# MC3P1: CS7646 - Machine Learning For Trading

Jeff Shi

February 20, 2017

## 1. Introduction

In this assignment, I implemented and compared the performance of a Random Tree Learner and Boot Strap Aggregating on the Istanbul data, which includes the returns of multiple worldwide indexes for a number of days in history. The overall objective was to predict what the return for the MSCI Emerging Markets (EM) index will be on the basis of the other index returns using these learning algorithms.

## 2. Learning Algorithms

### 2.1. Random Tree Learner

The first learning algorithm is a Random Tree Learner. Random Trees are similar to Decision Trees with several differences. First, the feature to split on at each level is determined randomly. It is not determined using information gain or correlation, or other deterministic method. Second, the split value for each node is determined by randomly selecting two samples of data and taking the mean of their values.

### 2.2. Boot Strap Aggregating

Boot Strap Aggregating, also known as "Bagging", is an ensemble algorithm that is designed to improve the stability and accuracy of machine learning algorithms for regression. Bagging also reduces variance and helps avoid overfitting.

Given a training set D of size *n, b*agging works by generating *m* new training data sets, each of size *n'*, by sampling uniformly from D with replacement. Because sampling with replacement is used, some of the data items in each "bag" will be repeated.

In this assignment, I applied Bagging to the Random Tree Learner.

## 3. Testing and Results

### 3.1. Methodology

First, Random Tree Learner and Boot Strap Aggregating were implemented in Python 2.7 as part of this project. I used LinRegLearner.py as a template to write RTLearner.py (for Random Tree Learner), and BagLearner.py (for bagging). After verifying that my implementation was correct using the Auto-Grader, I created new test scripts to setup the experiments I wanted to answer the following questions:

1) Does overfitting occur with respect to leaf size (for Random Tree Learner)? Which values of leaf size does overfitting occur?
2) Can bagging reduce or eliminate overfitting with respect to leaf size?
3) Does overfitting occur with respect to number of bags? How does RMSE vary as you increase the number of bags?

To answer the first question, I created a new test script called testRTLearner.py in order to run some experiments with RTLearner. Starting from a leaf size of 1 and incrementally increasing the leaf size to 499, for each leaf size, I trained the RTLearner on the training data set, and then ran a query using the testing data set. I then repeated this experiment 3 times and saved the results to a CSV file.

For the second and third questions, I created a new test script called testBagLearner.py in order to run similar experiments with BagLearner. I applied the BagLearner on RTLearner for this experiment. To answer the second question, I first set the number of bags to 20. Starting from a leaf size of 1 and incrementally increasing the leaf size

to 499, for each leaf size, I trained the BagLearner on the training data set, and then ran a query using the testing data set. I then repeated this experiment 3 times and saved the results to a CSV file.

To answer the third question, I first set the leaf size to 20. Starting from 1 bag and incrementally increasing the number of bags to 99, I trained the BagLearner on the training data set, and then ran a query using the testing data set. I only went from 1 to 99 bags due to issues with code execution and runtime, as the training and query process become more computationally exhaustive as the number of bags increased. I then repeated this experiment 3 times and saved the results to a CSV file.

After running the experiment as described, I conducted my analysis of the results using Microsoft Excel. First, I calculated the mean Root Mean Squared Error (RMSE) of the 4 trial runs conducted for each experiment. After making the calculations, I graphed the results for average RMSE for both In Sample test data set and Out of Sample test data sets, and compared them.

All experiments are done using the Instanbul financial data.

### 3.2. Overfitting With Respect to Leaf Size

In order to answer the question on whether or not overfitting depends on Leaf Size, I first graphed the results of RMSE (average of 4 trial runs) vs. Leaf Size for RTLearner. See Figure 1:

