Machine Learning for Trading – Project 3

Tri Nguyen
tnguyen497@gatech.edu

Q1: Does overfitting occur with respect to leaf_size?

The experiment: The Istanbul dataset is used to train and test the Decision Tree model with leaf size from 1-50 (step size of 2). The in-sample and out-of-sample root-mean-square-error are plotted.

Result (figure 1): The result suggests that overfitting occurs with respect to leaf-size. Overfitting occurs when the model fits the training data very well, but does not perform well with test (out-of-sample) dataset.

When the leaf size is large, both in-sample and out-sample errors are large, but they decrease slowly as leaf size is reduced, indicating that the model is becoming more accurate.

However, at around leaf size of 8 or less, the in-sample RMSE declines rapidly, but out-of-sample RMSE increases rapidly. Small leaf size causes each leaf to fit a small number of training samples, and they therefore do not generalize well with new data .

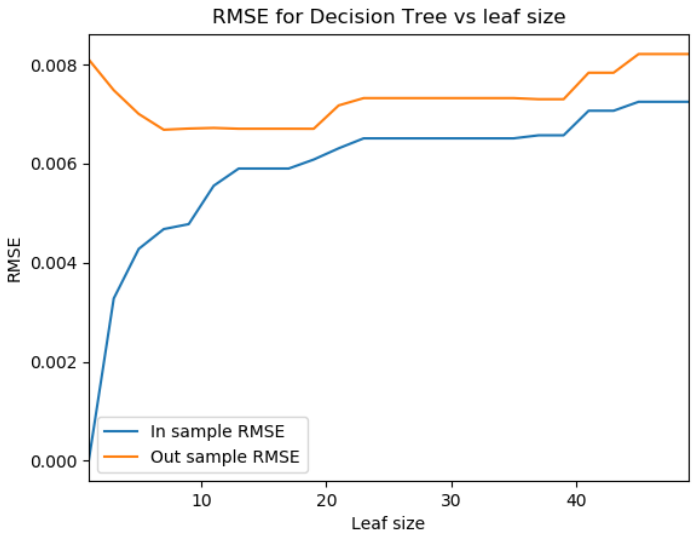

Figure 1: RMSE for Decision Tree vs leaf size

Q2: Can bagging reduce or eliminate overfitting with respect to leaf_size?

The experiment: The Istanbul dataset is used to train and test the Bagging model using Decision Tree and bag size of 10. Leaf size ranges from 1 to 50 (step size of 2). The in-sample and out-sample root-mean-square-error are plotted.

Result (Figure 2): Compared to figure 1, when leaf size decreases to below 8, the out-sample RMSE does not increase rapidly. In fact, it remains relatively flat, and the out-sample RSME at leaf-size of 1 is still lower than that at leaf-size of 50.

It is also noted that the overall RMSE is lower with Bagging Learner than with Decision Tree learner.

It therefore suggests that bagging can help decrease overfitting, and improve prediction in general. The out-sample RMSE does increases slightly approaching leaf size of one, so it is inconclusive whether overfitting can be entirely eliminated with this particular bagging size.

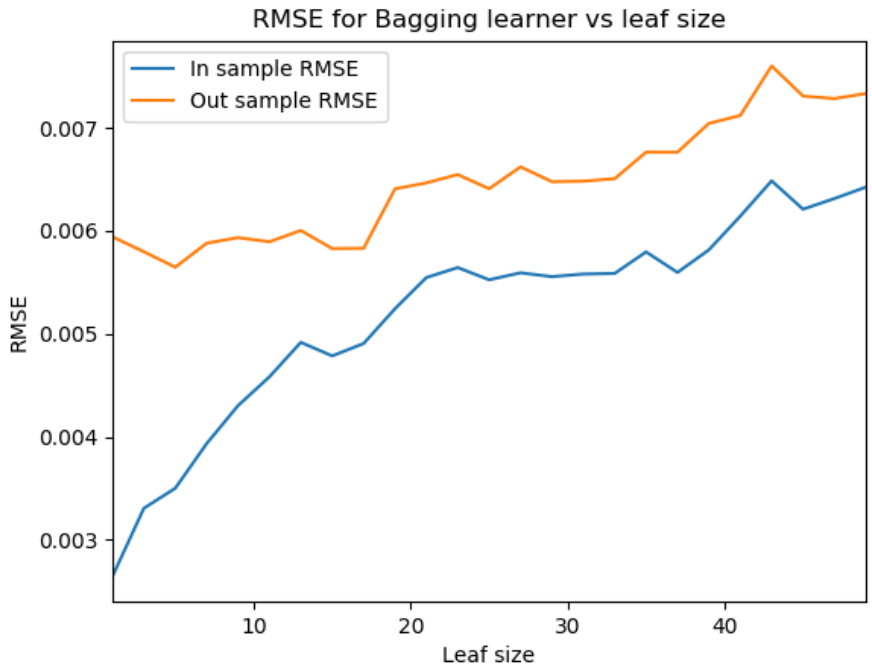

Figure 2: RMSE for Bagging Learner vs Leaf Size

Q3: Classic decision tree vs random tree

The experiment: The Istanbul dataset is used to train and test both Decision Tree and Random Tree learners using the following parameters.

