CS7645: Machine Learning for Trading
Project 3: Assess Learners
Jihoon ‘Jay’ Song (jsong350)

Experiment 1:

Overfitting tends to decrease as leaf size increases. In order to test this, using the Istanbul.csv dataset, I created a dataframe wherein the column (axis=0) represents the RMSE (Root Mean Square Error). This dataframe was indexed from 1 to 100, representing the number of leaf sizes used to calculate the RMSE. As shown in ‘testlearner.py’, I simulated running DTLearner (Decision Tree Learner) with varying numbers of leaf sizes. The ‘in_sample’ contained the results using in-sample data and ‘out_sample’ contained the results using out-of-sample data. I used the results of the two dataframes and plotted them side-by-side as shown in Figure 1.

A low number of RMSE indicates overfitting since lower number of variance indicates overfitting. Using that logic, according to the dataframe prints, we can see that when leaf size is equal to 1, in-sample RMSE is equal to 0. This means that there is complete overfit. This would make sense, since the same sample used to train is also used to test, and given leaf size is 1, it would have to equal the predicted value exactly, barring any outliers where the same value has multiple predicted values. Using the same logic, according to the dataframe prints, we can see that when leaf size is equal to 1, out-sample RMSE is equal to 0.007. This is because the data being used to test is different from the data that we used to train the model.

As the number of leaf size increases, the in-sample RMSE increases exponentially. This indicates that the degree of overfitting is being reduced. This is because given the leaf size is greater than 1, the predicted leaves can’t be transposed to the exact value from the in-sample data. Out-sample decreases in RMSE but does not get anywhere near 0. This is because the increasing number of leaf size helps the model perform better, since the increased number of leaf size is allowing for better predictions in this out-sample dataset. In conclusion, overfitting decreases as the leaf size increases.

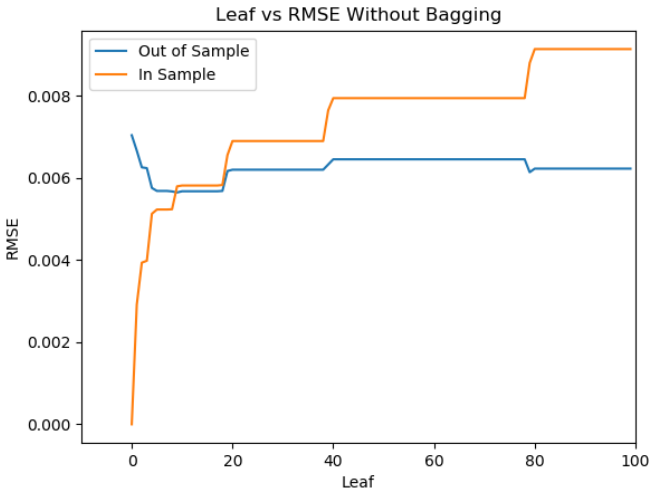

Figure 1. Leaf vs RMSE Without Bagging.

Experiment 2:

In order to test this, again, the RMSE was calculated for both in-sample and out-of-sample data using the DTLearner. In this Experiment, however, bagging was used. Specifically, 25 bags were used in this experiment for each dataset. Both results were plotted side by side for comparison as shown in Figure 2.

As shown in Figure 2. bagging can reduce overfitting. Similar to what we saw in Experiment 1, where we did not use bagging, the in-sample RMSE is still much lower relative to the out-sample RMSE. This indicates a higher degree of overfitting. However, we can also see that overfitting has been reduced significantly, even in the case where the leaf size is 1. Remember that when in-sample was used with leaf size 1, there was complete overfitting. In conclusion, although there is no way to eliminate overfitting, we can reduce it by bagging.

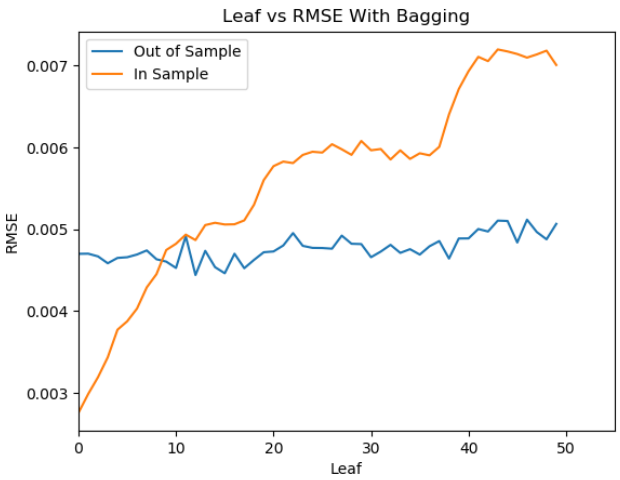

Figure 2. Leaf vs RMSE With Bagging.

