Project Overview

For this Project. 4 learning algorithms were trained on historical stock market data to see how they perform in predicting the daily stock market returns of the Emerging Markets Exchange given the daily return values data of 8 other indices.

The 3 algorithms used were Linear Regression, Decision Tree, Random Tree, and then each of these algorithms was again ran using "Bagging". The objective of the project was to see:

a) Which algorithms performed better and what parameters for the algorithms yielded the best performance on out of sample test data?

b) Did overfitting occur with certain parameters like different leaf sizes?

c) How (and if) did bagging reduce overfitting and out of sample error for each of the algorithms? What about bagging for different parameters like leaf sizes for the same algorithm?

Overall Results for All Models

In the data 60% of the dataset (istanbul.csv) was randomly assigned to be training data, while the remaining 40% was considered test data. The field 'EM' was the predicted variable. The 8 variables before that were the predictor variables used to predict EM. 'Date' field was discarded for this analysis. RMSE was then used as the error metric to see how well each of the base models performed on test data for each of the algorithms. (The leaf size for both decision and random trees are initially selected here to be 1).

Model	**In Sample RMSE (Train)**	**Out of Sample RMSE**
Linear Regression	0.0046	0.0051
Decision Tree	0.0011	0.0077
Random Tree	0.0029	0.0088

From the above, it can be seen that Linear Regression performs best on Out of Sample data. All models perform better on training in sample data, as expected, and worse on test data.

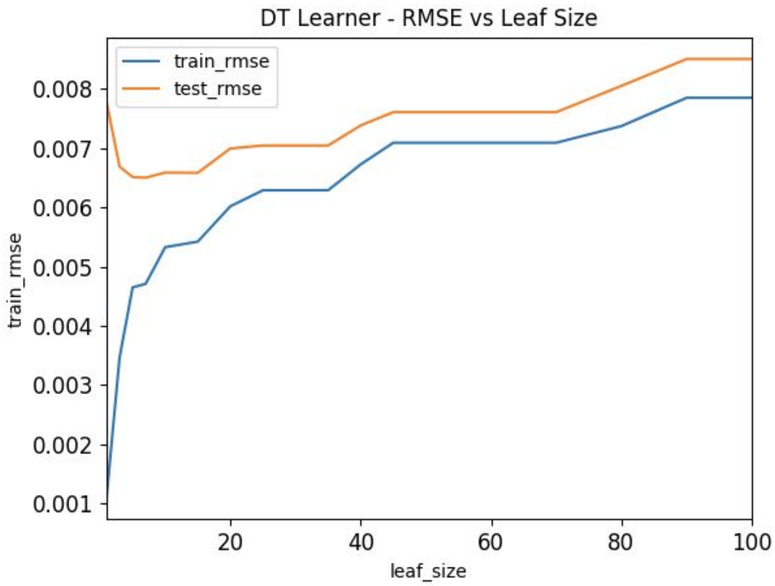
Effect of Changing Leaf Size on Model Performance for Tree Models

However, since leaf size is only 1 both Decision Tree and Random Tree Learner can see their performance change as the leaf_size parameter changes. For this reason both Decision Tree and Random Tree Learner were run again with varying leaf sizes. The results are summarized in the two charts on the next page.

We can see that Decision Tree Models perform on Out of Sample data better when the leaf size is slightly bigger than 1 ideally around the 5-10 leaf size range as the test rmse declines significantly even though the train rmse continues to decrease sharply. This suggests that for leaf sizes < 5 there is overfitting occurring in Decision Trees (especially so when leaf_size = 1).

