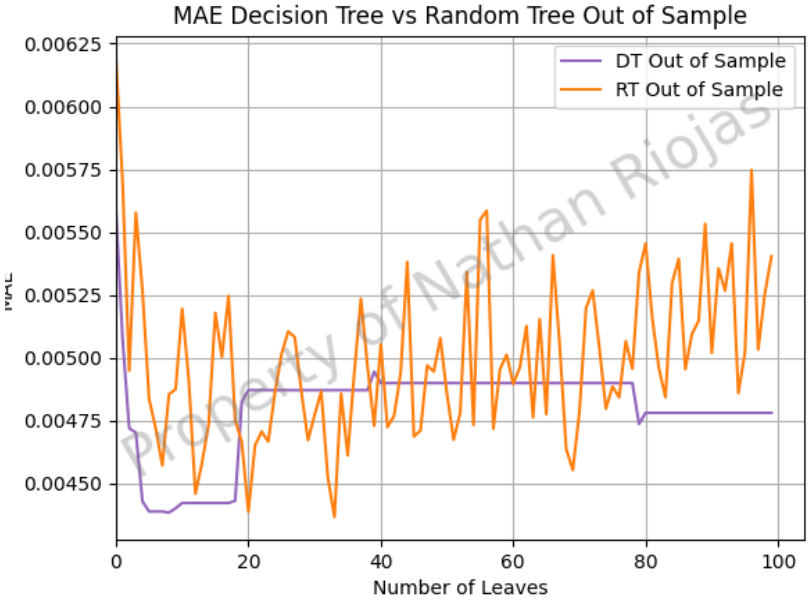

Figure 4—Comparison of Decision Trees and Random Trees MAE vs leaf size out of sample

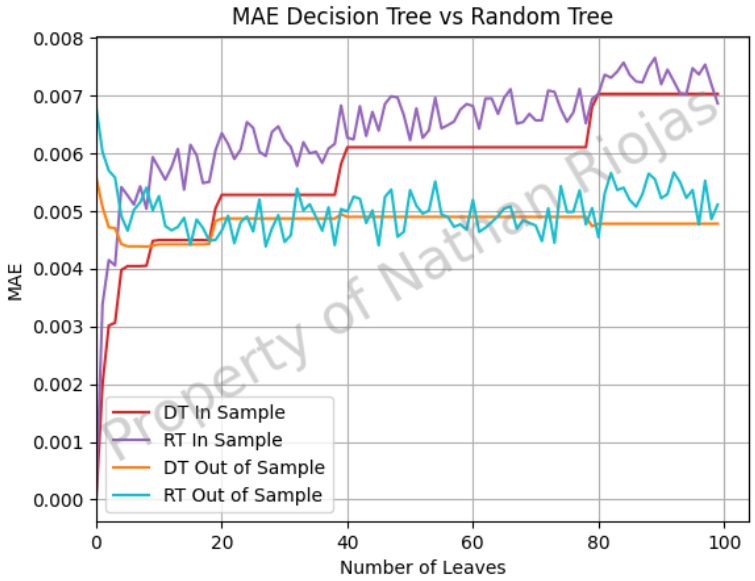

Figure 5 — Comparison of Decision Trees and Random Trees MAE vs leaf size in and out of sample

Interestingly though, when comparing the time to train random trees vs decision trees, random trees are superior. This can be seen in Figure 6. For smaller leaf sizes, the time to train random trees is observed to be about 5 times faster. It is not until a leaf size of around 80 or greater that the time to train begins to equal out.

The quick training time of random trees can obviously be attributed to the random selection of columns when building its tree. Since it does not perform the correlation calculations for each label against the training y values that decision trees perform, the training time is lessened. This means that for models with many labels, random trees will especially prove to be superior in training time.

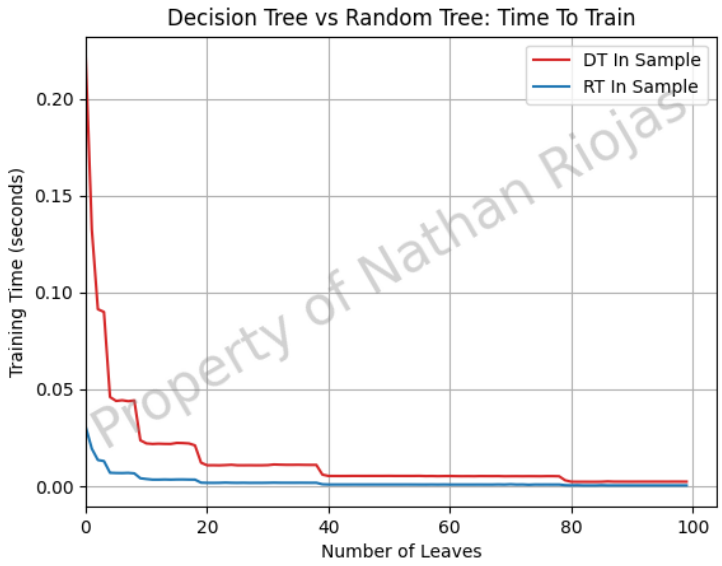

Figure 6 — Comparison of training time of decision trees and random trees vs leaf size

4 SUMMARY

Identifying regions where overfitting occurs is crucial when building a learner. For decision trees, overfitting tends to occur when leaf sizes are smaller, specifically in this investigation around a size of 9. Therefore it is preferable to build a decision tree with a leaf size higher than this to generate a better learner. Interestingly, bagging cannot be used to reduce this zone of overfitting in decision trees. As seen in this paper, the same region of overfitting seems to occur with or without bagging for decision trees.

Furthermore, when determining whether to use decision trees or random trees, it is crucial to identify the metrics that are preferable for the specific task. When better accuracy is the most important factor, a decision tree proves to be more predictable in terms of error and also tends to have less error as well. However, if time is the key factor, random trees provide a learner that is much quicker to train when compared to decision trees. So for large datasets or datasets with a high number of labels, random trees can prove more preferable due to the speed at which they can be trained.