Machine Learning for Trading
Project 3 Report
By: Sarah Hernandez

Question 1:

Methodology: Figures 1 and 2 both show the same run with DTLearner using the Istanbul dataset, with the latter graph showing a zoomed in scale for clarity. In-sample and out-sample RSME are compared.

Answer: For decision trees, overfitting does indeed occur with respect to leaf size. As leaf size decreases, overfitting is more likely to occur. This can be observed in the figures below. As leaf-size decreases, the RSME for in-sample data decreases, even reaching zero at leaf_size = 1. However, this alone does not constitute overfitting. Rather, it is the out-sample RSME line that confirms it. From leaf_size = ~10, as leaf_size decreases, out-sample RSME actually increases while in-sample RSME decreases. In other words, overfitting occurs when leaf_size is less than ~10. This is indicative of overfitting, as the smaller leaf sizes are too attuned to the in-sample data, and thus resulting in more errors trying in trying to predict out-sample data.

For leaf sizes greater than ~10, this relationship no longer applies, and both in-sample and out-sample RSME increase as leaf size increases. Thus, we can say that we can avoid overfitting when leaf size is greater than 10 (in this case), or about 3% the size of the training data.

Figure 1:
Leaf Size effect on In-Sample and Out-Sample RSME

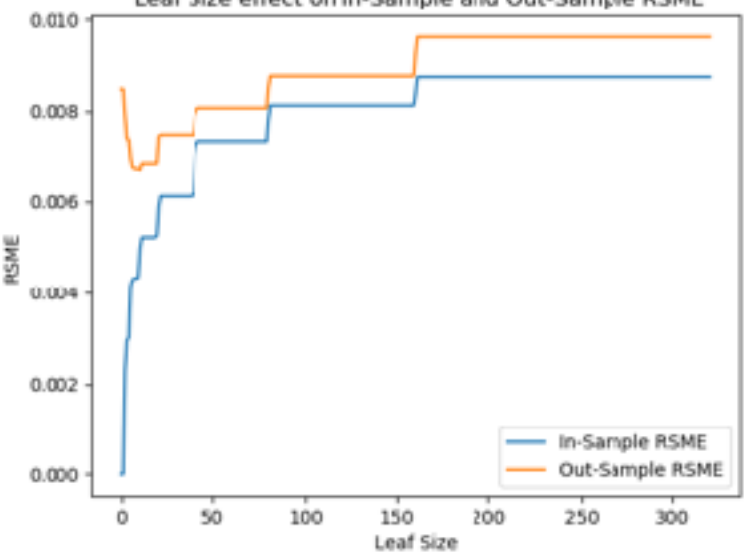
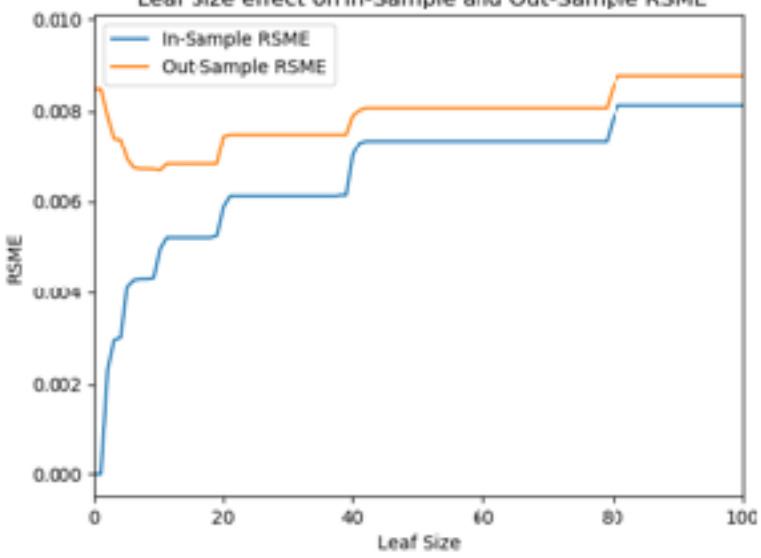


Figure 2:
Leaf Size effect on In-Sample and Out-Sample RSME



Question 2:

Methodology: Figures 3 and 4 both show the same run with BagLearner, which utilizes a DTLearner and, based on the data above, using 15 bags to prevent overfitting. Here, we use the Istanbul dataset, with the latter graph showing a zoomed in scale for clarity. In-sample and out-sample RSME are compared between bagged and non-bagged results.