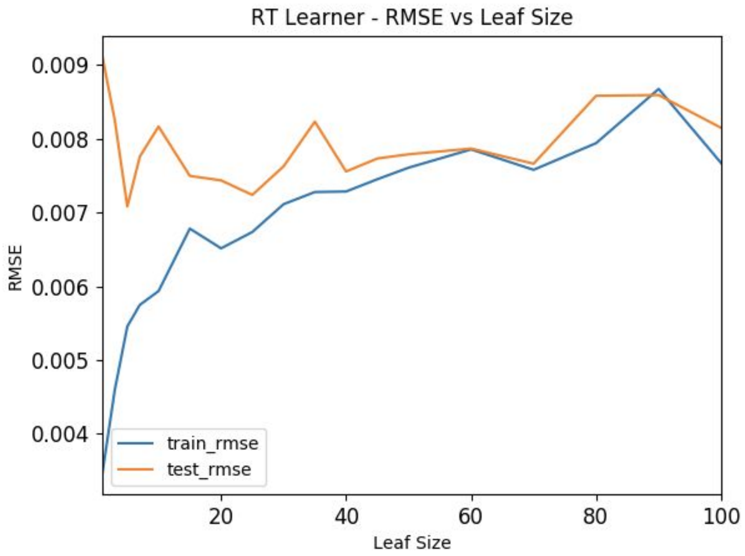
Between a leaf size of 5-15 there is minimal gain in test RMSE. However, after the leaf size of 15 the test RMSE starts to increase again along with the train RMSE. This tells us that after a leaf size of 15 underfitting has started to occur in the decision tree model. So a leaf size around 5-10 can be ideal for performance purposes as it leads to neither under overfitting or underfitting.

Figure 1 - Decision Tree (Test and Train) Performance with Varying Leaf Size



For Random Trees, however, the impact of changing leaf sizes is less obvious. There is a drop in test RMSE after increasing leaf size > 1 which suggests there is overfitting in random trees when leaf size = 1, but it is less obvious if test RMSE improves for leaf sizes > 1 and there is volatility in the trend. Leaf sizes > 1 tend to have RMSE fluctuate around 0.075 until leaf size 70 when they increase again.

Figure 2 - Random Tree (Test and Train) Performance with Varying Leaf Size



Impact of Bagging on overfitting and with increasing leaf sizes

We now try bagging for Decision and Random Trees to see if bagging improves Decision/Random Tree Model performance.The results are plotted in the chart below. (The number of bags used for bagging here was 15).

First to see if bagging improves model performance, we compare the orange test RMSE line in Figure 3 with the test RMSE orange line in Figure 1, we see that DT Models with Bagging are consistently lower in test RMSE than simple DT Models without bagging. Therefore bagging generally helps to improve model performance for decision trees by reducing test RMSE but only up to 10 bags. After 20 bags, we see very little improvement in RMSE in both train and test Data. (This is also summarized in Table 1 on the next page)

Figure 3 - DT Model RMSE with Bagging vs Number of Bags (xlabel = Bag Size) Used

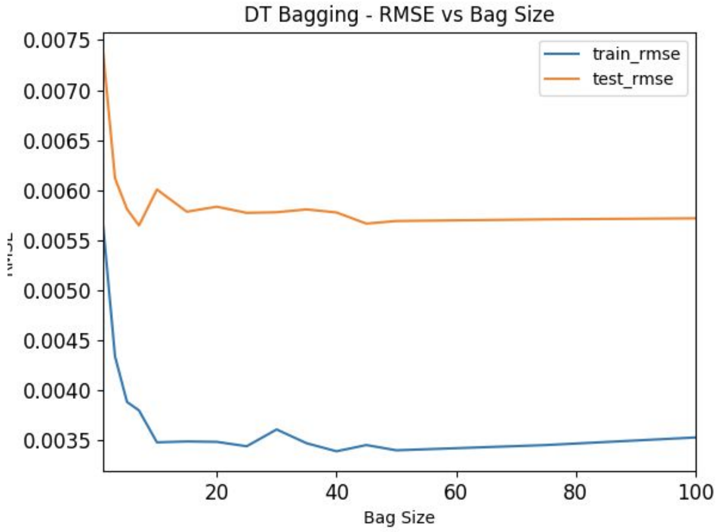
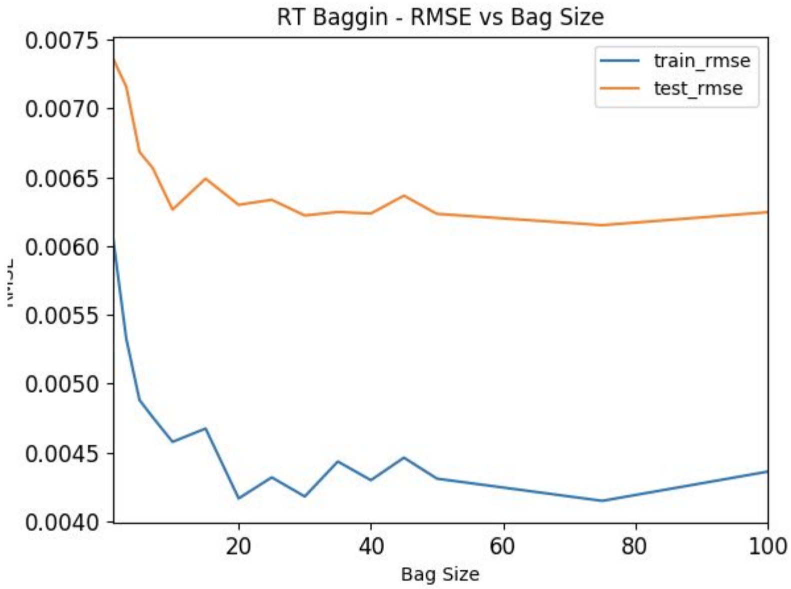


