CS 7646: Machine Learning for Trading September 24, 2018

Assess Learners

Instructor: Tucker Balch *Bhuvesh Kumar*

Experimentation setting:

All of the experimentation in this report has been done on a Quad Core Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz CPU with 8GB DDR2 RAM. For all experiments, Istanbul.csv was used as the dataset with first 60% of the points as training set and the remaining as test points. The dataset has 536 data points and 8 features for each point. The train set had 321 points and the test set had 215 points.

1 Analyzing over-fitting in Decision Tree Learner

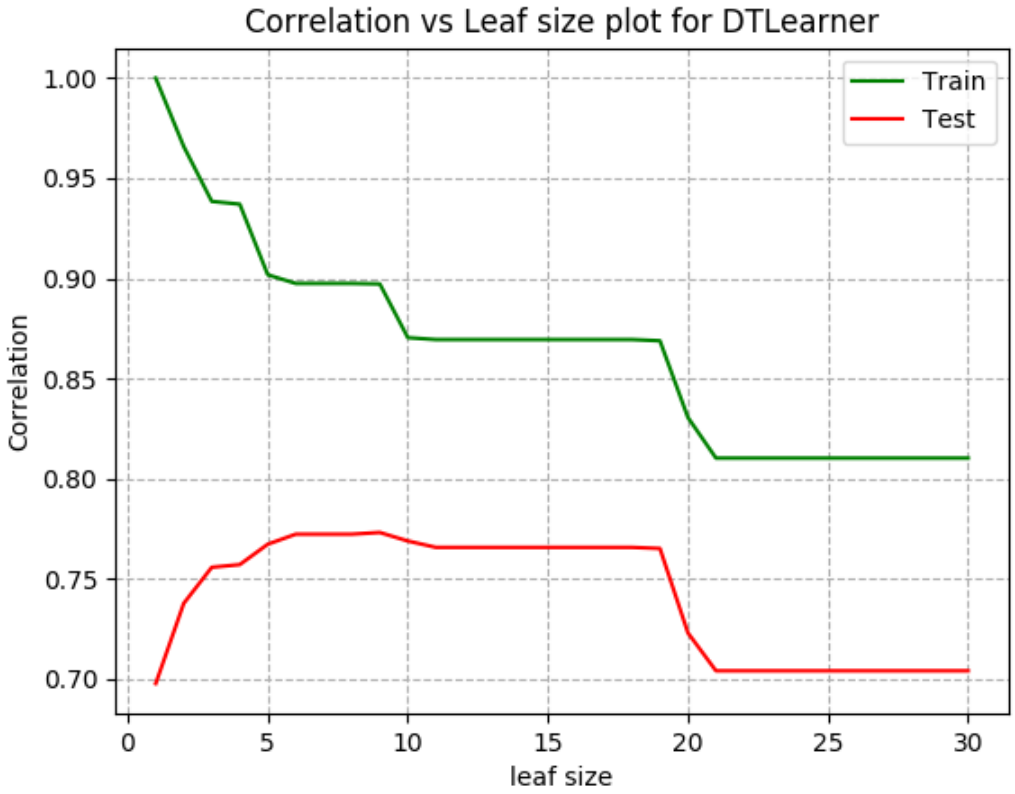

Figure 1: Correlation vs Leaf size for DT Learner on Istanbul.csv data set

From Fig 1, we can see that as the size of leaves decreases the correlation of the train set increase, but it does not necessariliy increase for the test set. This is because even the empirical error on the train set is decreasing, the generalization error is not necessarily decreasing. One of the reason for this kind of behaviour can be overfitting of the data by the model. As the leaf size becomes smaller, the tree grows and the model complexity also grows and it over-fits the data. To check if overfitting

is happening, we need to look at RMSE on the train set and compare it with the test set. In the

