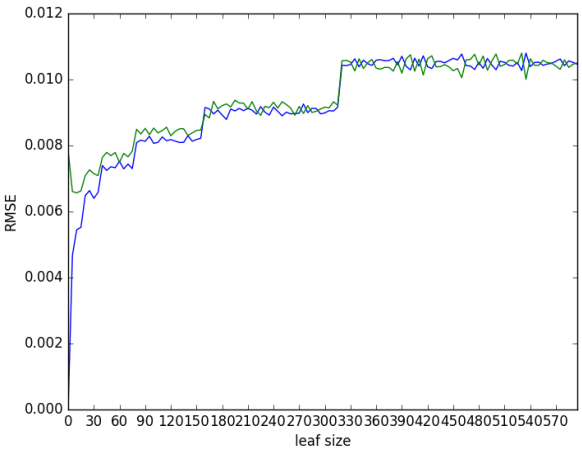
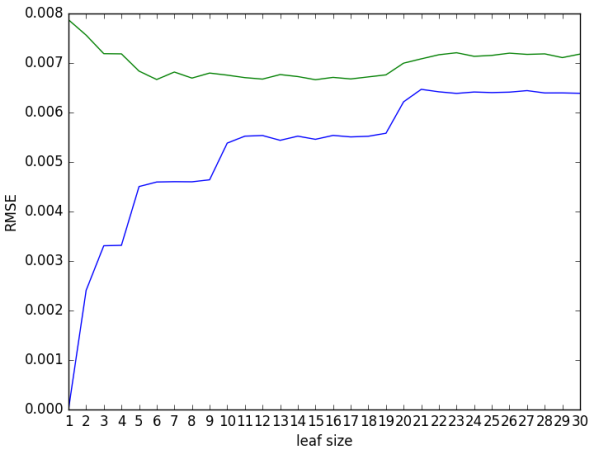
Shu-Ho Chou schou33

CS7646 Machine Learning for Trading HW3 REPORT

1. Overfitting with respect to leaf_size using Decision Tree.

Experiment Details: Since in reddit (https://www.reddit.com/r/cs7646_fall2017/comments/71gt1g/out_of_sample_error_for_istanbul_dataset/), the professor recommended us to randomly pick training and testing data, I tried to shuffle the data, and get the results. However, due to the small size of Istanbul.csv, the result is very volatile. Hence, I do the experiments 50 times, and in each time, I shuffle the data and decide the training and testing data, and then train the decision tree. Finally, average the RMSEs and get it on the chart.



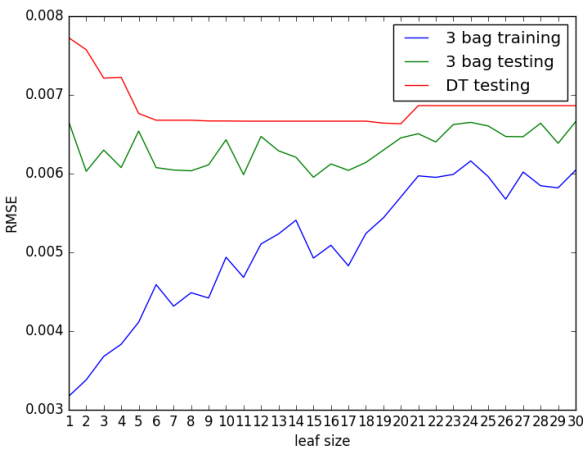


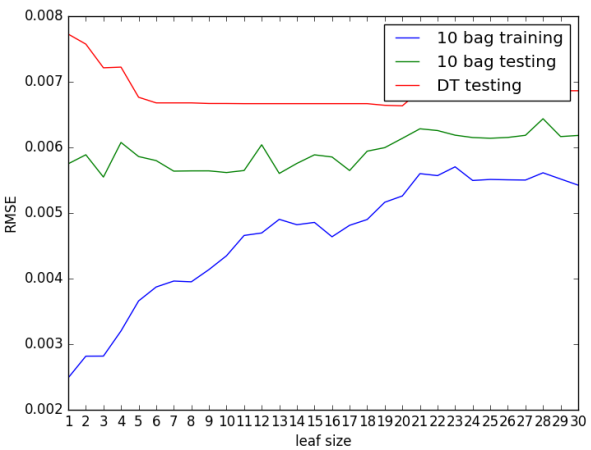
From the images above, we can see that overfitting happens when leaves is smaller than 30. So we can omit all sizes bigger than 30. The green and blue lines are testing and training curve respectively.