Table 1 - Test RMSE difference between bagged decision tree and regular tree vs leaf size

Bag Size (Number of Bags Used for training)	Bagged Decision Tree Test RMSE	Regular Decision Tree Test RMSE	Gain in RMSE b/w bagged and non bagged DT
1	0.00741	0.00777	**0.00036**
3	0.00642	0.00666	**0.00024**
5	0.00607	0.00651	**0.00044**
7	0.00565	0.00650	**0.00085**
10	0.00610	0.00658	**0.00048**
15	0.00589	0.00658	**0.00069**
20	0.00585	0.00699	**0.00114**
25	0.00571	0.00704	**0.00133**

We see a similar trend with RT trees. Bagging improves model performance of RT trees, but only up to 10 bags. After 10 bags are used the bagged RT model doesn’t improve in test or train performance signficantly.

Figure 4 - Random Tree Model Performance with Bagging vs Number of Bags Used



Does Bagging Reduce or Eliminate Overfitting with respect to leaf size?

Next we also see if changing the leaf size leads to better model performance for bagged decision trees and thereby reduce overfitting. From Figure 5 and 6 we can see that changing/increasing leaf size on DT models with bagging (where number of bags = 15) does not improve model performance significantly.

Interestingly, having a very small leaf size = 1 also does not lead to overfitting or worsening of test RMSE performance. In the non bagged version, very small leaf sizes < 5 were causing overfitting but this is not the case when we do bagging on trees anymore. (Note: around the 20 leaf size position we start seeing underfitting where the test RMSE starts increasing with increasing leaf sizes along with the train rmse which is also what we saw with regular decision trees).

This tells us that when bagging, leaf size doesn't matter as much for reducing overfitting as it does in regular decision trees. In theory, bagging by itself reduces much of the overfitting and tweaking leaf size additionally doesn't provide as much value after that.

We also repeat the above for Random Trees and see the same trend (Figure 6)