Answer: Bagging can drastically reduce, but not eliminate the risk of overfitting with respect to leaf_size. Using 15 bags of DT learner, Figures 3 and 4 below show the RSME difference between bagged and non-bagged results for the same dataset. For decision trees in general, overfitting occurs when the in-sample RSME decreases as the out-sample RSME increases. While this pattern is present and clear as day for the non-bagged trial, the pattern seems to break for our bagged trial. As leaf size decreases, in-sample RSME does dramatically decrease. But even with a leaf size of 1, the RSME never reaches zero, or a perfect fit. Likewise, unlike our non-bagged trial, as leaf size decreases, out-sample RSME never shows a consistent and clear increase. Thus, can we say that overfitting has been eliminated? Not entirely, as the in-sample RSME does still dramatically decrease as leaf size dips below ~10. But because the out-sample RSME remains fairly constant over this part of the graph, we can say that we've definitely reduced, but not eliminated overfitting. In other words, bagging will reduce the chance of the model overfitting in-sample data and reduce the likelihood of creating larger out-sample error.

Figure 3:
Leaf Size effect on In-Sample and Out-Sample RSME:
Bagged vs. Non-Bagged Results

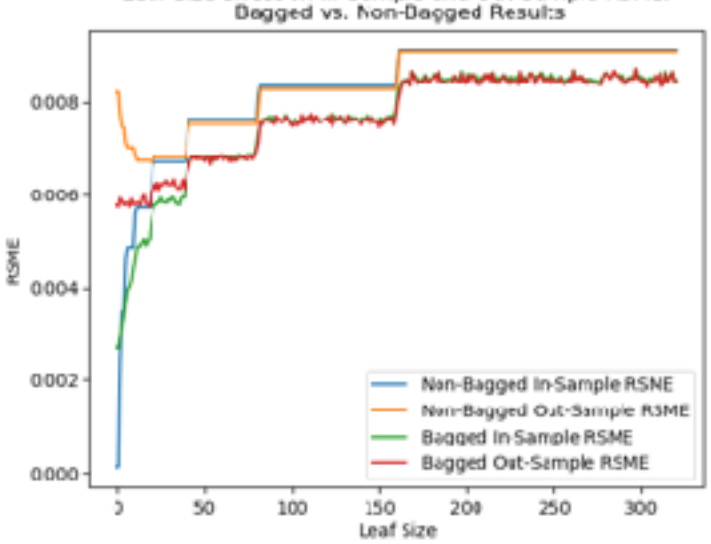
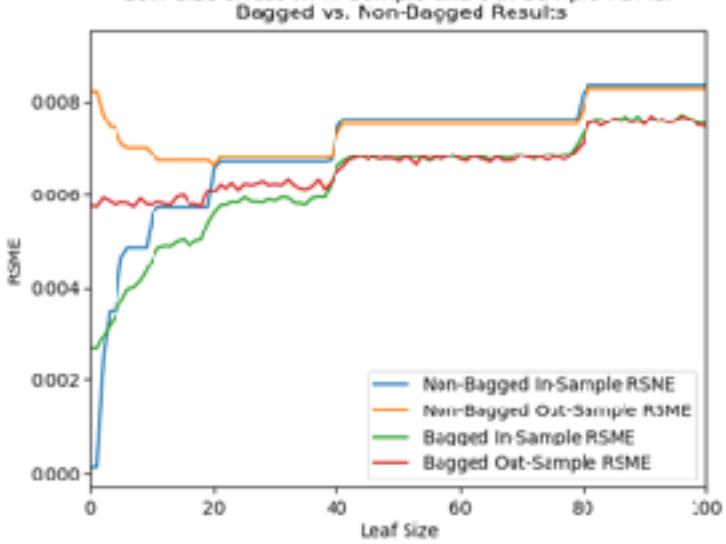