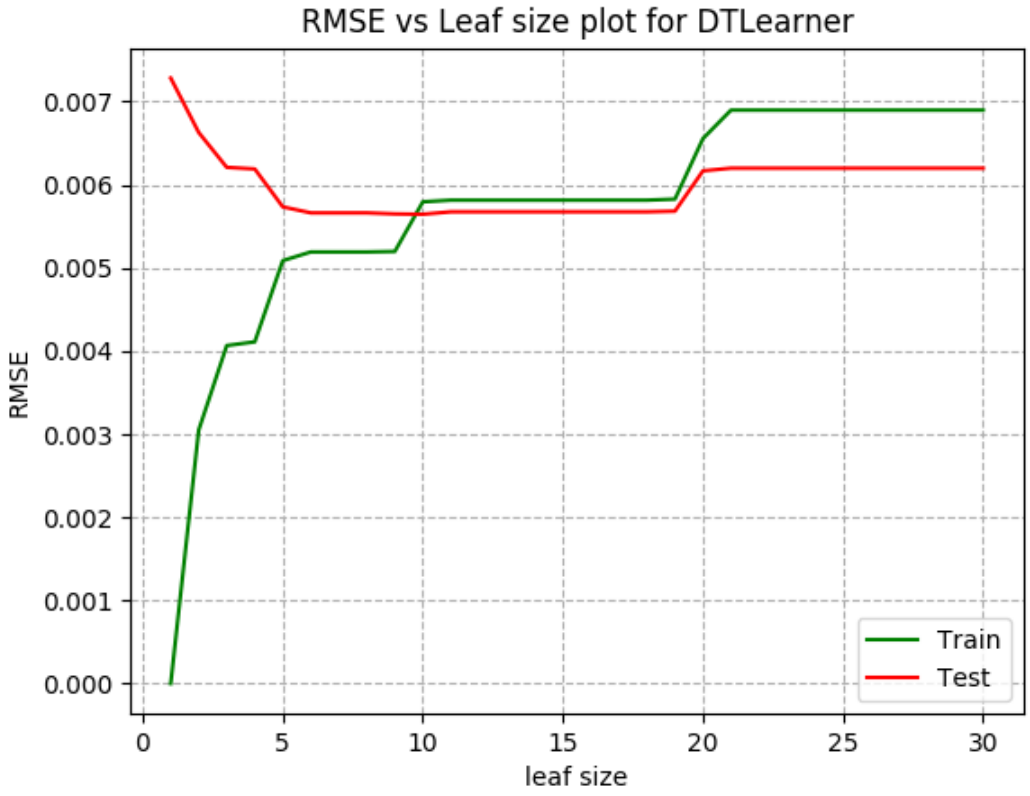

Figure 2: RMSE vs Leaf size for DT Learner on Istanbul.csv data set

fig 1, we can observe that for leaf sizes smaller than 10, the RMSE for the test set is much higher than the RMSE for train set. This shows that over fiting is happening in Decision Tree learner for smaller leaf sizes and in this case, it happens when the leaf size is smaller than 10. The philosophical principal of Occam's Razor rightly predicts overfitting in Machine Learning models, i.e that simple models that nearly explain the data are better than very complex models which perfectly explain the data.

2 Effect of Bagging - Does it reduce/remove over-fitting

In this experimentation setting, we analyze the effect of bagging and observe how it changes the behavior of the Decision Tree learner.

