# Assess Machine Learning Algorithms

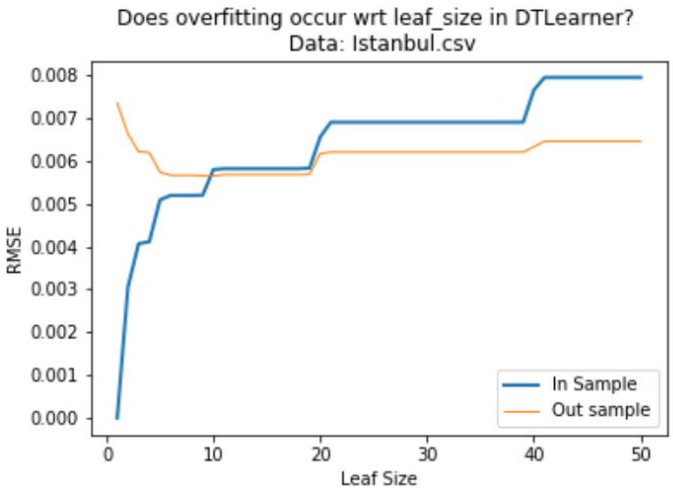
**Experiment 1: Does overfitting occur with respect to leaf size?**

To evaluate overfitting, I decided to run an experiment on the Decision Tree Learner varying the **leaf size values** from **1 through 50** and measuring the RMSE metric for the runs.

Overfitting is the phenomenon when we train our model to overfit data such that it even incorporates the random errors in the data and use it to model and make future predictions. As we can observe, **overfitting does occur** with respect to the leaf size. In the graph 1, for the in-sample data, the RMSE increases as leaf size increases. We can notice that, at **leaf_size 2**, the RMSE takes a steep increase and keeps increasing from there. For the out of sample data, the RMSE decreases significantly at **leaf size 2**, and the curve seems almost stable with small increases.

This might be since the data is grouped and generalized in the tree with increasing leaf sizes, which means that the model overfits even the random errors in the dataset and thereby uses it to make its predictions.

**Graph 1:**




**Experiment 2: Can bagging reduce or eliminate overfitting with respect to leaf size?**

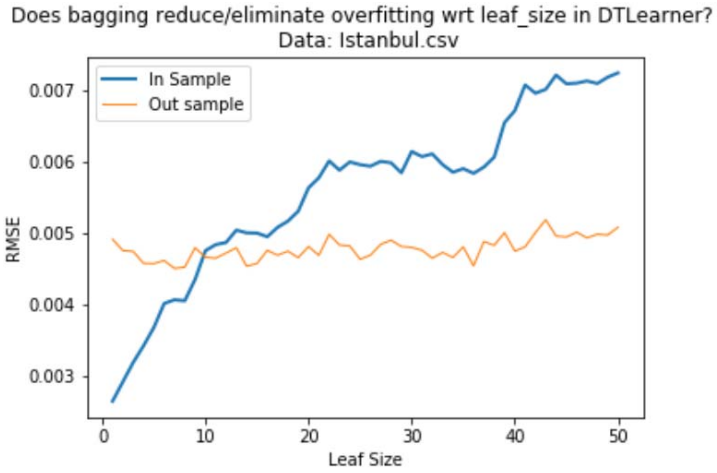
To investigate this question, I chose to use a **fixed number of 20 bags** and running the Bag Learner with Decision Trees varying the **leaf size** from **1 through 50** and measuring the RMSE values for each run.

In the graph 2, we can observe that for in sample data, the RMSE increases significantly from leaf sizes 2 to 7 and then gradually climbs up. This is due to the fact that, as the data is generalized and converged with increasing leaf sizes, the error in prediction also increases.

