REPORT

SUBJECT : MACHINE LEARNING FOR TRADING
NAME : NIDHI MENON
GT username : nmenon34

Assignment 2: assess_learners

Experimental methodology

- The project includes implementation of 3 main learners (DTLearner, RTLearner and BagLearner), besides the InsaneLearner. Another file testlearner.py has the code for plotting graphs. These learners work for the datafile Istanbul.csv.
- The testlearner.py can be run using the command:

python testlearner.py Data/Istanbul.csv

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).

A.

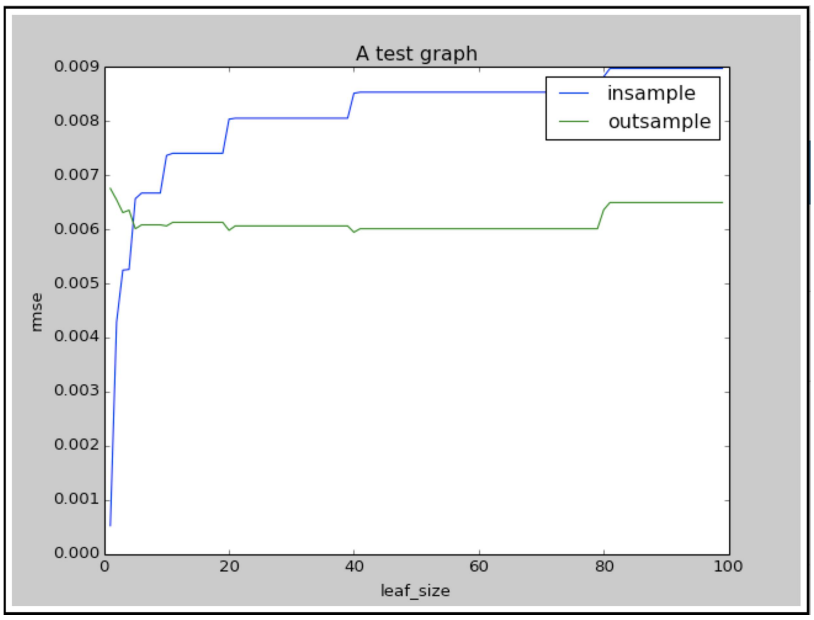

Fig.: RMSE vs leaf-size for DTLearner

Rmse on Y-axis represents root-mean-square-error. Overfitting occurs in this case at the point where insample-error starts decreasing and outsample error starts increasing. This happens for a small leaf-size close to 5, for a rmse value of around 0.006 as interpreted from graph. From this value onwards, overfitting occurs in the left part of the graph.

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 and/or tables to validate your conclusions.

A.

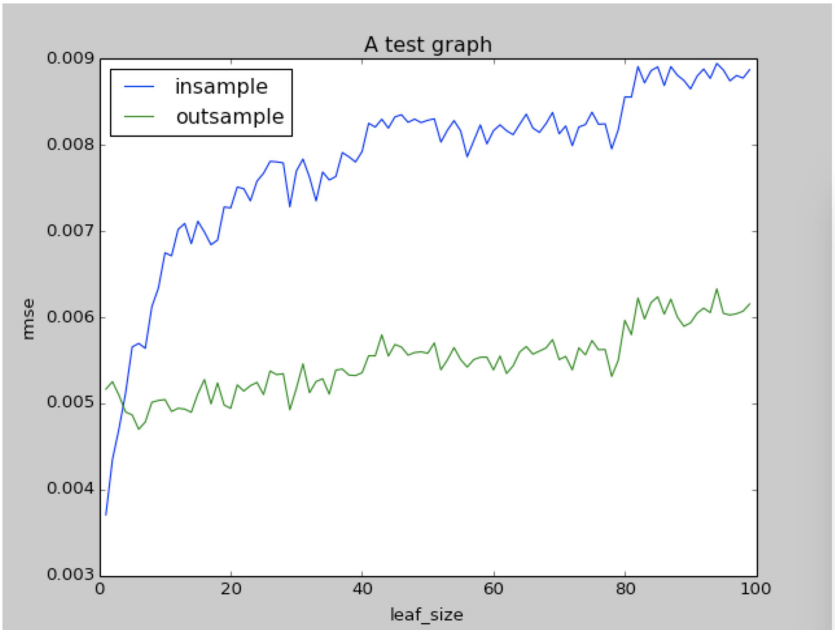

Fig.: RMSE vs leaf-size for BagLearner implementing DTLearner

The above graph is the output from a DTLearner that was implemented using a BagLearner. It plots the insample error and outsample error. In this case, overfitting still occurs in the left-most region of the graph for a lower leaf-size below 5 (as interpreted from the graph), for a rmse of 0.005, which happens to be lesser than what was observed in the previous case. Thus, bagging can reduce overfitting with respect to leaf-size. The number of bags was fixed at 10 for this analysis.