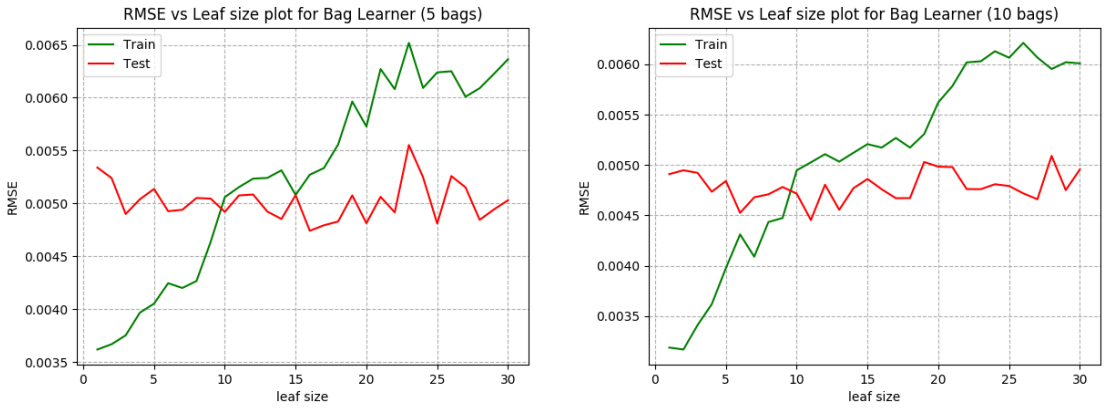

Figure 3: RMSE vs Leaf size for Bag Learner (with DT Learners) on Istanbul.csv data set

If we see in fig 3, since bagging is a randomized algorithm, the performance of the model can be slightly erratic, hence to properly study the model performance and behaviour, it makes sense to study the expected performance of model. Note that by expected performance, we don't mean that the performance of the expected predictions, but how bagging performs on expectations. Hence it makes sense to repeat the experiments for such random experiments and then observe the average performance of the model but note that we are not taking the average of predictions from the multiple predictions as that would be equivalent to increasing the bag sizes. So for this experimentation, we repeated the experimentation 10 times to get the expected performance of the bag learner.

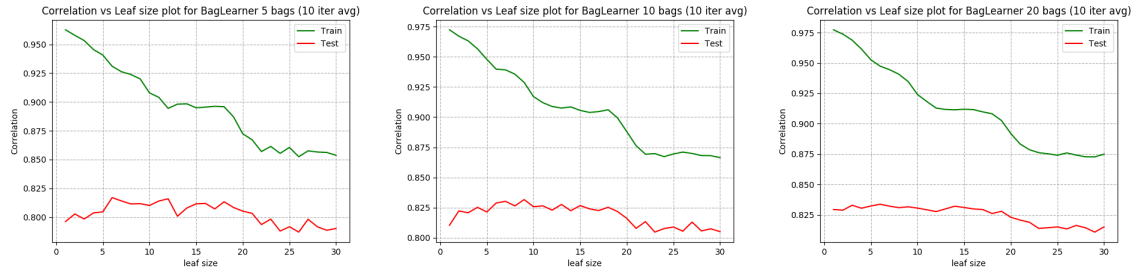

Figure 4: Correlation vs Leaf size for Bag Learner (with DT Learners) on Istanbul.csv data set

If we compare fig 4 with fig 1, we can clearly see that for bagging, the correlation is higher for both train and for test sets for bags of size even 5,10, or 20. We can see that as the number of bags are increasing, the graphs are becoming more smoother as well. because as the number of bags increases, the model takes us closer to the expected result of the distribution. We can also see that the correlation for the test predictions is also very high even for slightly larger leaf sizes, this is because of the randomness in the data sampling, the points which reach the leaves are different in different bags and hence we get good performance even for large leaf sizes. So to observe over - fitting, we observe the RMSE vs leaf size graph:

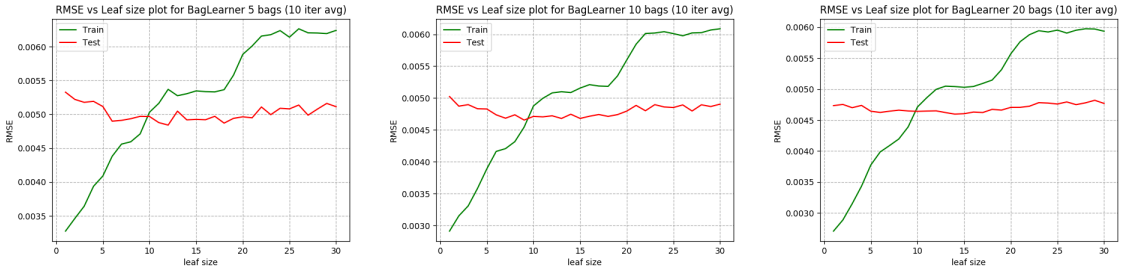

Figure 5: RMSE vs Leaf size for Bag Learner (with DT Learners) on Istanbul.csv data set