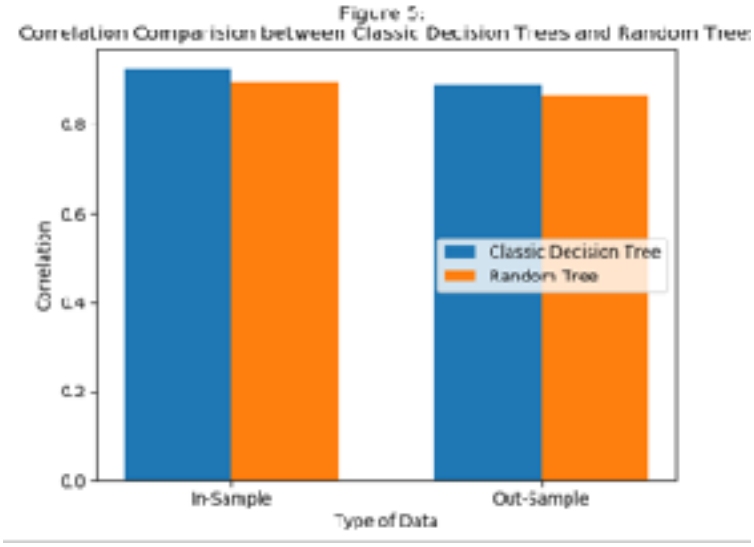
Figure 4:
Leaf Size effect on In-Sample and Out-Sample RSME:
Bagged vs. Non-Bagged Results



Question 3:

Methodology: For this experiment, I compared the DT and RT learners. Based on information about leaf size learned in questions 1 and 2, I set the leaf size to 15 for both learners. Both DT and RT were run five times each, and their results averaged to produce the conclusions below. Three metrics were measured and used for comparison: Correlation, Time to Build the Tree, and Time to Query the Tree.

Metric 1: Correlation:
To compare classic decision trees versus random trees, I measured and charted the average correlation for in-sample and out-sample results for both types of trees. The results are shown in Figure 5 below, with the exact numbers printed out on the right:



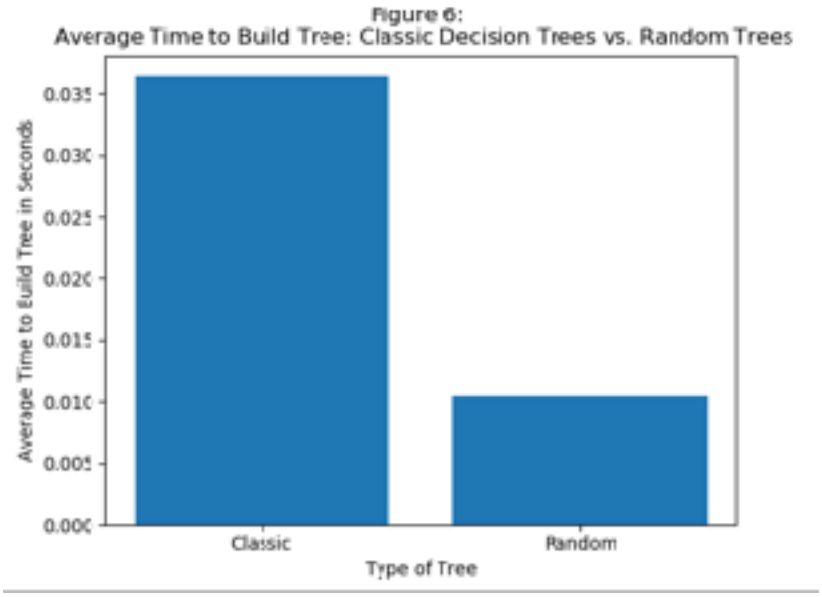

```
Average In Sample
Correlation:
DTLearner:
0.9224100402187336
RTLearner:
0.8967128960369877

Average Out Sample
Correlation:
DTLearner:
0.892150641967324
RTLearner:
0.8651420713871698
```

