Mehmet Oguz Kazkayasi
mkazkayasi3 – 903369796

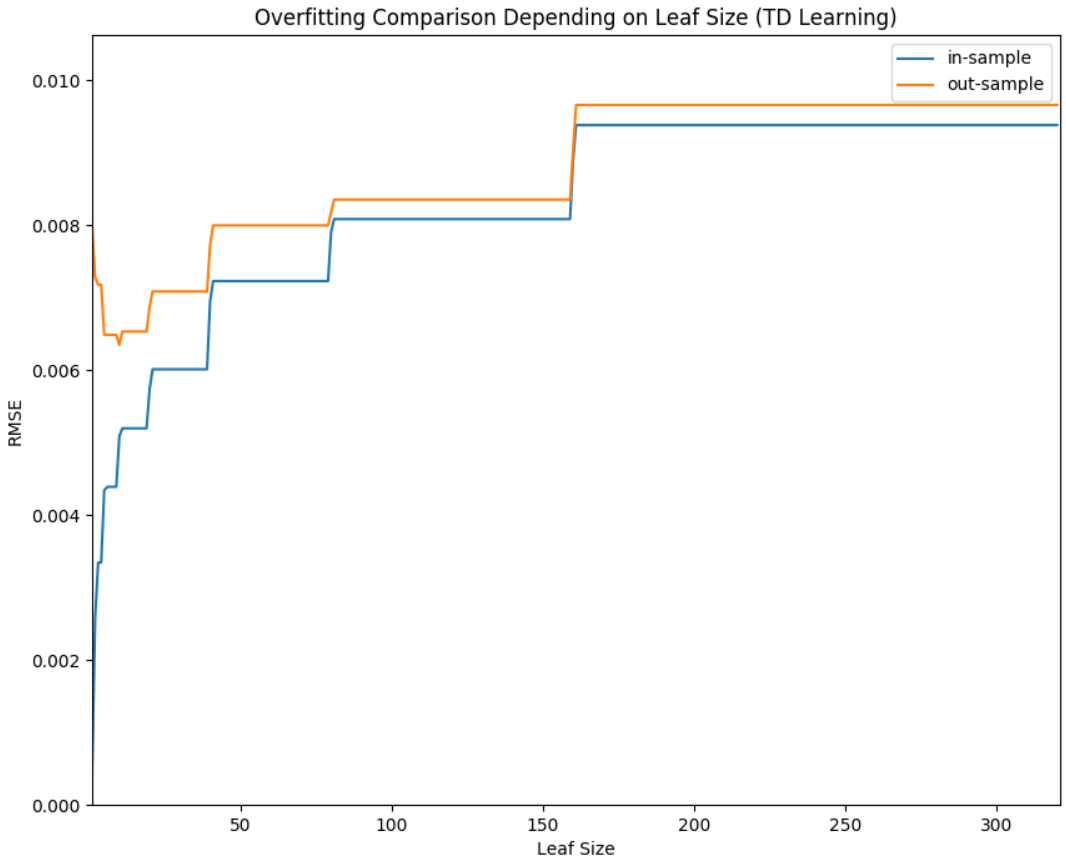
ASSESS LEARNERS REPORT

1. Does overfitting occur with respect to leaf_size? Consider the dataset `istanbul.csv` with DTLearner. For which values of leaf_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging)