We see that in fig 5 that after bagging, for small leaf sizes, the RMSE difference is not very high in the Bagging case as compared the difference in RMSE for test and train in DT Learner (fig 2). Hence the difference in RMSE value has shrunken. Also we can see that the RMSE value for train is increasing as it should as leaf sizes increases, but for test it is not increasing that much which shows that that distribution learned when leaf size is very small is also similar to the distribution with slightly larger leaf sizes, hence denoting a clear reduction in over-fitting by the model in case of increasing model complexity, i.e. decreasing leaf size. Still, we cannot claim that bagging has completely gotten rid of over-fitting because there is still difference in the RMSE value of train and test in case of small leaf sizes. Hence the model does over-fit for leaf sizes smaller than 9 but the extent of over-fittting is less than in the case of Decision trees and the model is also more robust to change in leaf sizes for changes in test RMSE. So, yes, bagging does reduce over-fitting.

3 Decision Tree Learner VS Random Tree Learner

In this section we compare the performance of the DT Learner with RT learner. Similar to the case of Bagging, RTLearner has randomness component to it. Hence it makes sense to compare the average performance of the RTLearner with the DTLearner. We would again like to reiterate that we are working with the average performance and not the performance of the average. Working with the performance of average of RTLeaner would just be like comparing a bag of RTlearners with DTLearner, which we are not doing. So, for all these experiments, we use the same dataset. All the experimens were repeated 20 items and the average of the performance has been taken. We also measure the time taken by the learners and for performing that experimentation, even the experiments for DTLearner were repeated 20 times and the average of the performance was taken. This ensured that noise in the time because of other processes running on the computer could be minimized.