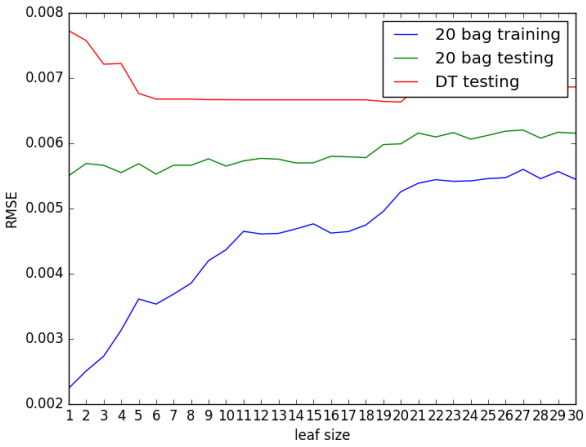
See the graph on the right from right to left, which has the range of leaf size from 1 to 30, we can clearly see that when the leaf size drop to 19, RMSE for testing data already get to near the minima, and when leaf size goes from 6 to 5, testing RMSE goes up drastically, as the training RMSE goes all the way down. I would say for this task, leaf size 19 is the best hyperparameter, however, 5 is the point where it really starts to overfit.

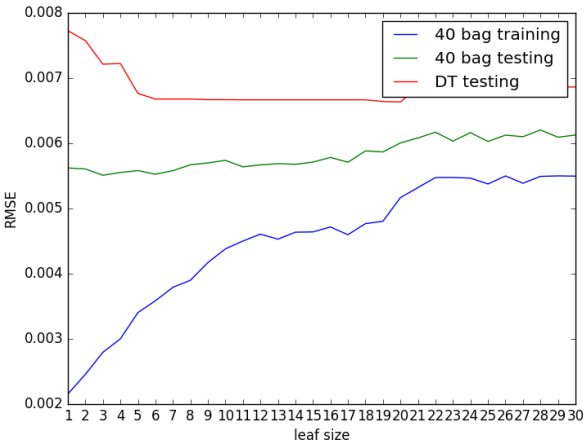
2. Can bagging reduce overfitting with respect to leaf size?

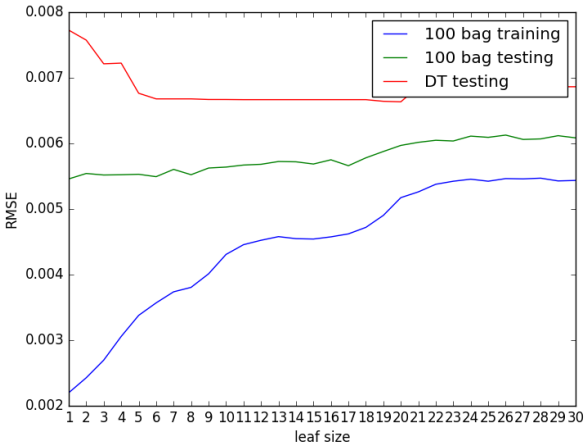
Yes, it can reduce overfitting, if not eliminate. In my experiment, I shuffle the data once and train and test for leaf sizes from 1 to 30 (same as Q1), and I fix the bag sizes as 3, 10, 20, 40, 100 respectively, for the 4 graphs.







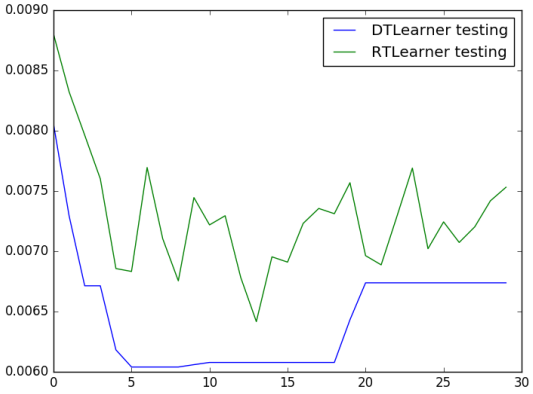




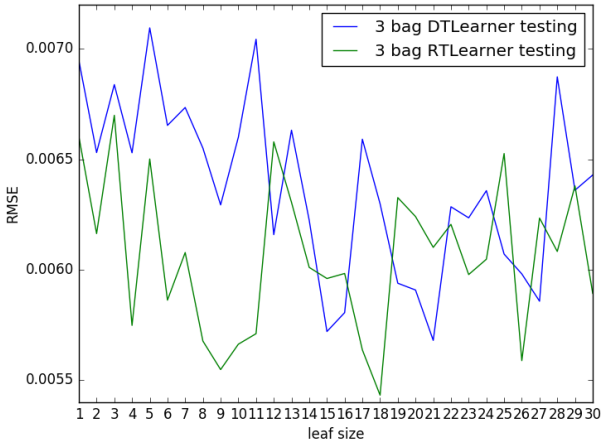
If we see from the graphs, when the bag number is small, there are some fluctuation, but for bigger bag number, there is no overfitting, since bag learner randomly samples points and can thus reduce the biases created by the decision tree, which is what ensemble model aims to do. So as the decision tree learner overfits at leaf size of 5, bag learner doesn't overfit. However, it also suggests that you can stop at leaf size about 30, since the performance for testing doesn't really improve much from 30 to 1.