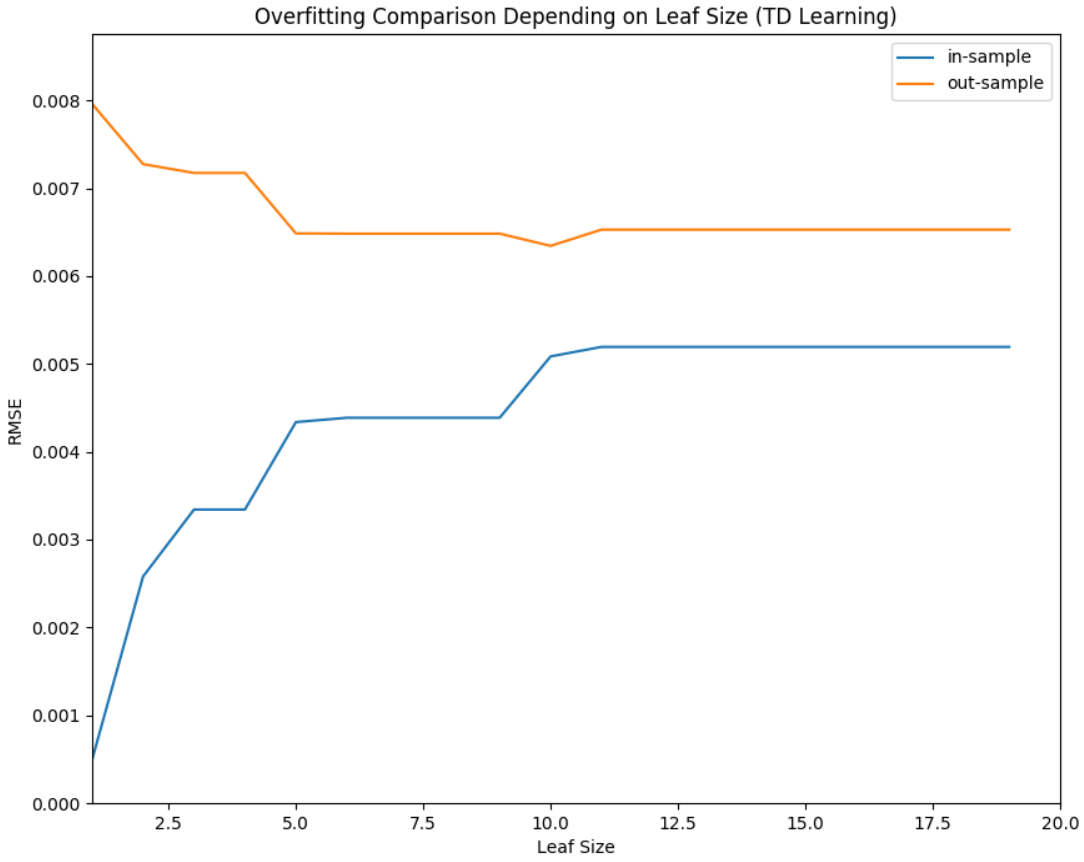
DTLearner algorithm that is implemented in this assignment uses the most correlated feature to create the decision tree. leaf_size parameter decides on whether to keep dividing or just leave the node as a leaf depending on the size of data at that node. The main reason why algorithm doesn't go to the maximum depth by using this parameter is overfitting. Overfitting does occur with respect to leaf_size.

In the training data, there is the signal that can be learned and generalized outside of training data. But, at the same time there is the noise, which cannot be explained by the features of dataset. A learning algorithm must learn the signal and keep the noise out. Whenever we use very little leaf_size parameters, the decision tree algorithm learns the training data very well but it cannot keep its noises out. Because of that, the decision tree created by the algorithm cannot be generalized to the data outside of training set. And the target of the learning is to make good enough predictions outside of training data.

To demonstrate the overfitting, `istanbul.csv` dataset is used and different decision trees are constructed and tested using different leaf_size parameters. The first figure is to show every possible leaf_size and their root mean squared error values when tested against in-sample and out-sample datasets. It can be seen that in-sample testing always gives better results than out-sample. This is because the decision tree is formed using it, so it is biased against the training data.



It is also demonstrated that lower leaf_size parameters construct better decision trees. The out-sample error and in-sample error behaves similar to each other until a certain level. It can be seen that they both diminishes when the leaf_size gets lower. But, after a leaf_size value, in-sample error diminishes greatly but out-sample error starts to increase. What we look for is the minimum RMSE for out-sample data. So, the optimum leaf_size parameter for `istanbul.csv` dataset is 9. If we use lower values than 9, the decision tree starts to learn more noise than signal. If we use higher value, it can't learn as much signal.



