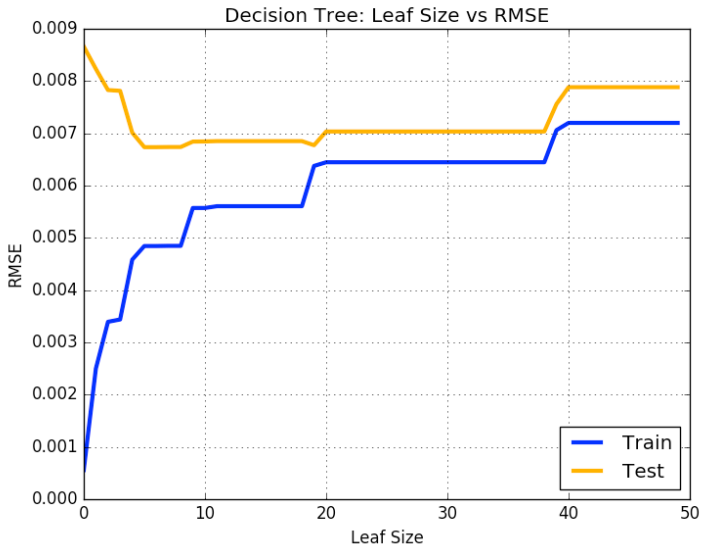
Assessing Learners:
Decision Trees, Random Trees, & Bootstrap Aggregation

Karel Klein Cardeña

Effect of Leaf Size on Overfitting

Upon implementing a deterministic decision tree (DT), we observe `leaf_size` as a hyperparameter with a potentially great impact on the size and shape of the resulting decision tree. We can expect that as leaf size approaches 1, the resulting DT will fit the training data increasingly well, while a leaf size approaching the number of training samples will result in the opposite effect: a failure to fit the nuances of the data at all.

To gather empirical data on this question, we can design an experiment where we build a series of DT learners, each with a different `leaf_size` hyperparameter. We can then use `Istanbul.csv` data to train each model, and measure the resulting root mean squared error (RMSE). Since we are analyzing continuous numeric data, utilizing RMSE as an error metric will be appropriate. We will measure the error for both training and test sets in order to assess the resulting impact on overfitting, and visualize the results by generating a chart of Leaf Size versus RMSE. The chart resulting from this experiment is seen here:



We notice a sharp increase in training error from 0.0 to 0.0055 when leaf size increases from 1 to 10 samples, after which the error continues increasing but at a small constant rate. This meets our prior expectations: the DT will be able to perfectly predict the dependent variable when there is a leaf for each sample of data. In this case, the DT essentially acts as a 'map' leading each sample's features to their corresponding y value. As leaf size increases, error is injected into the model as the DT learner aggregates data samples to each leaf and fails to model the data exactly.

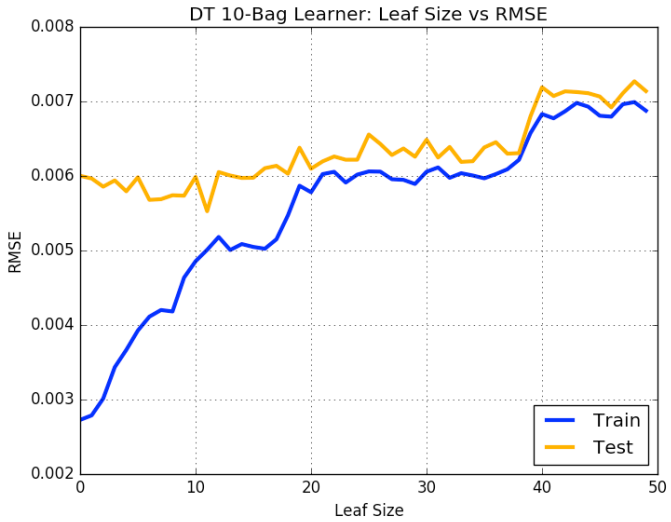
Looking at the same segment of the chart (for `leaf_size = [1,10]`), we can see that when `leaf_size=1`, test error is at its highest point of 0.0086. This reflects the fact that the DT was trained to exactly model the training data, and consequently fail to generalize data it has not yet seen. In essence, the DT is too specific to the data it was built with and cannot be used to model the global pool of data from which that sample was drawn. As soon as leaf size gets larger, the test error immediately responds with a sharp decrease in RMSE, heralding the ability of DT learners better suited to generalize data.