As you can see, the classic decision tree performs better for both in-sample and out-sample data. This is to be expected, as split values for random trees are chosen at random, rather than by the greatest correlation, as they are in decision trees. Still, I am surprised how accurate random trees remain despite that factor.

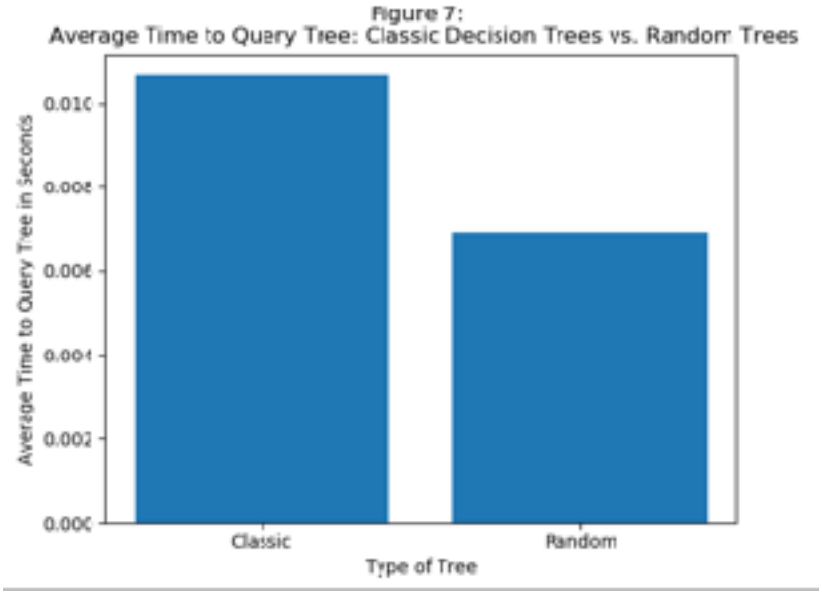
Metric 2: Time to Build the Tree:
Though the correlation between predicted values and real values is higher for the classic decision tree than the random tree, it is more time intensive to build a decision tree. I build each type of tree five times and average the time it took each tree to build itself. The results are shown on the next page in Figure 6.

As you can see, it took the classic decision tree about 3 times long to build itself as the random tree. Though we are talking about microseconds here, we are only working with a few hundred samples. For larger decision trees in industrial settings, the time discrepancy in building a tree would be a factor in considering what type of tree to build in the first place.

In conclusion, although classic decision trees are more accurate than random trees for both in and out sample data, random trees are much faster to build.



Metric 3: Time to Query the Tree:
Build the tree, however, is only theoretically done once, or only after acquiring more sample data. A much more useful metric for data scientists is the overall time it take to query the tree once it is build. Once again, the results below are the average of five runs each of DT and RT learner, with leaf sizes of 15. The results can be seen below in Figure 7:



As you can see, it takes nearly twice as long for the classic tree to be queried as the random tree. I'd imagine that this is because the classic tree is more precise. Its precision results in longer branches and a more complex tree when built, thus resulting in higher querying times.

Conclusion:
Although the classic decision tree is more accurate in both in-sample and out-sample testing (as shown by correlation), the classic tree requires more build and querying time. Though the differences shown here are minimal, that is due to our small sample data. In a larger setting, these differences would become more pronounced. The pros and cons of each type of tree are important to keep in mind when deciding what to use in a professional setting.