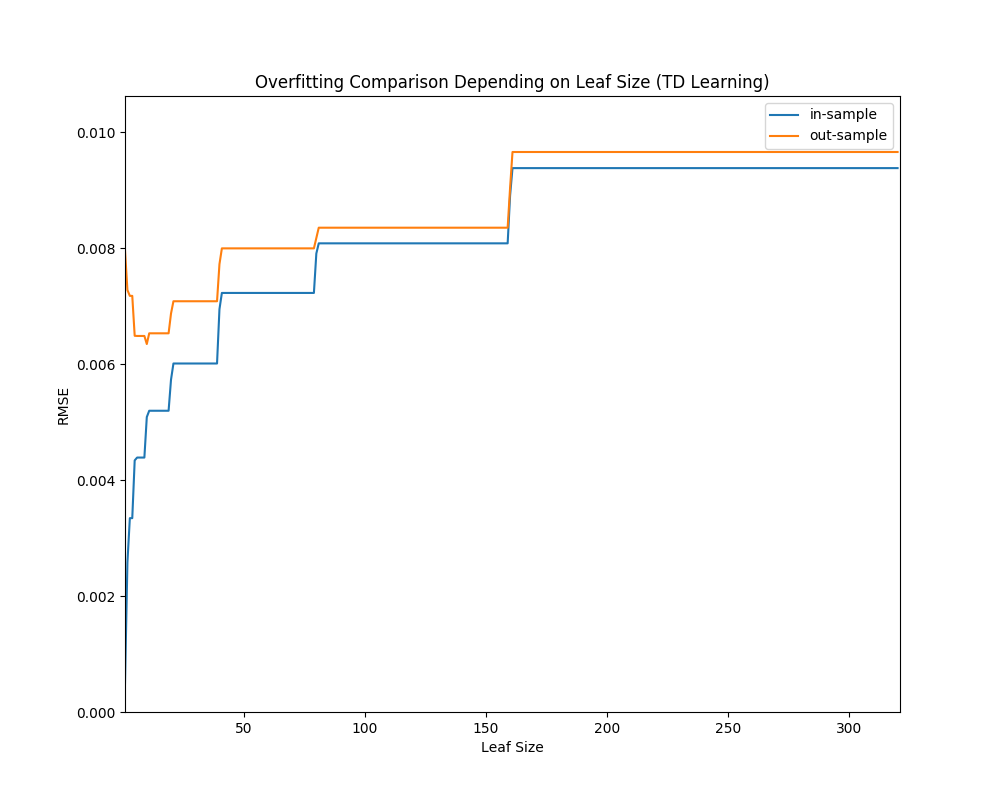
ASSESS LEARNERS REPORT

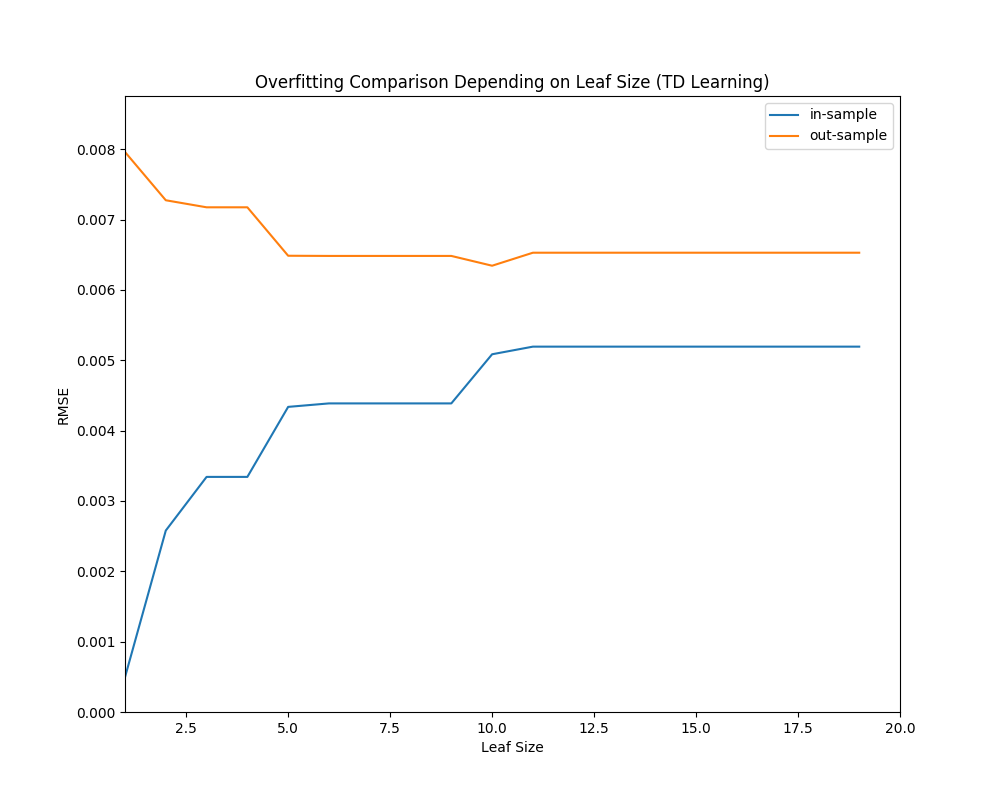
1. Does overfitting occur with respect to leaf\_size? Consider the dataset istanbul.csv with DTLearner. For which values of leaf\_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging)

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.



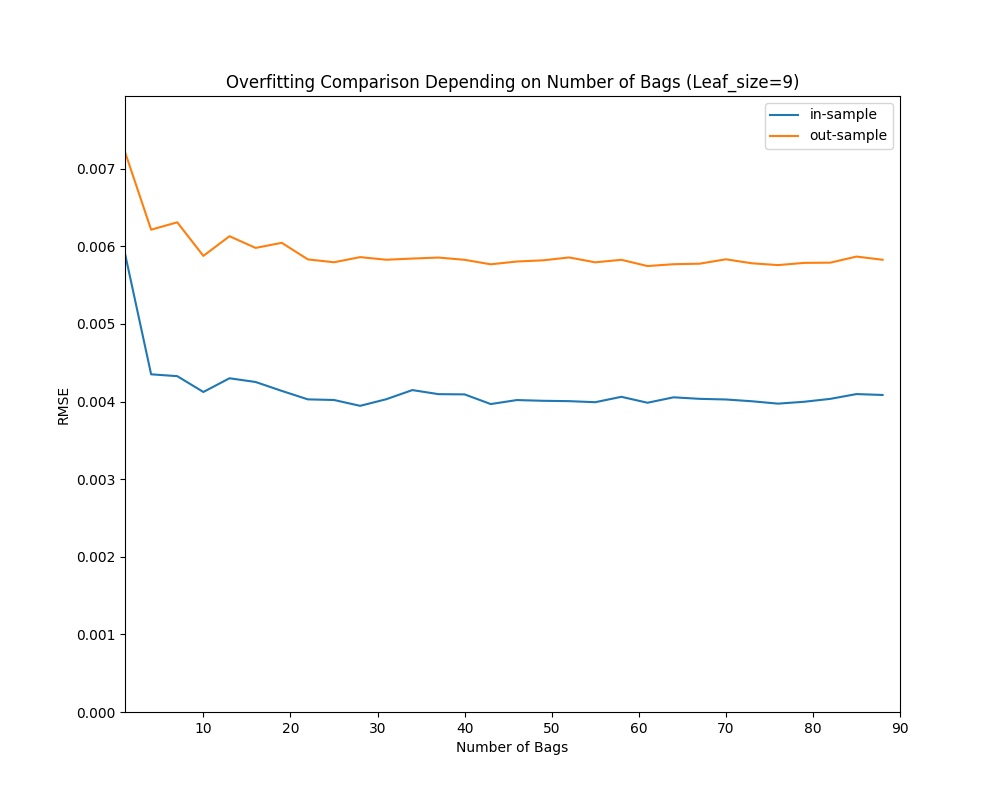
It is also demonstrated that lower leaf\_size parameters construct better decision trees. The out-sample error and in-sample error behaves similar to each other until a certain level. It can be seen that they both diminishes when the leaf\_size gets lower. But, after a leaf\_size value, in-sample error 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. So, bagging can reduce the overfitting but it can reduce it no matter what leaf\_size is. So, we can’t say it eliminates overfitting with respect to leaf\_size.

