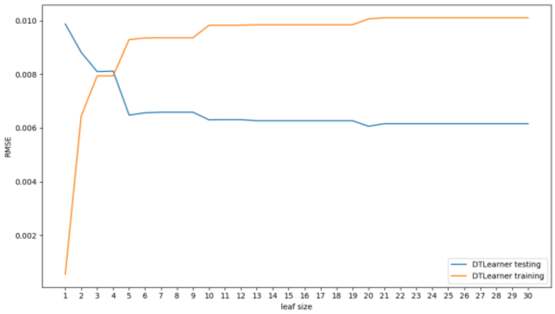
Machine Learning for Trading: Assess Learner

Name: Chiamin Wu

GT Account:cwu392

Q1: Does overfitting occur with respect to leaf_size? Consider the dataset istanbul.csv with DTLearner. For which values of leaf_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging).



A1.1:

Based on my simulation result, I found when **leaf size greater than 4**, my training accuracy will start to increase and my testing accuracy will begin to decrease.

A1.2:

Overfitting means when we observe our training accuracy is still decreasing but our testing data have equal or worse accuracy. In this case, when we **decrease leaf size from 5 to 3**, we can observe our training data have better accuracy but our testing data become worse.

Q2: Can bagging reduce or eliminate overfitting with respect to leaf_size? Again consider the dataset istanbul.csv with DTLearner. To investigate this choose a fixed number of bags to use and vary leaf_size to evaluate. Provide charts and/or tables to validate your conclusions.

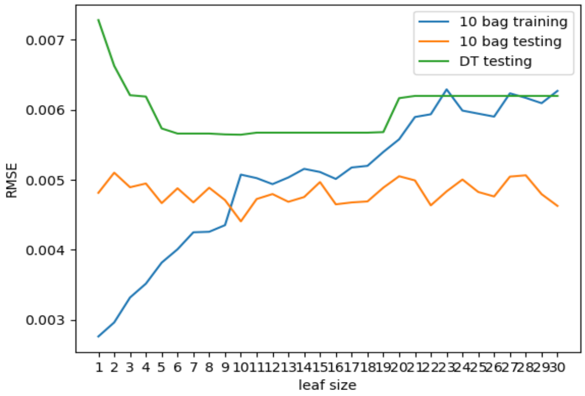


Fig2.1: Bag Learner with 10 Bags of DT Learner

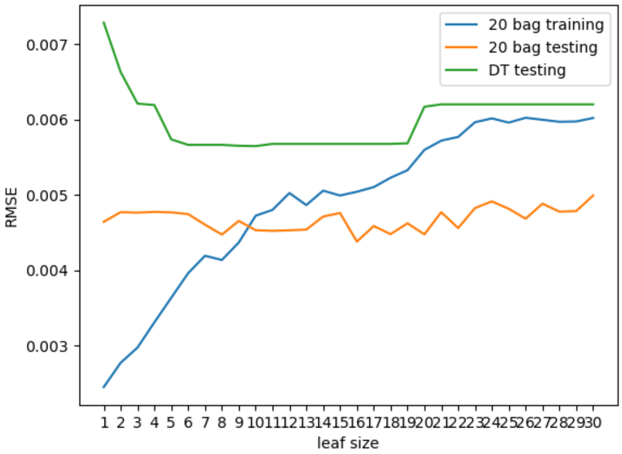


Fig2.2: Bag Learner with 20 Bags of DT Learner

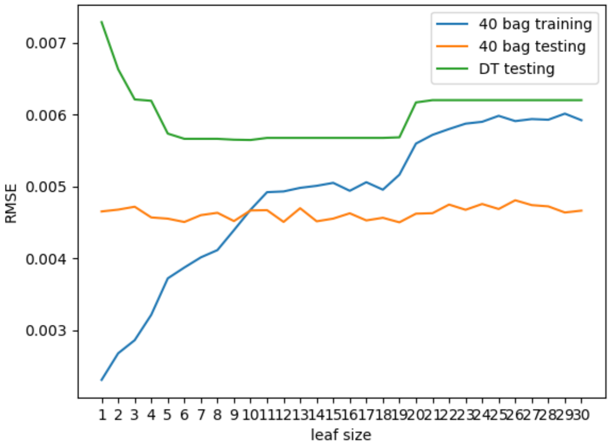


Fig2.3: Bag Learner with 40 Bags of DT Learner

A2:

The following 3 figures are Bag Learners with **10, 20 and 40 bags** of DT Learners and **sweep leaf size from 1 to 31**. We can easily found that with more bags, the fluctuation phenomenon decrease. The Green Line is the Baseline, which is DT Learner. From Figure 1, we can see Bag Learner's fluctuation is worse than DTLearner. However, with 40 Bags in Figure 4, we can have observe **flat curves** in both training and testing data with a better accuracy.

Q3:

Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other?

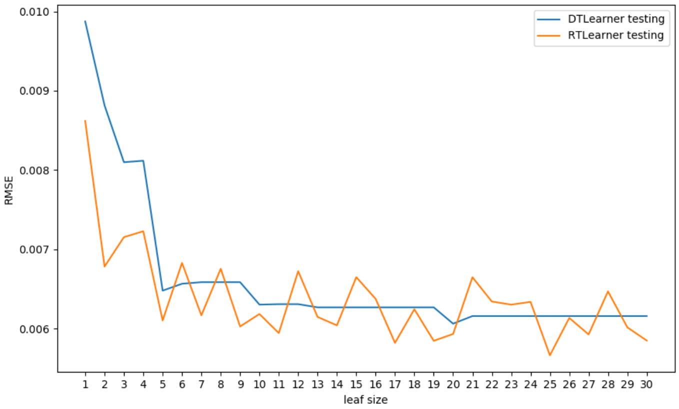


Fig 3: DTLearner vs. RTLearner

A3:

Theoretically, RTLearner has better performance benefit from its bias less. In this case, selection with randomness could avoid bias accumulation and contribute to the average effect of the data noise. However, we can observe from Figure 3. If we use small leaf size (<=5), we can see the testing result of **DTLearner is about 10% worse than RTLearner.** This is because with small leaf size, each training point occupies its information and noise, which will diminish our performance. However, if we choose leaf size larger than 30, then the performance of **DTLearner is much smoother** than RTLearner. In conclusion, if we have small dataset, we could choose RTLearner to relieve the noise/bias from our training data. On the other than, if we have some very accurate data with less or no bias, or we have a relatively great leaf size, then DTLearner should have a comparative performance but smoother curve than RTLearner.