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

A.

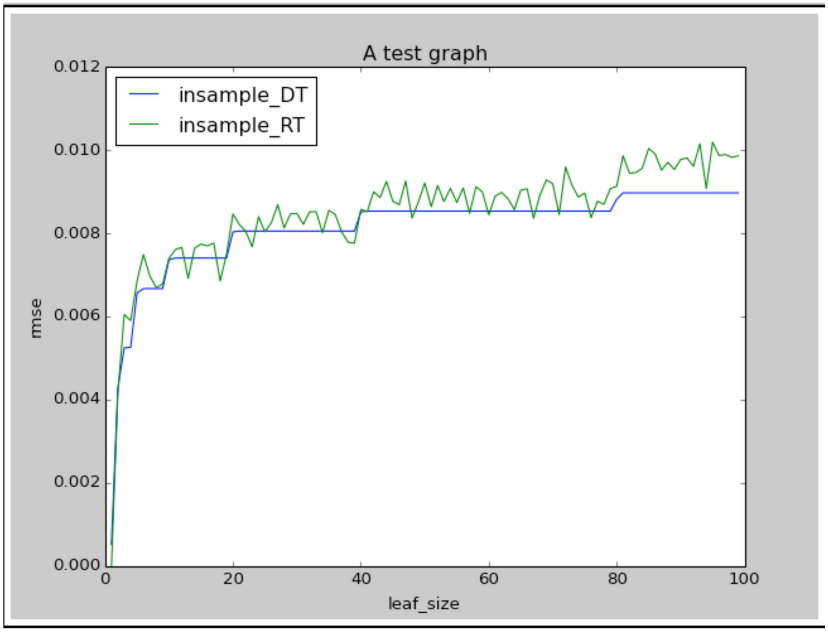

Fig.: RMSE vs leaf-size for insample errors of DT and RT learners

The above graph plots the insample errors for a DTLearner and a RTLearner. As is evident from the graph, the rmse values for RTLearner (max value being close to 0.10) are higher than those of DTLearner (max values being close to 0.09). The error values for RTLearner also have a lot of fluctuations, while DTLearner seems more stable. At certain places in the graph, they even overlap each other. This can probably be attributed to the randomness induced in RTLearners. Since the randomness is limited, the difference is also not too high for DT and RT.

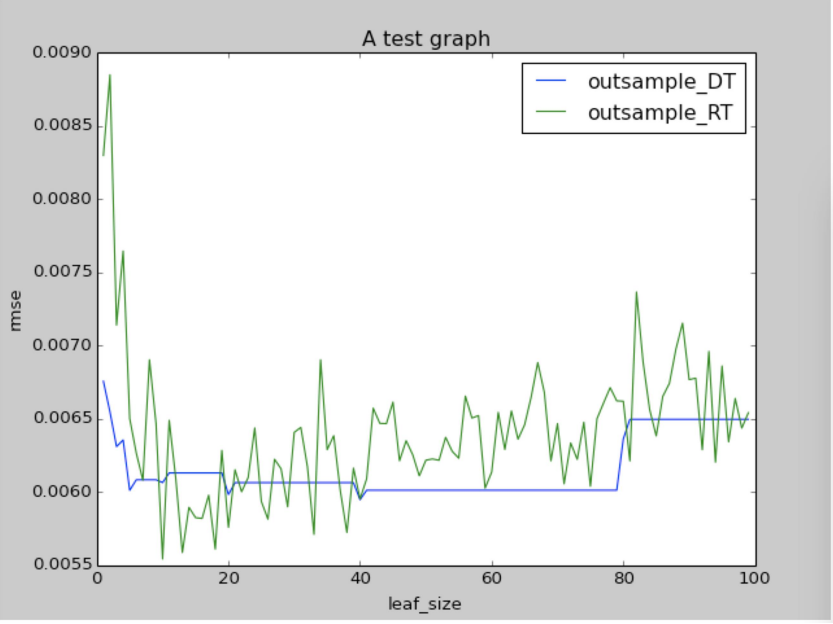

Fig.: RMSE vs leaf-size for outsample errors of DT and RT learners

The above graph plots the outsample errors for a DTLearner and a RTLearner. As is evident from the graph, the outsample rmse values for RTLearner (max value being close to 0.009) are also higher than those of DTLearner (max value being lesser than 0.007). The error values for RTLearner also have a lot of fluctuations, while DTLearner seems more stable. They at times also overlap each other. This can probably be attributed to the randomness induced in RTLearners. Since the randomness is limited, the difference is also not too high for DT and RT.

Thus, on the basis of the comparison of both, insample and outsample errors (2 distinct quantitative properties) for DTLearner and RTLearner, I believe that quantitatively DTLearner seems to be better than RTLearner.