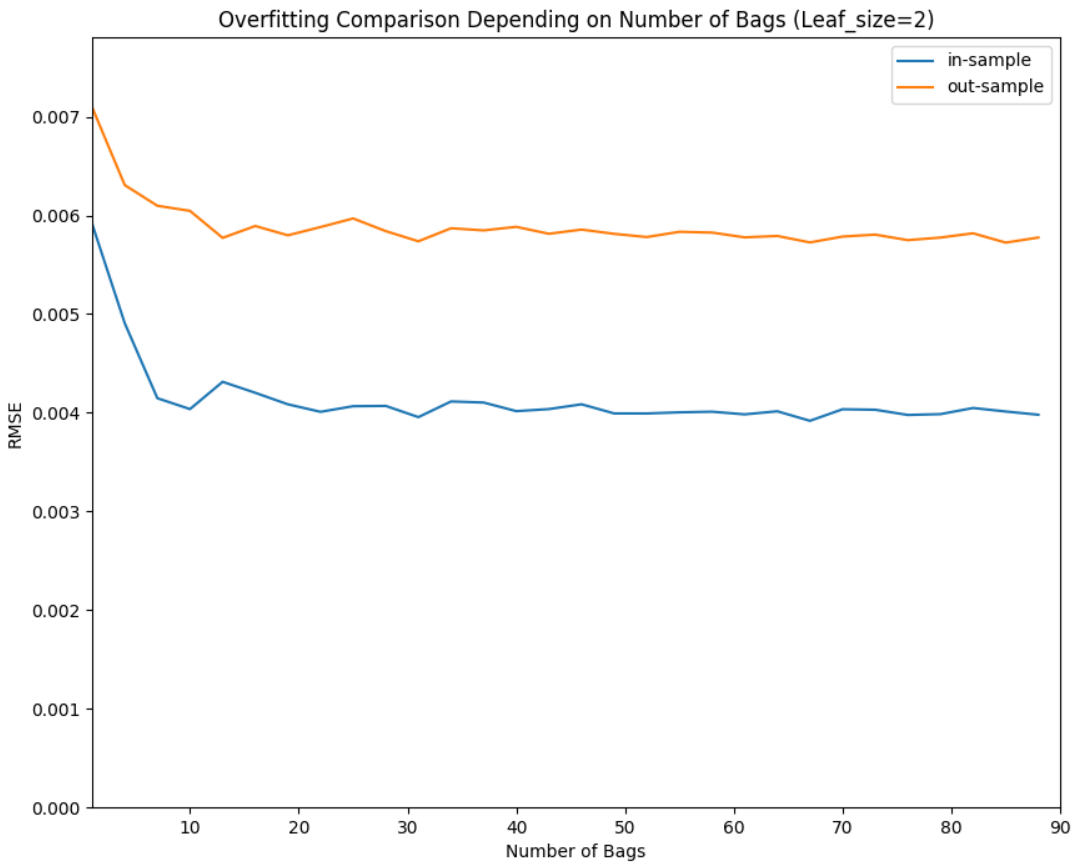
The optimum value for the leaf_size can be seen in better detail at the figure above. If leaf_size is lowered to 8, it is seen that RMSE of in-sample data decreases significantly, but RMSE of out-sample data increases. This level stays almost stable until 5 and leaf_size 4 gives a worse result than leaf_size 20. This shows the overfitting caused by too low leaf_size.

2. Can bagging reduce or eliminate overfitting with respect to leaf_size? Again consider the dataset `istanbul.csv` with DTLearner. To investigate this choose a fixed number of bags to use and vary leaf_size to evaluate. Provide charts to validate your conclusions. Use RMSE as your metric

Bagging uses more than one decision tree (forest) to make the predicts. What makes each decision tree different than each other is the fact that each sample is chosen randomly with replacement. So, in each tree, some data points are more significant than others. By doing this, we get a set of more biased trees than a normal decision tree. These biased trees construct a single predict for a new data point and their biases cancel each other out leaving us a less biased predict.

In this experiment a fixed size of 20 bags is used and the RMSE results of different leaf_size parameters are tested. As seen in the figure below, the leaf_size parameter doesn't have a significant effect on overfitting when used together with bagging. This is because of the structure of bagging, which prevents overfitting at smaller leaf_sizes. It is seen that, the in-sample RMSE value never gets close to 0, and out-sample RMSE value stays on the same level with some random oscilations.

It'd be better to check a leaf_size that leads overfitting as a single tree. RMSE value of a single decision tree with leaf_size 2 is over 0.007, which is quite high comparing to the optimum value. So, in this experiment bagging is used to reduce the overfitting. As a result, it is seen that the best RMSE value given by different sizes of bags with leaf_size 2 is 0.005773, which is even lower than the best bagging result of leaf_size 9. This experiment demonstrates that leaf_size 2, which causes overfit when used in single decision tree for `istanbul.csv` dataset, is a better parameter when used with bagging. Also, when we check the in-sample dataset, it is seen that leaf_size 2 gives a very low RMSE when used in single decision tree but, the RMSE of bagging with leaf_size 2 is not that low, which is a sign of reduced overfitting.