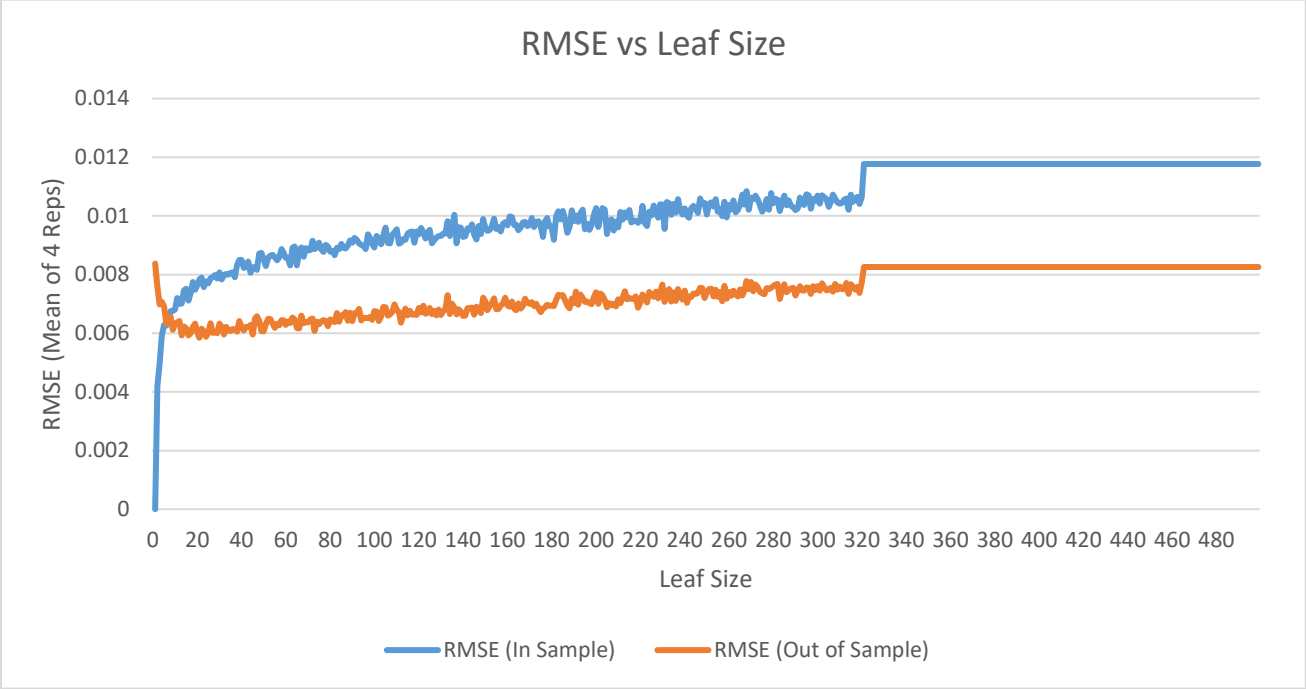


**Figure 1: RMSE vs Leaf Size**

As stated in the lectures, overfitting occurs when the RMSE for the Out of Sample test data set starts to increase after a decreasing trend that follows the RMSE for In Sample test data set. From looking at the graphs, we can deduce that overfitting occurs for **Leaf Sizes that are less than 20**. We see that even though the In-Sample RMSE continues to decrease, the Out-of-Sample RMSE starts to increase.

### 3.3 Overfitting and Bagging (Fixed Bag Number, Variable Leaf Size)

In order to answer the question on whether or not overfitting occurs in bagging for different leaf sizes, I first graphed the results of RMSE (average of 4 trial runs) vs. Leaf Size for BagLearner. As mentioned before, for this experiment, I fixed the number of bags to 20, and tested leaf sizes from 1 to 499. See Figure 2:

