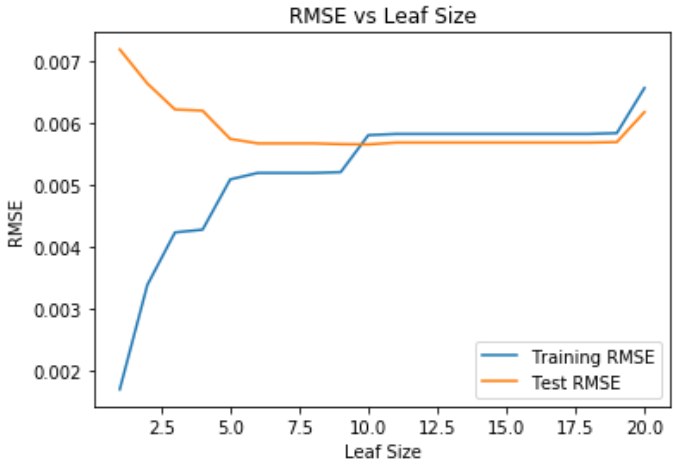
ML4T – Assess Learner

Q1: Does overfitting occur with respect to leaf_size? Consider the dataset istanbul.csv with DTLearner. For which values of leaf_size does overfitting occur?

A1: The Algorithm: DTLearner.py, project the target variable by employing the Decision Tree regression methodology. Decision Tree recursively splits the training data set into subsets based on selected feature that highly correlates to the target variable. And it then generate "leaves", from which we obtained the target variable projection values given the values of a sub data set contained in one leaf. The number of variables in the sub data set is called leaf size. We manipulate the Decision Tree algorithm by changing the leaf size from 1 to 20, evaluating overfitting with respect to the change of leaf size.

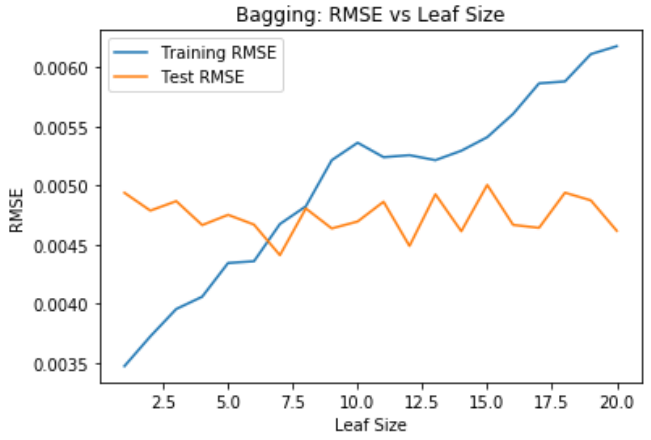
In order to assess overfitting, we split the data set into two groups: training data and test data. Training data is implemented to build up a model, while test data is used to validate the model performance. For each group, we calculate root mean square error (hereafter referred as "RMSE") to quality the model performance. The graph below illustrates the comparison between training data and test data in RMSE with respect to the change of leaf size.

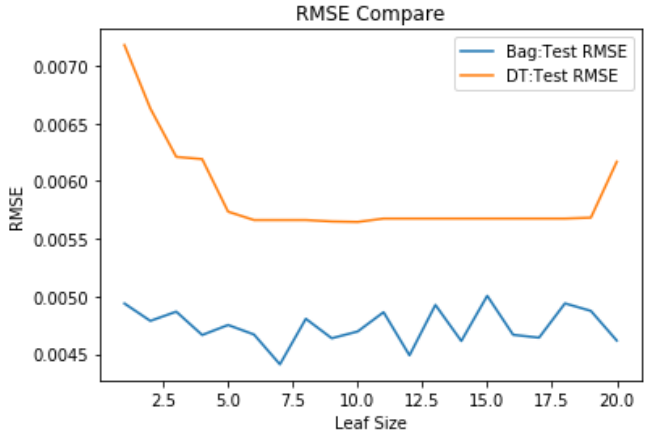


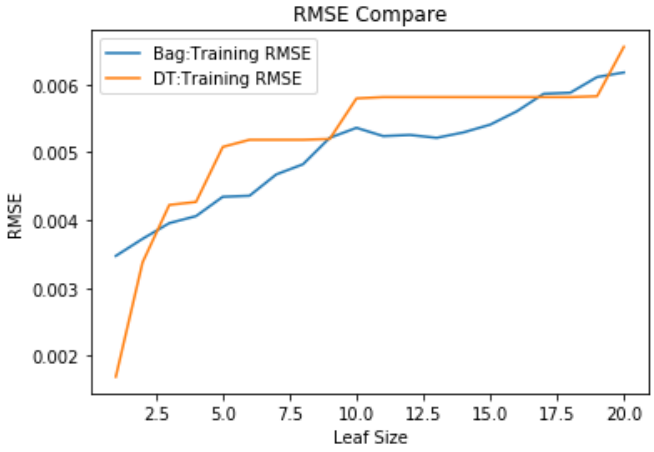
As observed from the graph above, the RMSE varies a lot. The RMSE in test data gradually decreases, while the RMSE in training data climbs rapidly from 0.002 to 0.006 as the leaf size increases. It is typical sign of overfitting. The algorithm focuses on to a particular set of data, fitting closely to it (with relatively minimal error), yet fails or underperforms to fit additional data set, which is not used for algorithm training. From the graph, when leaf size is below 6 (from 1 to 6), the algorithm overfits the train data, as the it perfectly models the training data, yet generates relatively larger error for test data. And the overfitting is remedied as leaf size increases and above 6, as we observed that both of RMSE for training data and test data become more stable.