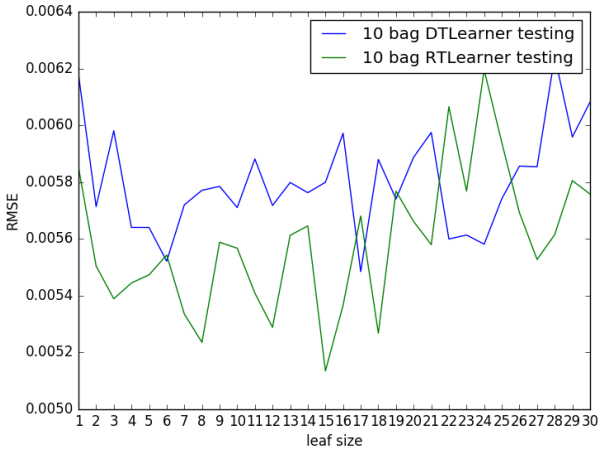
3. Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other?

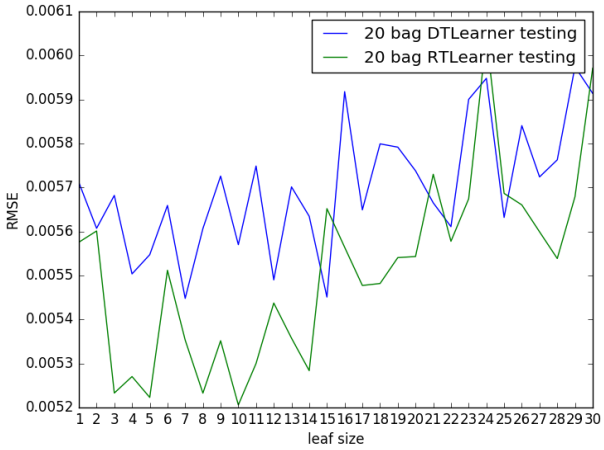
When testing Istanbul.csv, we can clearly see that DTLearner performs better than RTLearner when bag is not used. The results on the right side is generated by shuffling the data before training.

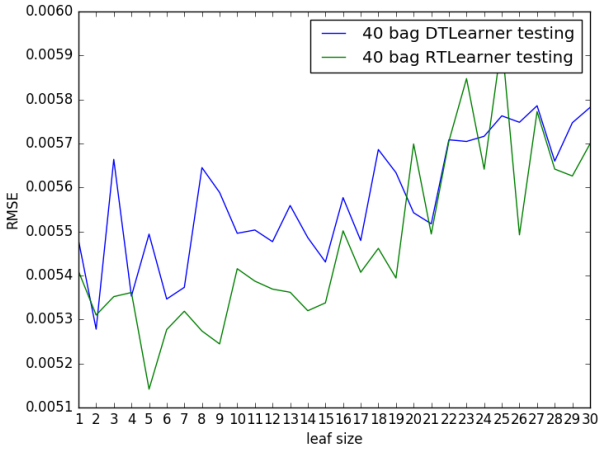


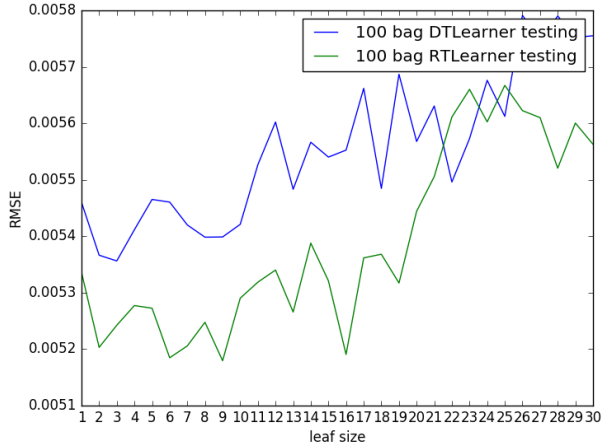
However, when bag is involved, RTLearner can perform better than DTLearner. The results below show the decision tree and random tree comparison with bag size 3, 10, 20, 40, 100.