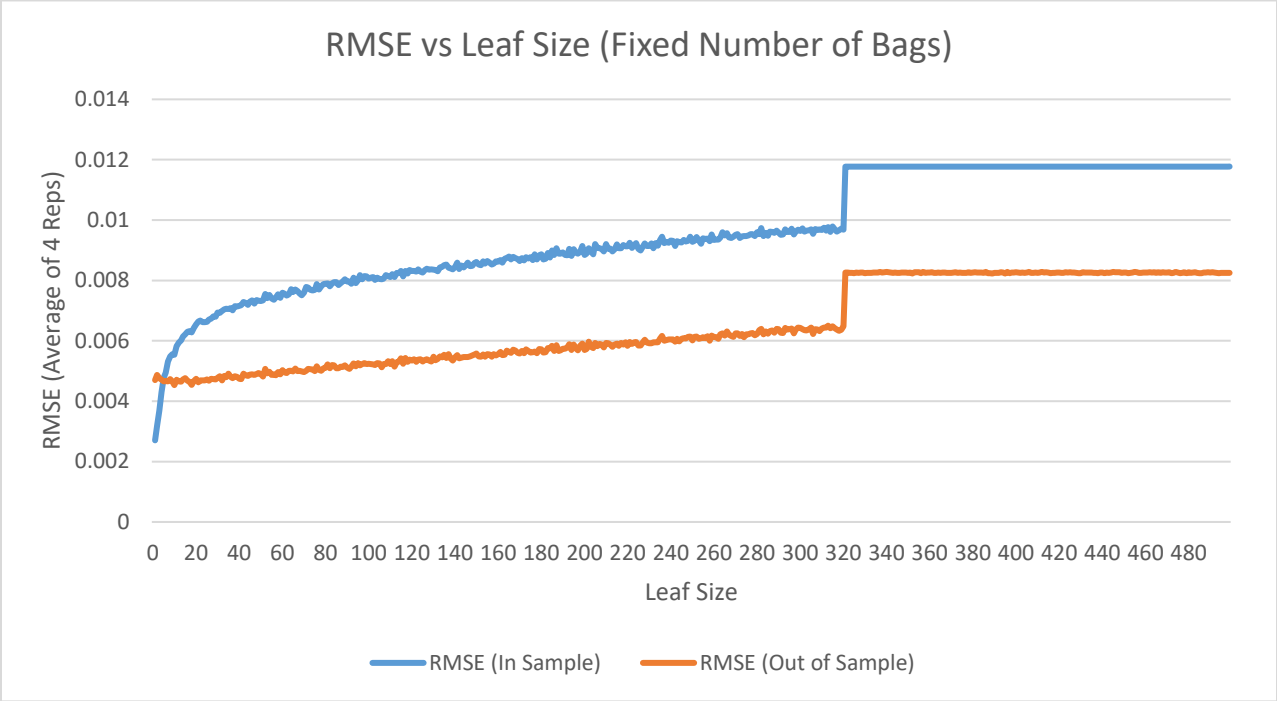


**Figure 2. RMSE vs Leaf Size (Fixed Number of Bags)**

Contrary to the results from the previous experiment, Bagging, when applied to a Random Tree Learner is more resistant to overfitting than just using a Random Tree Learner. We can still observe a slight increase in the RMSE for the Out of Sample test data set as the Leaf Size decreases below 20, but the rate of the RMSE increase is drastically lower than the rate observed in just the Random Tree Learner. However, it does not eliminate it completely.

### 3.4 Overfitting and Bagging (Variable Bag Number, Fixed Leaf Size)

Finally, in order to answer the question on whether or not overfitting occurs in bagging for different numbers of bags, I first graphed the results of RMSE (average of 4 trial runs) vs. Number of Bags for BagLearner. As mentioned before, for this experiment, I fixed the leaf size to 20, and tested number of bags from 1 to 99. See Figure 3:

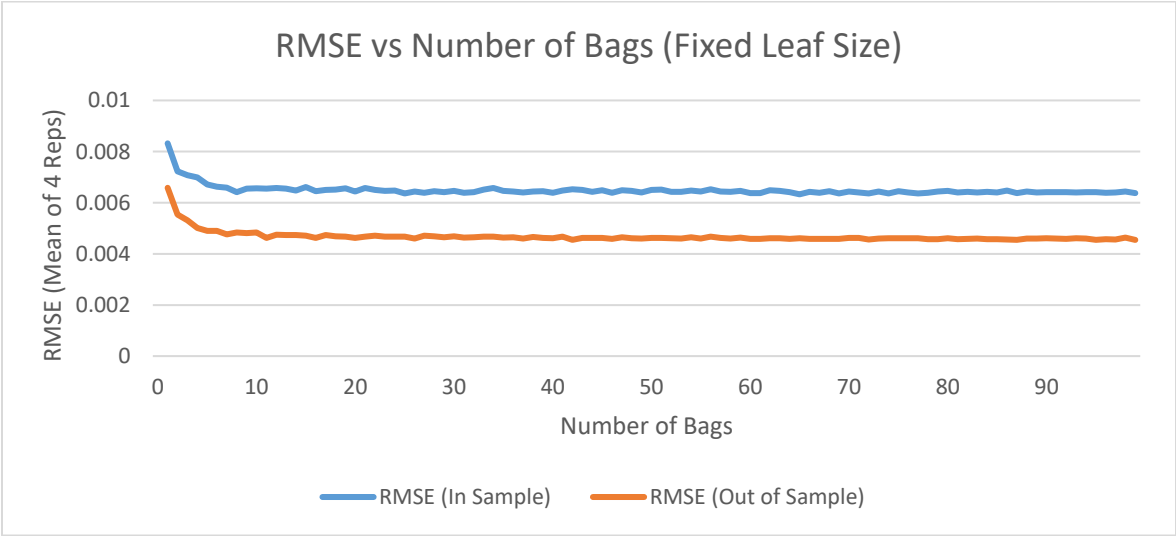


**Figure 2. RMSE vs Number of Bags (Fixed Leaf Size)**

These results are clearly different than the results from the previous two experiments. The RMSE for the Out of Sample test data set continues to follow the pattern of the RMSE for the In Sample test data set, as the number of bags increases. According to the graph, as the number of bags increases, the RMSE decreases until it almost levels out completely past 10 bags. Based on these results, we can conclude that adding more bags, although more computationally expensive, can virtually eliminate overfitting. However, Leaf Size also remains a cause of overfitting.

## 4. Conclusion and Other Thoughts

Based on these three experiments, we can conclude the following:

1) Using a Random Tree Learner without bagging can result in overfitting depending on the Leaf Size. As the Leaf Size decreases below 20, overfitting begins to show in the RMSE for Out of Sample test data set.
2) Applying Bagging on a Random Tree Learner can still lead to overfitting depending on the Leaf Size. Similar to the results in the first experiment, overfitting begins to show as the Leaf Size decreases below 20. The degree of overfitting, however, is much lower than just using a Random Tree Learner alone.
3) Applying Bagging on a Random Tree Learner does not demonstrate overfitting for a variable number of bags.

However, more research and experimentation is necessary for a deeper analysis of the overfitting in Bagging and Random Tree Learner. Provided additional time and resources, I would have used Design of Experiments and Response Surface Methodology to execute more test trials to get a deeper dive on how the number of bags and leaf size affects overfitting. I would have conducted an experiment in which I modify both the number of bags and the leaf size in order to gain more insight.