- Varying leaf size from 1 to 50 with step size of 2.

- Mean (out sample) absolute error is calculated at each leaf size for each learner.

- Training time and query time at each leaf size for each learner are measured.

Result (Figure 3, 4, 5):

Figure 3 indicates that Decision Tree results in better accuracy than random tree, with mean absolute error lower than Random Tree across all leaf sizes, except for leaf size of 1. However, as noted in question 1, very small leaf sizes are when overfitting occurs and it affects the out-sample performance of Decision Tree.

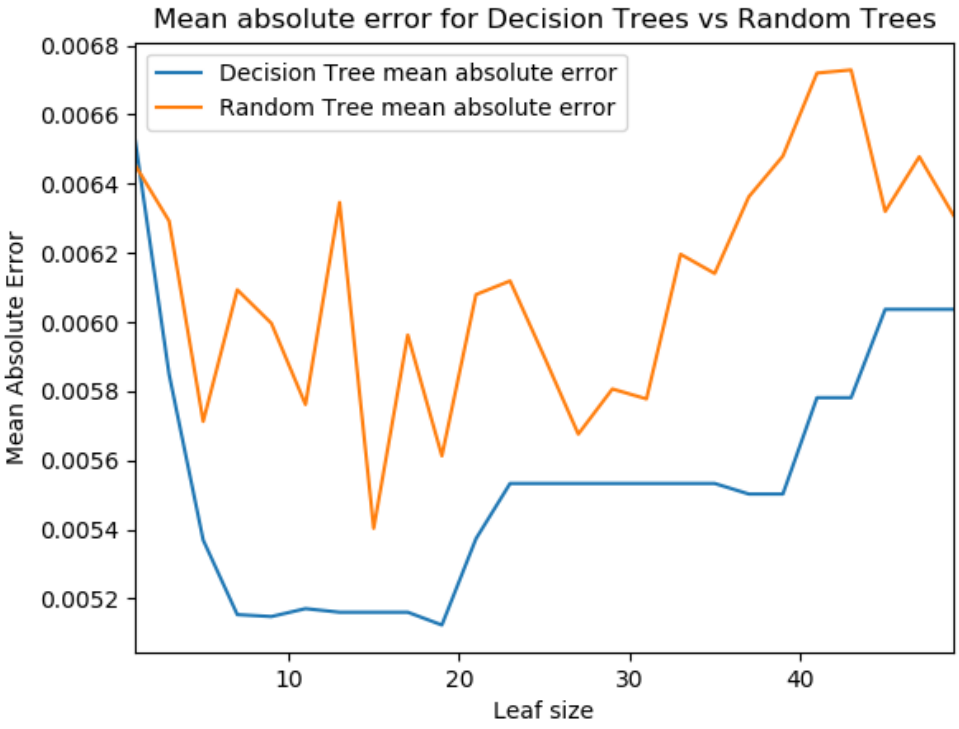

Figure 3: Mean absolute error for Decision Tree vs Random Tree (out-sample)

However, Random Trees outperform Decision Trees in terms of training time. Decision Trees' training time increases exponentially when leaf size approaches 1; whereas, Random Trees' training time remains flat with only slight increase at leaf size of 1.

This is due to the fact that Decision Trees have to calculate the correlation across all data points when they build a new node. Whereas, Random Tree picks a feature to split at random.

