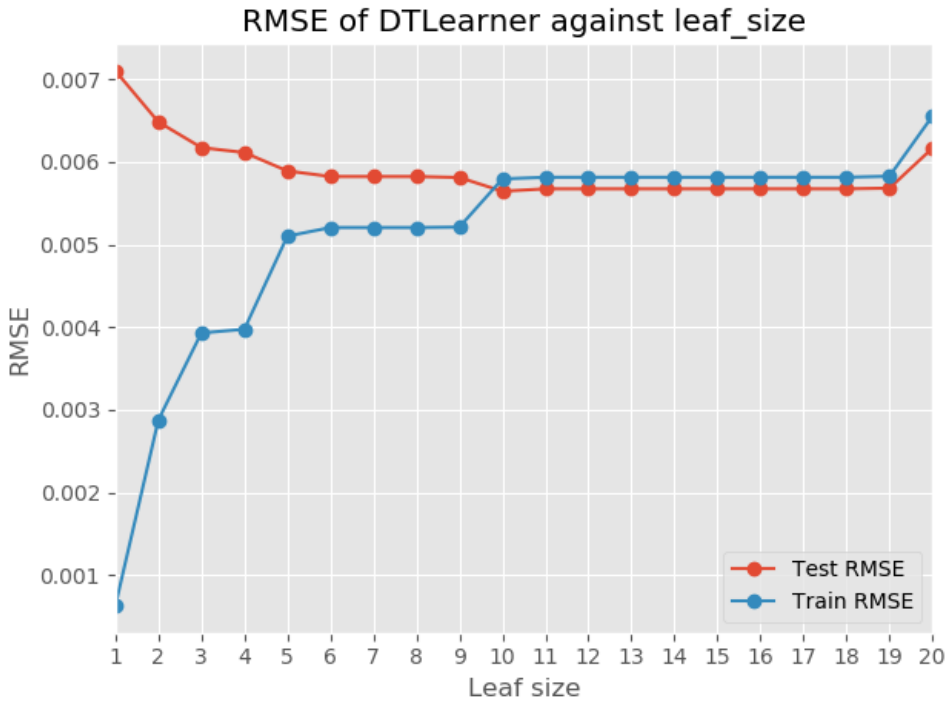
CS7646 Machine Learning for Trading - Spring 2019
Assess Learners

Sylvain Marchienne
(`smarchienne3`, `903430342`)