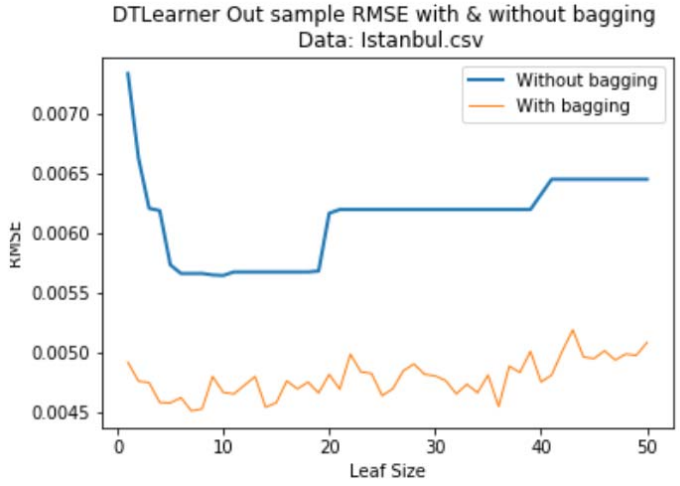
For the out of sample data, the curve is almost stable with small fluctuations. RMSE is lowest at leaf size 7. Thus, we can state that **bagging has not eliminated overfitting** altogether.

I decided to plot just the out of sample RMSE with and without bagging for better inference of the results (since, future predictions seem more appropriate in terms of evaluating overfitting). From graph 3, we can see that the out of sample RMSE curve with bagging is significantly way lower than the one without bagging. Thus, we can state that, **bagging does reduce overfitting significantly**.

**Graph 2:**




**Graph 3:**




**Experiment 3: Quantitative comparison of "classic" Decision trees (DTLearner) versus Random trees (RTLearner). In which ways is one method better than the other?**

For comparing the two models, I decided to use the following performance metrics:

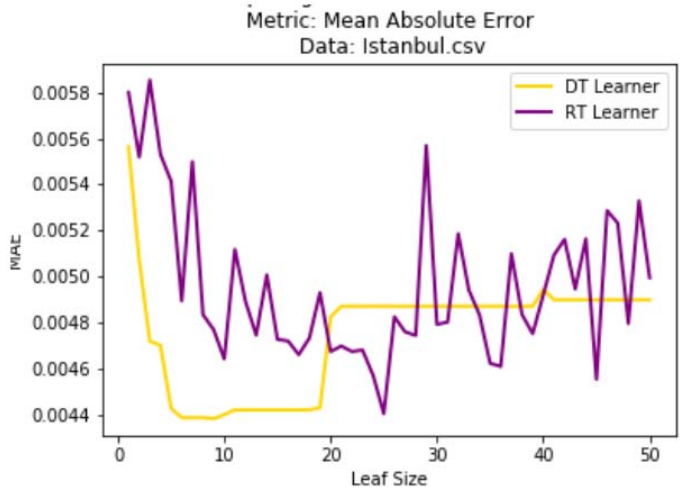
Accuracy – **Mean Absolute Error (MAE)**

Execution Time – **Time taken to build the tree**

For the MAE metric, I decided to run both DTLearner and RTLearner with **out of sample values** and varying only the **leaf sizes** from **1 through 50** and for each run measuring the MAE as the mean of the absolute difference between the actual and predicted values, i.e. **Mean(absolute(actual_output – predicted_output))**

From graph 4, we can see that the MAE curve for DTLearner drops significantly at leaf_size 2, but then around leaf_size 20 increases and almost stabilizes from then on. Whereas, the RTLearner curve, though does drop low sometimes, has varied fluctuations throughout. Thus, we can conclude that the **DTLearner seems more stable and reliable than the RTLearner** according to the **MAE metric**.

**Graph 4:**




For the Time metric, I decided to **train** the DTLearner and RTLearner with data varying only the **leaf sizes** from **1 through 50** and for each run measuring the time taken to build the decision trees by both models.

From the graph 5, the tree building time curve for the DTLearner reduces significantly around leaf sizes 2 to 5 and then keeps reducing and almost stabilizing with small fluctuations. This is due to the fact that the best feature is selected at each node by calculating the correlation coefficients of all the features. At leaf size 1, it takes the maximum time, since the decision tree has the maximum number of nodes there.

For the RTLearner too, there is a noticeable drop in time around leaf sizes 2 to 5 and then the curve follows a stable path with small fluctuations.

But, from the graph its obvious that the tree building time for the **RTLearner is always better than the DTLearner**, since the best feature is always selected randomly.

**Graph 5:**