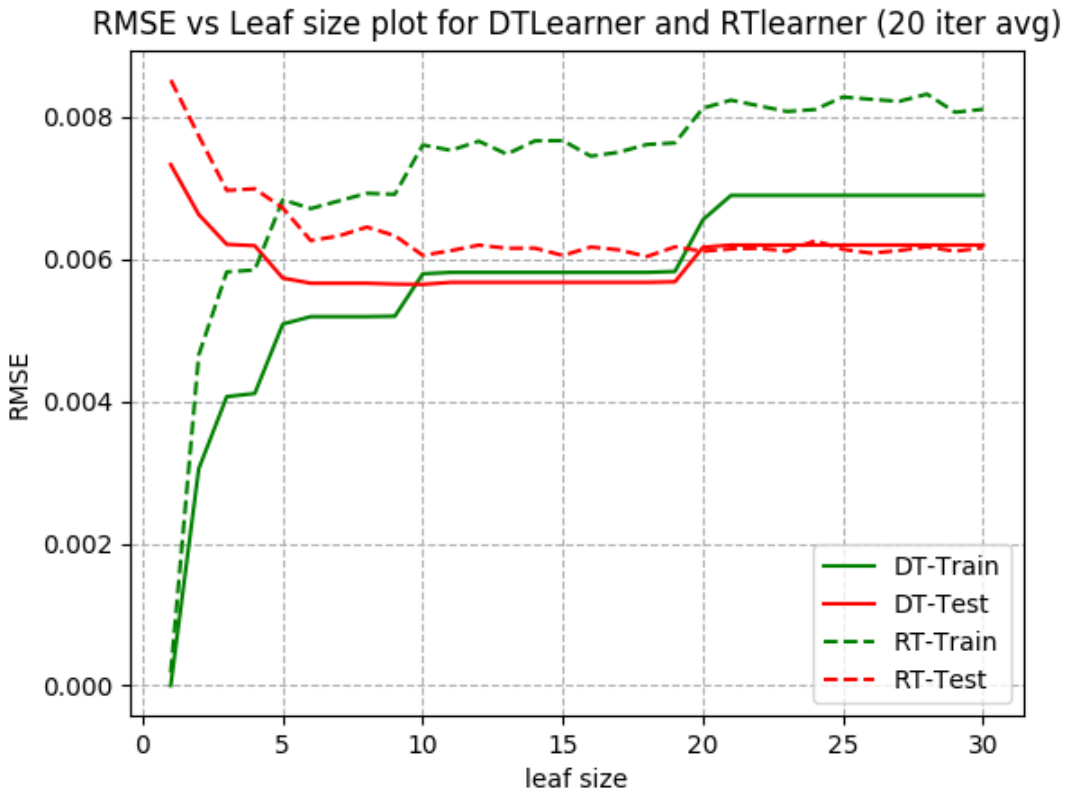

Figure 6: RMSE vs Leaf size for DTLearner and RTLearner on Istanbul.csv data set

For small leaf size, we can the test error for DT learner is slightly less than RT learner. The difference in the test error and train error is higher in DT than RT, which may indicate that DT does more over-fitting than RT learner. Looking at the RMSE value we can see that the DT learner is over-fitting the train data for leaf sizes even as big as 10, but for RT learner, the RMSE for test and train becomes small as leaf size reaches even 5. Hence we see that RT learner does less over-fitting than DT Learner, hence RT Learner is better to avoid over fitting.

Now comparing the time taken by both these models.

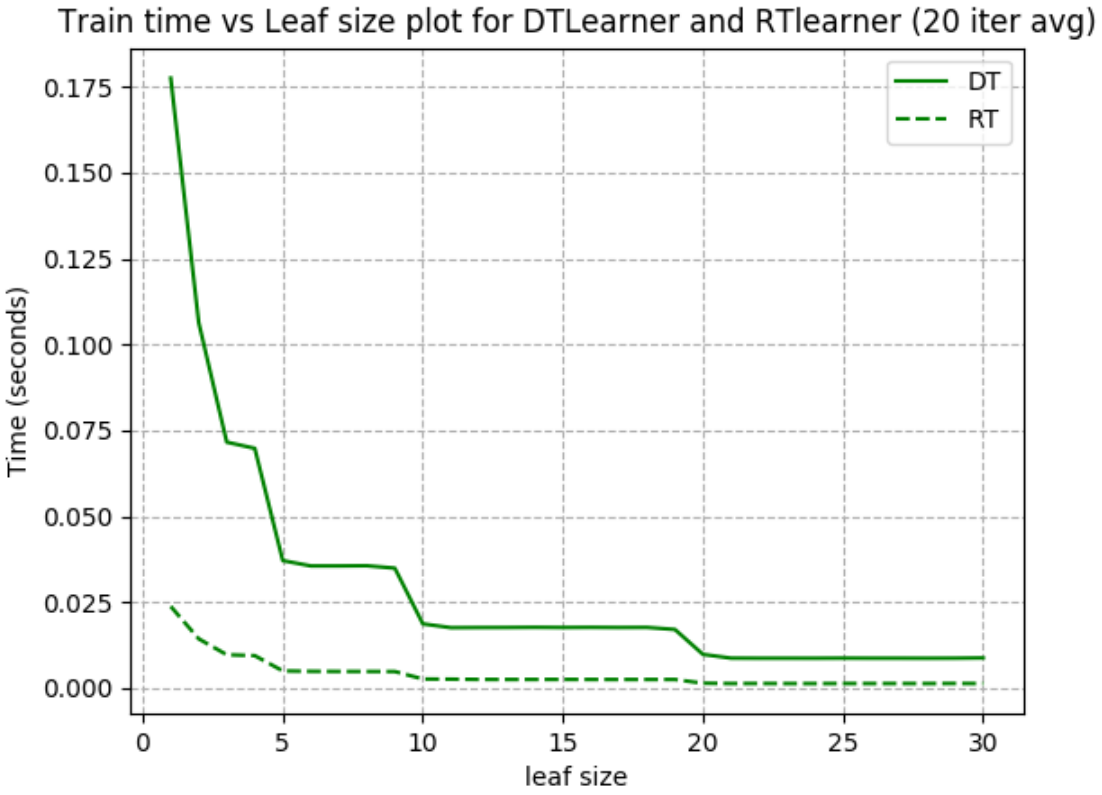

Figure 7: Train time vs Leaf size for DTLearner and RTLearner on Istanbul.csv data set