From these observations, we can conclude that overfitting is taking place when leaf size decreases from 5 to 1. In this area of the graph, we see train error drop to 0 as leaf size approaches 1, while test error increases dramatically following the same direction in the x-axis. These are the chief features of a model that is overfitting the data they aim to represent. As leaf size gets larger than 5, overfitting ceases to be a problem and we see both train and test error stabilize. From this experiment, it is therefore clearly evident that leaf size can be largely responsible for overfitting in deterministic decision trees.

Bagging & Overfitting

Bootstrap aggregation utilizes multiple copies of a learner trained with shuffled data and generates a single prediction by aggregating all learners' output. Because we are looking at regression, the aggregation step is simply done by taking the arithmetic mean of the learners' predictions. We are thus interested in inquiring whether this ensemble learning method could reduce or eliminate overfitting with respect to leaf size.

To investigate the question, we conduct an experiment with a Bag Learner (BL) containing 10 decision tree learners, and reproduce the previous chart in order to evaluate the effect of leaf size on our error metric of RMSE. The results are seen below.



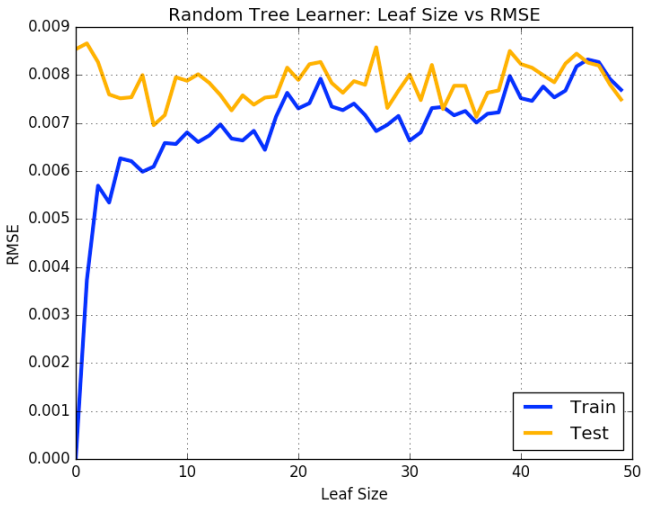
The pattern shown in the chart resembles the results of our first experiment. As leaf size increases from 1 to 11 samples, training RMSE increases rapidly from 0.0028 up to 0.0052, after which it continues to increase but at a subdued pace. In this same region, test error starts at 0.006 when `leaf_size=1` and drops to 0.0055 when `leaf_size=11`. This leaf size range describes the region of overfitting: the characteristic convergence of train and test errors as leaf size

increases exemplifies a learner that began by fitting its training data too well and failing to predict unseen samples, but gradually increased its predictive value as the hyperparameter allowed it to sacrifice training error.

While bagging cannot eliminate the problem of overfitting, it does seems to reduce its overall impact when comparing to the single DT learner. Referring back to the DT experiment, we observed a test error of 0.0086 when leaf_size=1. At the same size, BL saw a test error of 0.006 -- that is a 30% reduction in test error simply by using an ensemble of the same DT learners. Likewise, the respective end of overfitting at leaf_size=5 and 11 for DT and BL saw a decrease of 24% in test error when bagging was used. This supports the claim that an ensemble of learners perform better than the sum of their parts, and can provide a powerful, yet simple, method to reduce the impact of overfitting in decision trees.