February 7, 2019

Question 1

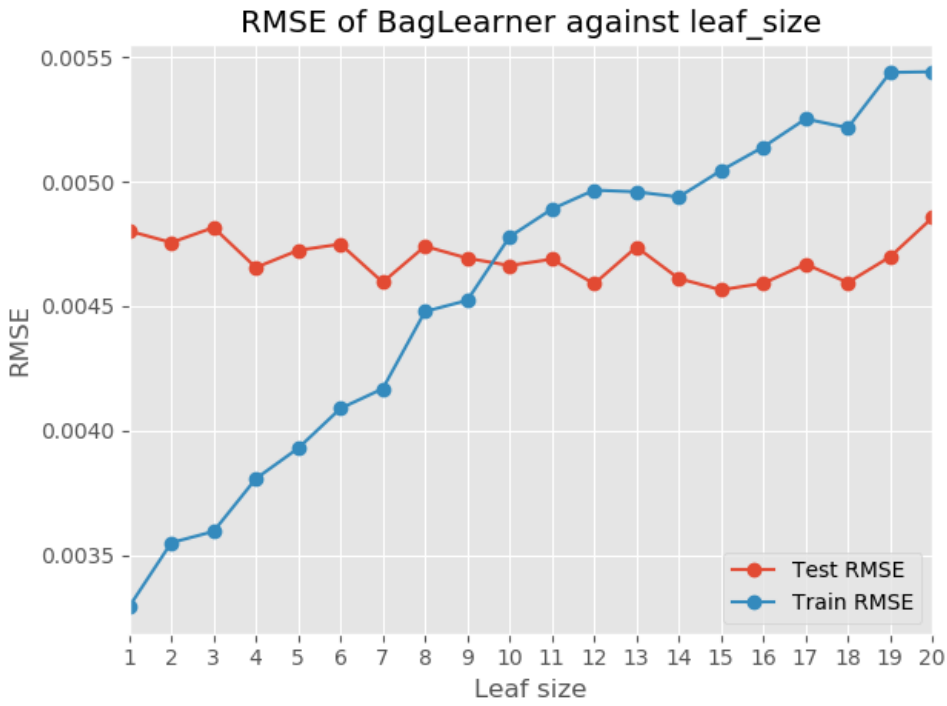


We plotted the root mean square error (rmse) of the learned `DTLearner` model against the parameter `leaf_size` on the istanbul dataset. As we can see, initially with the default value `leaf_size = 1`, the training rmse is very low while the test (out-of-sample) rmse is the highest. From here there are two things to notice. When we increase `leaf_size`, the training error increase and the test error decrease. At `leaf_size = 10`, the training error is higher than test error. If we increase more, the errors are very stable and start to increase both at `leaf_size = 19`.

This is indeed the expected behaviour as `leaf_size` is a regularization parameter. Indeed, at `leaf_size = 1` the training set is perfectly fitted but do not generalize at all. For this reason the training error is almost zero while the test error is the highest. It is said that we overfit when the training error decreases but the test error increases. This is exactly what happen when `leaf_size`

goes to 1. By increasing `leaf_size` we allow less flexibility to the tree then we push for more generalization and less overfit. It works up to `leaf_size = 19` where the error starts to rise.

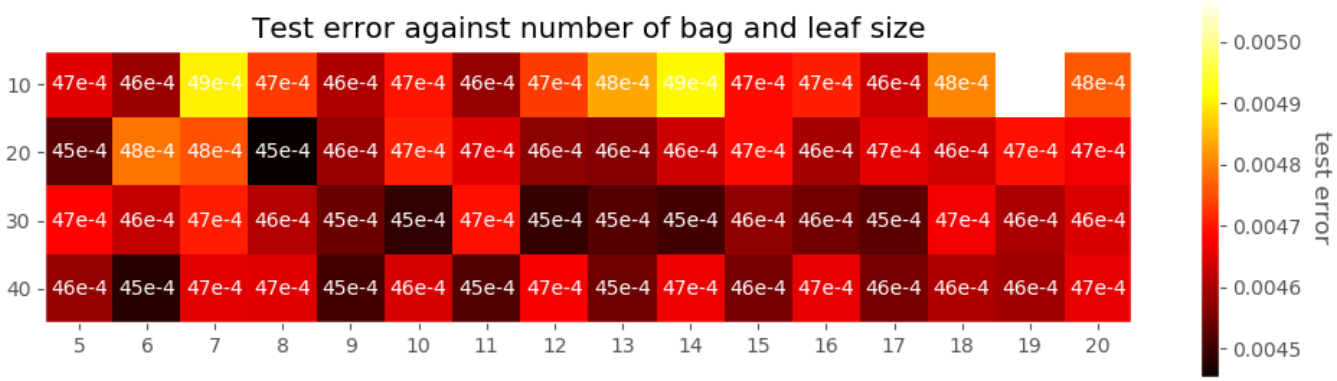
Question 2



This figure is the result of a bagging training with 20 bags of `DTLearner`. The first thing to compare is the maginitude of the error compared to the `DTLearner` we trained alone at the previous question. Here we can see the test error is at most around 0.0047 while the lowest error was around 0.0057. The bagging strategy is definitely better than a learner itself.

Furthermore about overfitting, as we can see varying the the regularization parameter `leaf_size` has almost no effect on the test error. It remains quite stable. The hypothesis is that bagging is much less sensitive to overfitting because of its randomness characteristic. Indeed, building the tree with a bootstrap sampling is actually injecting generalization in the final bagged model. Thus even at the worst generalization value `leaf_size = 1`, the test error is already quite low and keep constant with `leaf_size` constant.