Q2: Can bagging reduce or eliminate overfitting with respect to leaf_size?

A2: Generally speaking, model variance leads to overfitting. Bagging (also known as Bootstrap) is a variance reduction technique, which repeatedly samples from the training data, and build model based on the sample data. Bagging averages a bunch of learners (number of bags), thus neutralizes the overfitting effect that results from relying on single learner. In order to investigate the overfitting reduction effect of bagging, we employed a bagging algorithm, which includes 10 bags of Decision Tree, with leaf size from 1 t0 20, based on the aforementioned training data and test data.



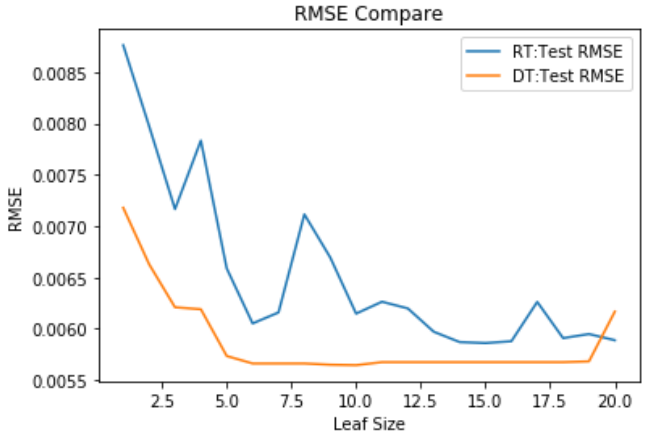


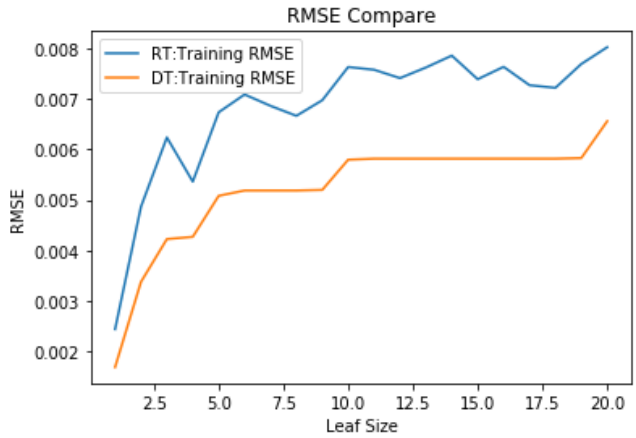


The first graph summarizes the comparison between RMSE of training data and test data for bagging, with respect to the change of leaf size. The second and the third graph illustrates the comparison between bagging and decision tree, in terms of RMSE. For bagging, the RMSE of training data is comparable to the error of one Decision Tree, however, the RMSE of test data notably reduces from 0.006 to 0.005. Additionally, unlike single Decision Tree, the RMSE doesn't vary too much with respect to the change of leaf size. The RMSE of test data with leaf size below 6 is comparable to the other learners with larger leaf size. Therefore, the Given the fact of test data RMSE reduction and stability, we conclude that bagging is able to reduce the overfitting significantly.

Q3: Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other?

A3: In the methodology perspective, the major aspect that differs Random Trees from Classic Decision Tree is that as each recursive step, Random Tree randomly designate feature for splitting the data set, instead of searching for the optimal feature, which achieves the highest correlation to the target variable. To compare these two methodology, we also calculate RMSE of both Random Tree and Decision Tree method trained by same data set to facilitate the comparison. The graphs below summarize the comparison between these two methodology.





Obviously, Classic Decision Tree outperforms the Random Tree in terms of projection accuracy. Classic Decision Tree has less error (i.e. lower RMSE) than Random Tree consistently (except for test data RMSE with leaf size 20).

Speaking of overfitting, the trend of RMSE are very similar between Random Tree and Decision Tree. As demonstrated in the first section, both of these two methods exhibit overfitting.

The model accuracy is compromised and the overfitting issue is not addressed by implementing Random Tree. Hence, we deem that Decision Tree is better than Random Tree method.