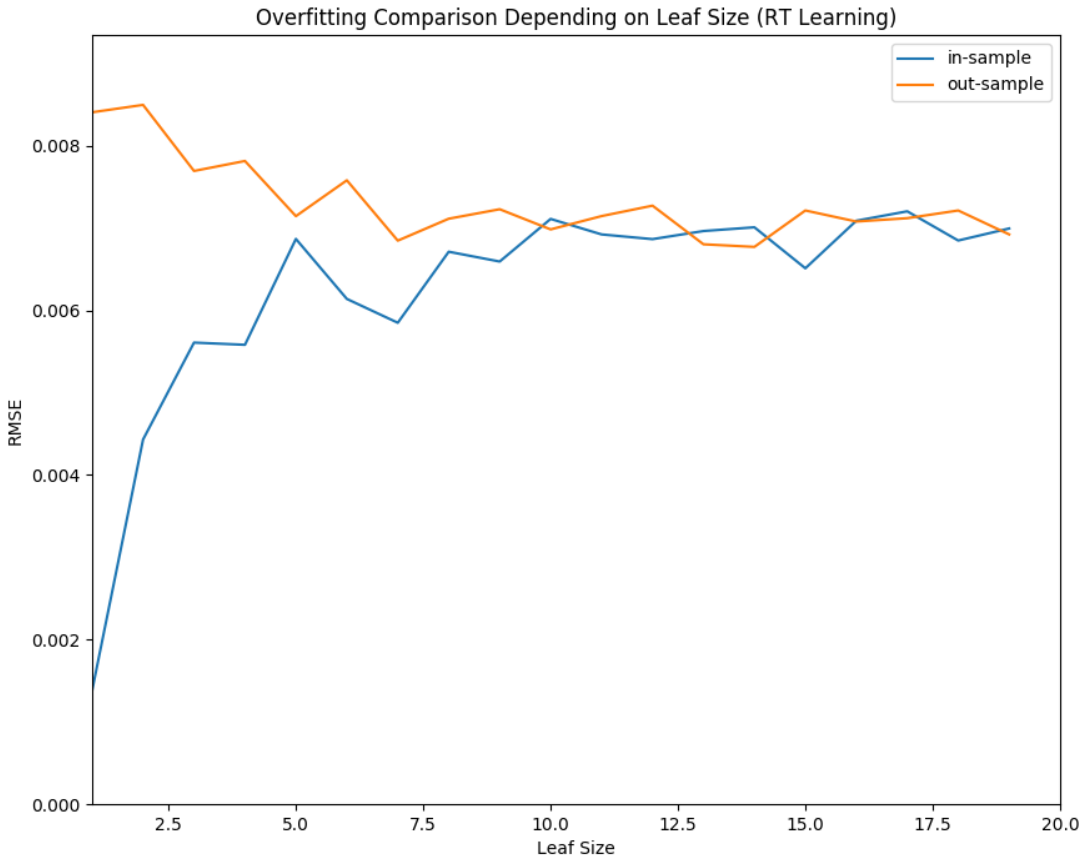


3. Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other? Provide at least two quantitative measures. Note that for this part of the report you must conduct new experiments, don't use the results of the experiments above for this.

a. First of all, classic decision trees use a more complicated algorithm, which tries to choose the best algorithm using different measures such as correlation. On the other hand, random trees just pick a random feature and divides the data using it. Intuitively, classic decision trees are expected to perform better comparing random trees.

First quantitative measure that can be used for comparison can be simply the best RMSE values that both algorithms produce for out-sample data. The best leaf_size for classic decision tree is 9 using `istanbul.csv` dataset. And, it produces an out-sample RMSE of 0.006143 without any bagging or boosting technique used. On the other hand, random trees result in worse RMSE values.

In this experiment, for random trees every size of leaf_tree is run 30 times (because of randomness) and the average of best leaf_size is calculated as 11,30. So, 11 is used as a leaf_size and the out-sample RMSE value of 11 leaf_size is 0.007184, which is higher than what classic decision tree produces.



In the figure above, it can be seen that out-sample RMSE can get lower than in-sample RMSE values in random trees. So, it can be said that random trees generalize better than classic trees when leaf_size is larger than a certain level. But, even though they get lower than in-sample RMSE, they can't get lower than what classic decision trees produce.

b. The second quantitative measure can be the effect of bagging on classic and random decision trees. We can see how much bagging can decrease the RMSE values for each algorithm.

Using the optimum leaf_size 9 for classic decision tree, we can get 0.006143 with a single decision tree. When we use bagging of a size larger than 20 with the same parameter, this value decreases to 0.005794 (mean of bagging 20 to 50), so there is around 0.000619 improvement, which is around 10%.

It is stated that the optimum leaf_size is 11 for a random tree according to empirical results. And this result in a RMSE of 0.007184. If we use a bagging size larger than 20, this value decreases to 0.006873, which is lower than 5%.

This experience shows that bagging has relatively lower improvement on random trees comparing to classic decision trees.