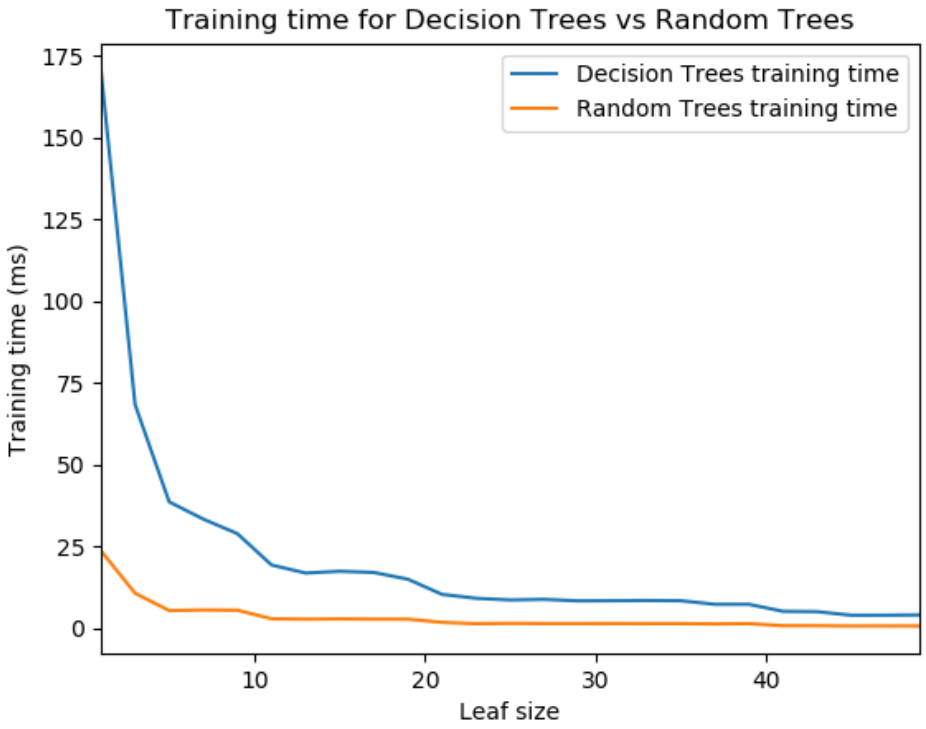

Figure 4: Training time for Decision Trees vs Random Trees

Query time, however, does not vary much between Decision and Random Trees (figure 5).

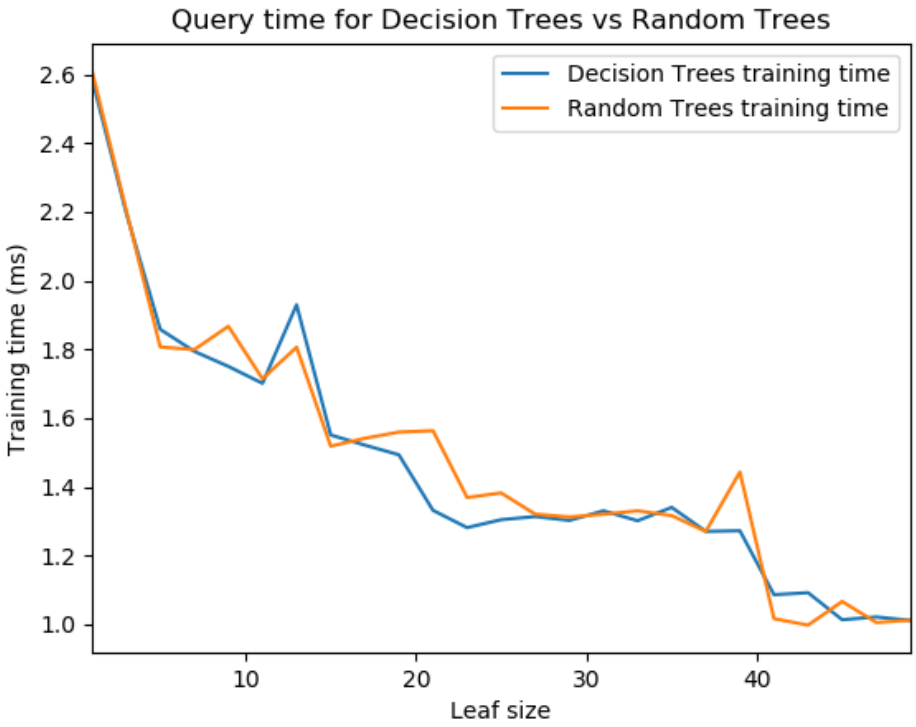

Figure 5: Query time for Decision Trees vs Random Trees

Conclusion: Decision Trees are superior in terms of accuracy. However, it does that at the cost of slower training time. Random Trees sacrifice accuracy for training time. However, as can be seen from questions 1 and 2, Decision Trees suffer from overfitting at smaller leaf sizes, which can be rectified with bagging. As bagging also improves overall accuracy, it is suggested that Random Trees can be used in conjunction with Bagging to achieve both accuracy and faster training time, while minimizing overfitting.