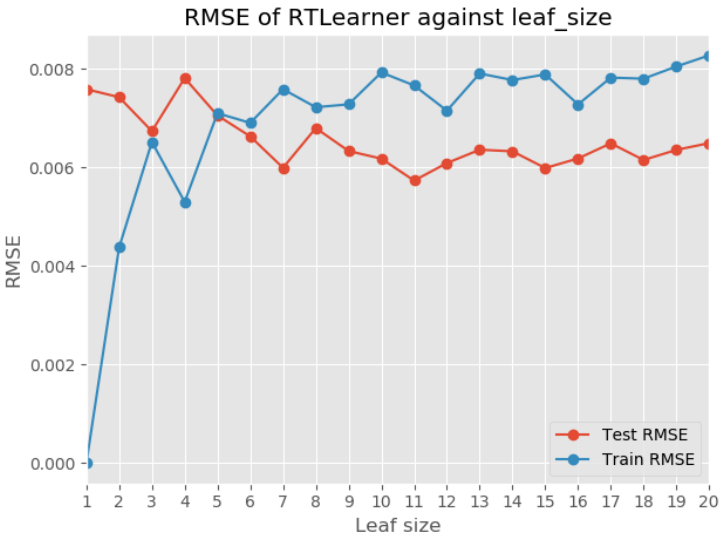
Apart from the robustness towards overfitting, performances (test error) are simply better than what we can expect from a standalone tree because we add more trees and it reduces the bias. In the mean time, bagging sampling method also decorrelates the trees and reduces the variance when we average the trees. We end up with a more complex model but less sensitive to overfitting.

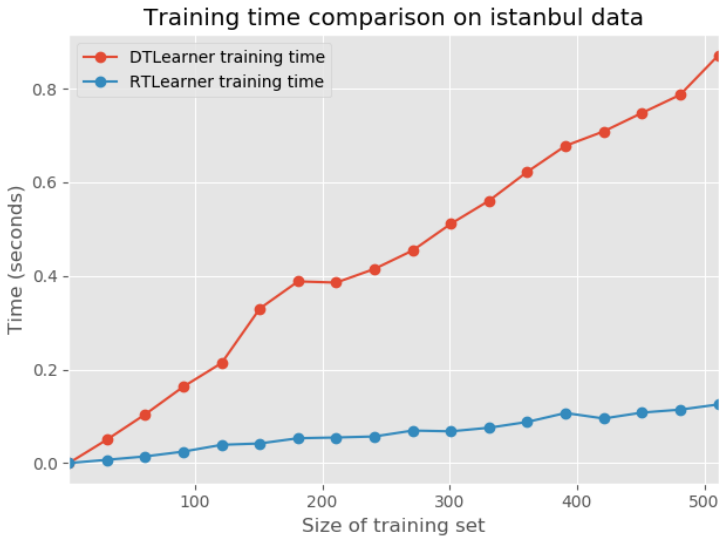


Another interesting thing to compare is how the performances evolve with both the leaf size and the number of bags. We plotted this heat map representing the test error. The lines represent the bag size (10, 20, 30, 40) and the columns the leaf size (5 to 20). Black cells are the best (the minimum test error).

We notice that if we increase the bag size, we must balance with an increase of leaf size. Indeed, increasing the number of bags make the model more complex then prone to overfitting. Increasing the leaf size fix overfitting. As we go from the first row to the last one, the darkest cells are (on average) shifting to the right.

Question 3





On the first figure (top left), we simply plot the performances of the random tree learner to compare with the question 1. As we can see, it is also sensitive to overfitting and the leaf size must be large enough to balance. Neither it is a better model in terms of rmse. Relatively speaking to bagging, it is similar to the correlation tree alone but slightly worse.

On the second figure (top right), we plot the time required to train the models against the size of the training set. The discrimination is obvious: the random learner is way more fast because it doesn't have to compute the correlations between the features and the target variable. The correlation decision tree has clearly a linear time complexity of the size of the training set, with a much bigger slope than the random tree learner.

Finally, we plot the size of the tree (number of nodes) against the training set size. What we can see is that the space complexity in both models is linear with only a few variations between

the two in higher training set sizes.

Conclusion

From this analysis, we learned that bagging is robust to overfitting, both the decision trees with correlation and the random selection of features are almost equivalent in terms of performances (in our case). Moreover the training time of random decision trees is very low compared to the correlation one. Thus why not combining random decision trees with bagging? This is very close to what Random Forest is. Random Forest is a bagging method but when building the decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors. But the choice of which feature to split on among the m ones is deterministic (using information gain metric such as Gini or Entropy for classification and mean square error for regression).