Experiment 3:

In order to quantitatively compare DTLearners versus RTLearners, I compared two metrics. Professor Balch mentioned that one of the important ways to assess learners are to compare how quickly they are able to be trained, and how quickly they are able to query to generate predictions. He also mentioned the importance of RAM/memory efficiency of the learners. In order to test this, I measured the start time and end time between training and querying data for both DTLearners and RTLearners.

As shown in Figure 3., RTLearners performed much better than DTLearners when training. This was especially so when the leaf size was smaller. As the leaf size became larger, the training times became more or less the same. This makes sense since it takes much more computing time to figure out the split value (correlation in this example), compared to generating a split value randomly.

Also shown in Figure 3., it was found that there was no significant difference in querying time between DTLearners and RTLearners. Given there were no significant differences, I repeated the experiment 50 times to make sure. Even after 50 trials, there were no significant differences.

As shown in Figure 4., it was found that there was no significant difference in the number of nodes created between the two learners. They are both equally space effective.

In conclusion, RTLearners perform much better than DTLearners when training, especially when the leaf size is smaller. In respect to querying, RTLearners and DTLearners perform about the same, as expected. Similarly RTLearners and DTLearners both perform about the same in respect to space efficiency.

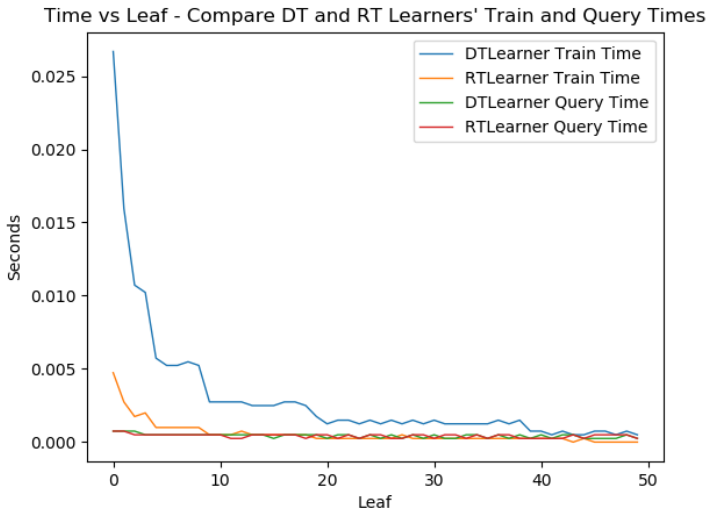

Figure 3. Time vs Leaf to Compare Training and Query Times of DTLearner and RTLearner.

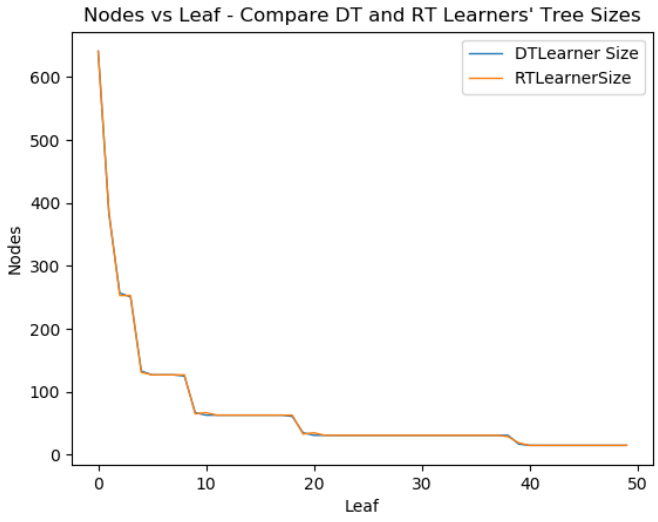

Figure 4. Node vs Leaf to Compare Tree Sizes of DTLearner and RTLearner.