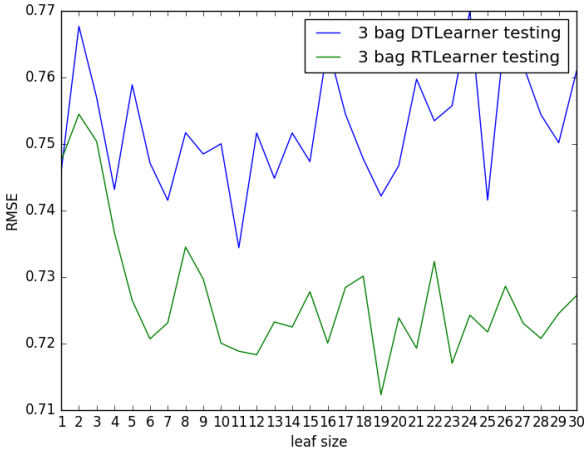


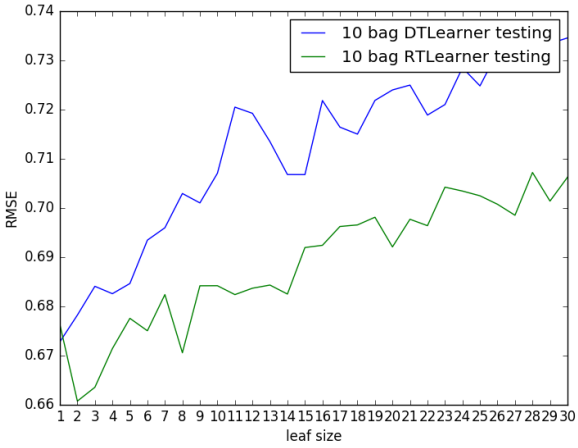


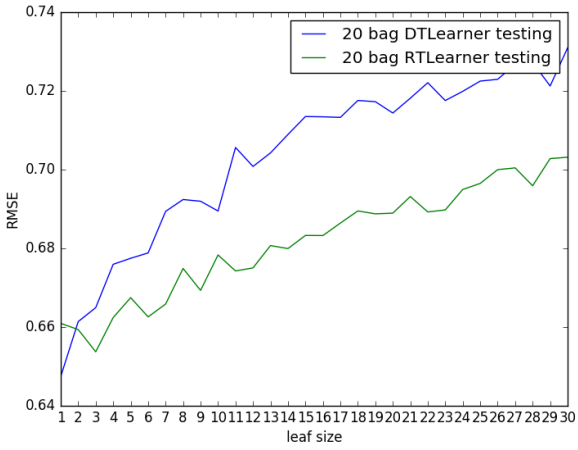


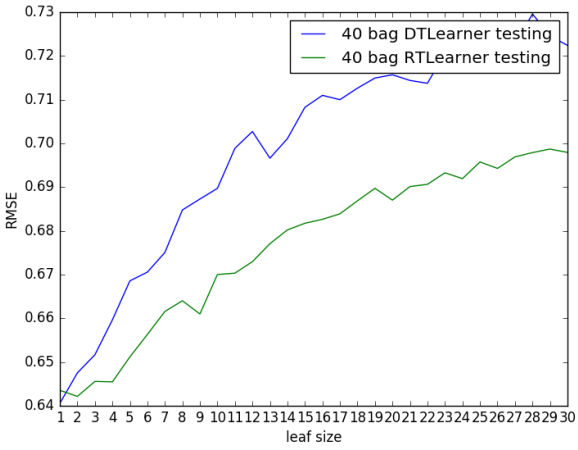


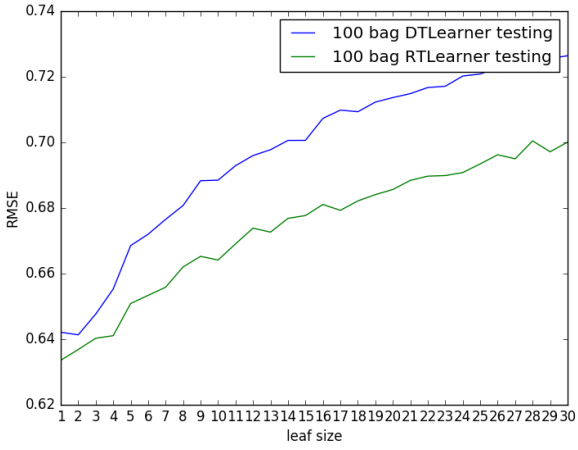
From the results above, we can see that as the bag size increases, bag of random trees perform better and better, and beat the result of bag of decision trees. Since only testing Istanbul.csv might be not convincing enough, I also tried on the biggest data *winequality-white.csv.*











And they again show that bag of Random Tree Learners perform better than bag of Decision tree learners, not by the characteristic of the small dataset.

Analysis: When there are no bags, Decision tree learner better described what the training set looks like, and so it performs better. However, describe better means higher chance to get overfit,

so when bags of learners are combined, the biases of decision trees aggregate greater than random trees because random trees can somewhat eliminate the biases created by one another.