We observe from fig 8 that the DT Learner takes a lot more time than RT learner becasue the step where we find the feature with maximum correlation with the labels is a computation heavy step and hence RT learner is more quicker in training because the split feature is just chosen randomly.

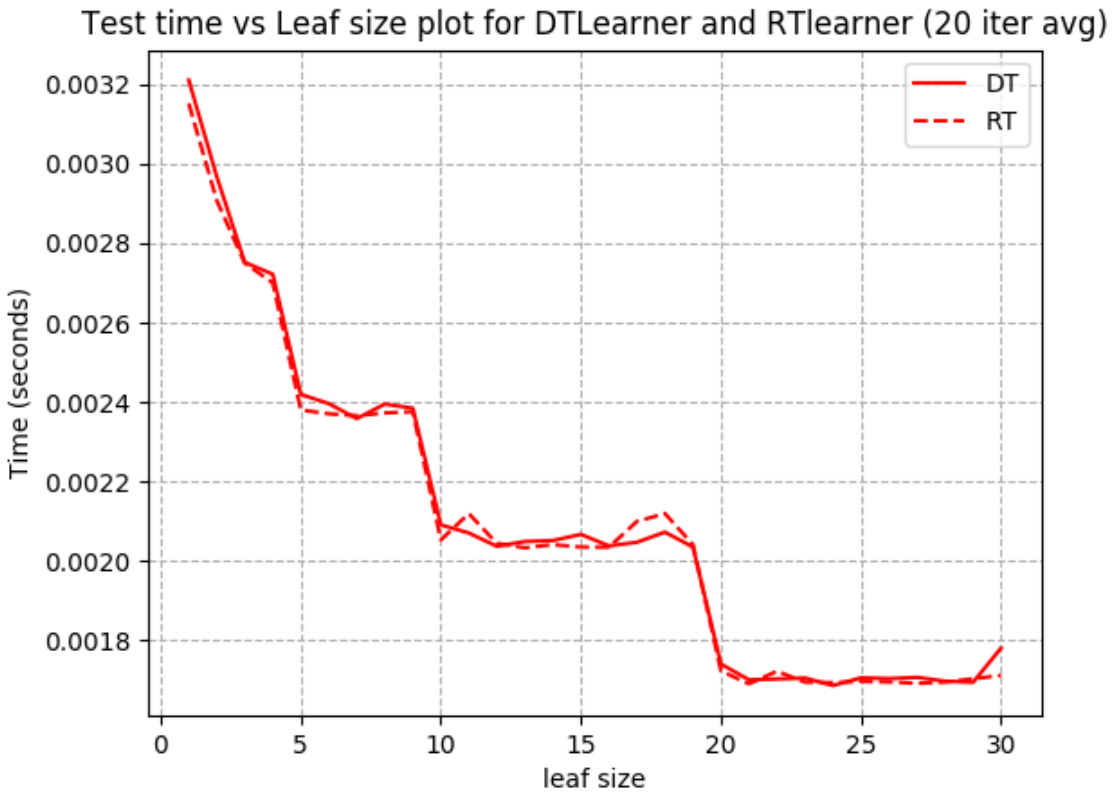

Figure 8: Test time vs Leaf size for DTLearner and RTLearner on Istanbul.csv data set

As expected, the test time is similar for both DTLearner and RTlearner as the test time only depends on the tree height which will be similar for both the learners when the leaf sizes are similar.