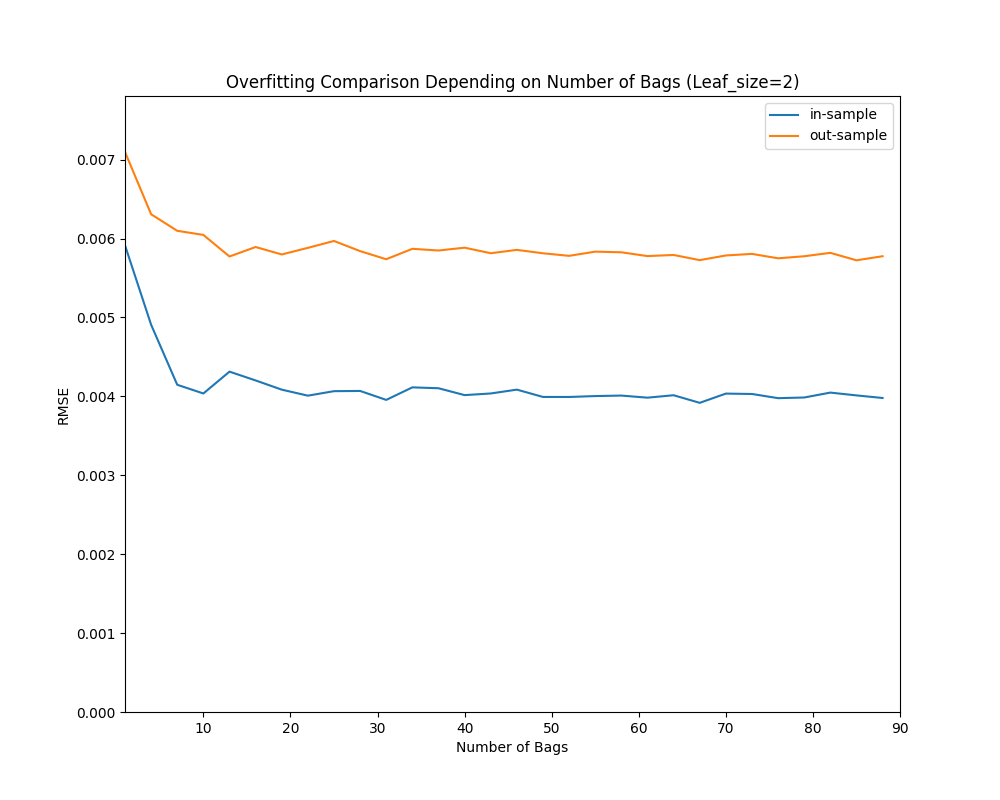
9 was the best leaf\_size parameter for single decision tree. We can check how it behaves under bagging algorithm.



The RMSE value of a single decision tree with leaf\_size 9 is around 0.006143. As we can see, a random forest consists of only 1 decision tree gives a worse performance than that (over 0.007). This is because some of the data points are left out and some data points are used multiple times in the decision tree. But, as number of bags increase, we can see that the RMSE gets much lower than that (0.005794 is the minimum value in this graph).

So, it gives a better result for leaf\_size 9. But, we didn’t encounter a great overfitting with leaf\_size 9. 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.

As seen at both figures, leaf\_size 2 and leaf\_size 9, there is nothing that demonstrates an 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.