Decision Tree vs Random Tree

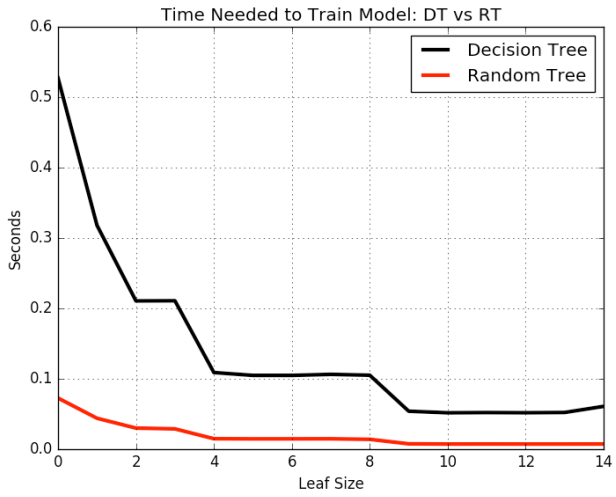
Random trees (RT) have a small yet critical difference from decision trees. Instead of selecting the feature to split on using a relevance metric such as correlation or information gain, it simply selects a feature at random. To test whether how this method compares to the classic DT, we repeat the first experiment and instead use a RT learner; we then vary leaf size and record the resulting errors. Below is the resulting graph.



The graph looks much like that of the DT experiment: a notable period of overfitting occurs while leaf size approaches 8, where train error elevates from 0 to 0.0065 and test error decreases from 0.0085 to 0.007 when `leaf_size=8`. After this initial stage, both train and test error oscillate in a fixed range. Relative to DT, RT has the same test RMSE at `leaf_size=1`, and shows a 3% increase in test error when the period of overfitting ends, respectively. At this point, both learners reach their global minimum RMSE and perform at their best, showing that despite choosing split features randomly, random trees are able to perform almost as well as deterministic decision trees.

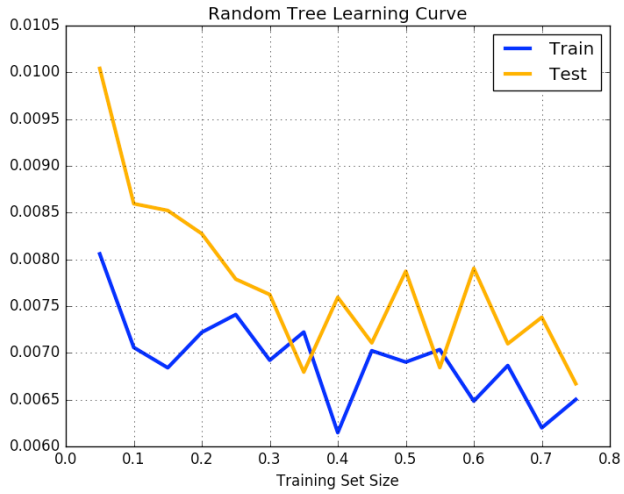
Given that RT select features at random when building its tree, while DT Learner calculates and ranks the correlation for each feature at each step, a key distinction between the two models becomes training time. Logically, we expect RT to have a faster training time, but how much

faster? For this experiment, we instantiate a learner of each kind, and time how long it takes each to build its tree structure(using the `addEvidence( )` method). We then repeat this for different number of leaf sizes.



This graph confirms our hypothesis: regardless of leaf size, RT takes significantly less time to train. In fact, RT is at least 5 times faster than DT at all points in the experiment. While this effect is hardly felt on the 500 instance dataset used for training, most datasets used in machine learning are orders of magnitude larger, which may imply that only a RT learner could perform the task in practical time.

In order to examine the quality of learning by each algorithm with respect to the amount of training data, we can build learning curves that increment the percentage of all data used to train and measure the train and test errors of the resulting models.



In both the DT and RT learning curves, a similar pattern is observed. Training and test error begin high and drop dramatically (perhaps exponentially) as the training set size reaches 40%. After reaching this point, error rates continue to decrease but begin to form a plateau. Looking at the graphs individually, we see DT is able to reach a test RMSE of 0.0065 once it is trained

with 30% of data, and remains stable around this level thereafter. In the case of RT, it lowers its error to 0.0068 and then oscillates around this error mark for higher training set sizes. From this experiment, DT become more favorable when there is little data to train with. They produce more accurate results, and also are likely to be less volatile as the dataset size increases.

This set of experiments displayed the ability of random trees to produce similar results as those of decision trees. DT were slightly more accurate when varying leaf size, as well as when varying training set size. However, the increase in RT testing error of only 3% in both the leaf size and learning curve experiments give much credibility to the method. With the dramatic speed difference observed in training both learners, RT proves the most reliable when time is of essence, while DT lays claim to robustness and stability in results.