We know that RTLearner is a random learner, hence it is expected to be more robust to noise and corruptions in the data. To test this hypothesis, we corrupted 20% of the train data of Instanbul.csv. Instead of the original feature values, we randomly replaced 20% of the rows with features generated from a normal distribution with high variance. Hence, this will act as corruptions. We trained both the models on this new train data and observed the performance on the clean test data and this new train data as well.

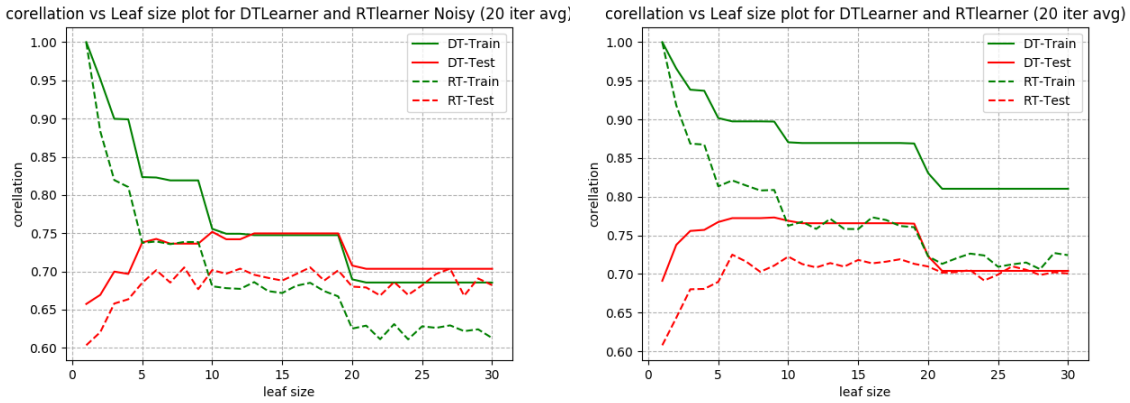

Figure 9: Correlation vs Leaf size for DTLearner and RTLearner on noisy and clean Istanbul.csv data set

In figure 10, the left plot is for noisy (corrupt) data and right plot is for clean data. We see that there is a sharp decrease in the correlation value of DT Learner when corruptions are added whereas the correlation of RT Learner have decreased less. This shows that RT Learner is slightly more robust to noise. Still the difference is not as drastic as expected.

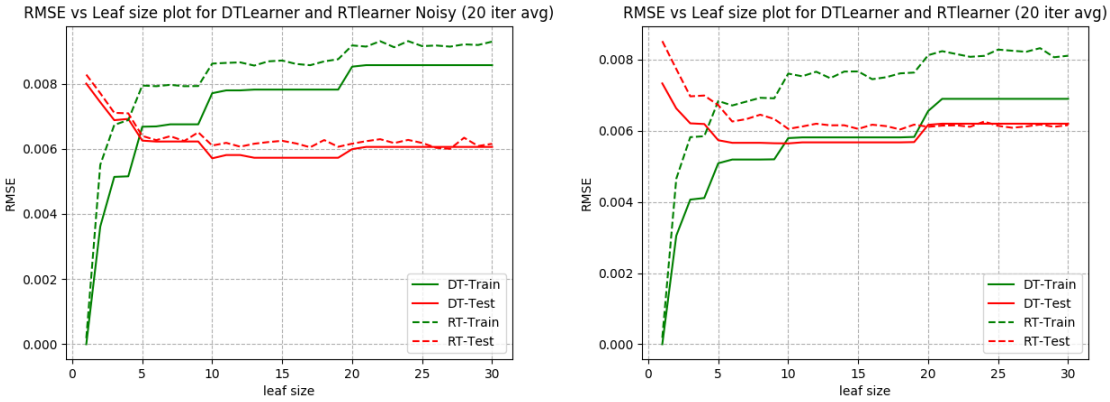

Figure 10: RMSE vs Leaf size for DTLearner and RTLearner on noisy and clean Istanbul.csv data set

Similar behaviour is observed in RMSE values. The increase in RMSE is more for DT than RT. Hence RT is more robust to corruptions or noise.