Figure 5 - DT with Bagging RMSE vs Leaf Size

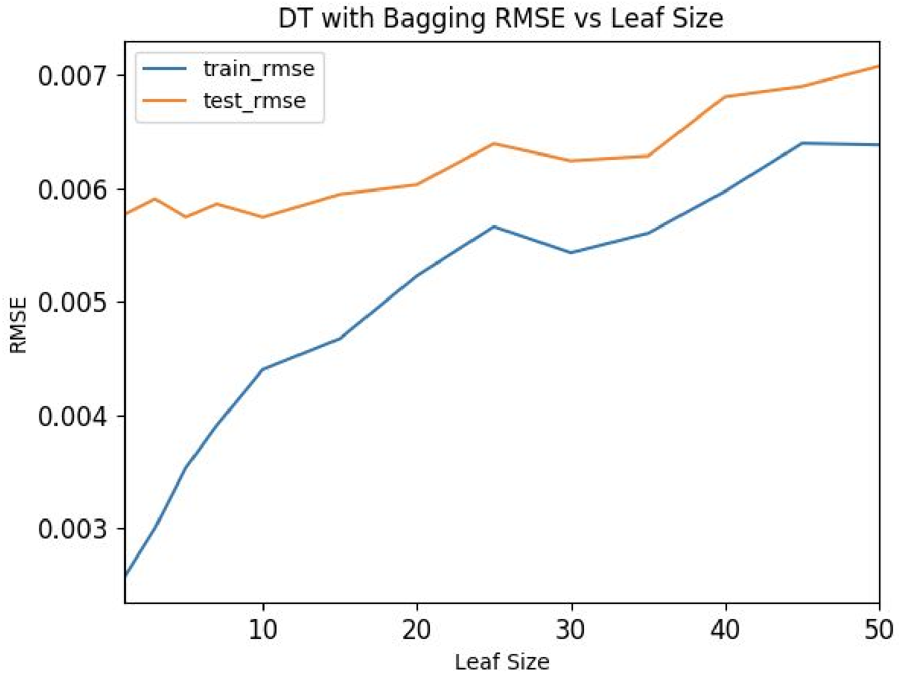
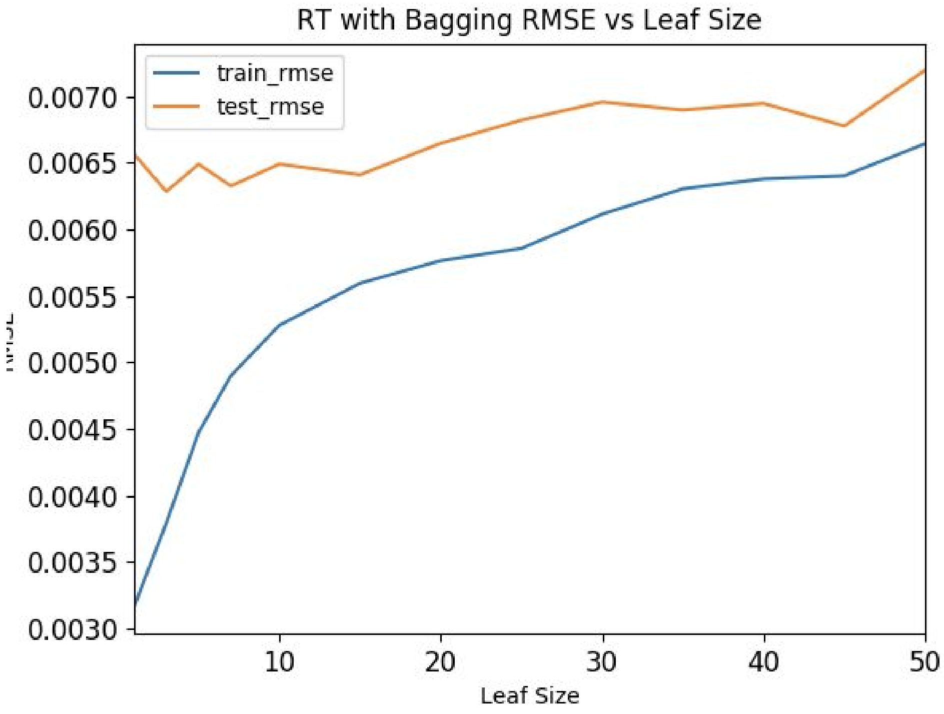


Figure 6 - RT with Bagging RMSE vs Leaf Size



Comparison of Decision Tree vs Random Tree Learners

In the above experiments, we looked at both Decision and Random Tree learners. From the experiment results and charts above it can be gathered that Decision Tree Learners tend to perform better than Random Trees when only a single tree is constructed at optimal leaf sizes (let's say leaf size =5). In such a case Decision Tree Learners have a test RMSE (test) of 0.065 while the Random Tree has an RMSE(test) of 0.072.

Decision trees also perform better on out of sample data once bagging is introduced. Both models see a sharp improvement in performance and overfitting but Decision Trees are still slightly better.

Decision trees also have less volatility/variance in RMSE when experiments are repeated. See below for another chart Figure 7 of RMSE vs Leaf Size for Random Tree repeated with a different seed and compare it with Figure 2. You will see that the charts look different although the trend is similar. Now Look at Figure 8 which has Decision repeated with a different seed. It looks very similar to Figure 1 so there is very little variance in results when Decision Tree is repeated. Hence, DT is more reproducible and consistent as an algorithm while Random trees vary in results when repeated.

Figure 7 - RMSE vs Leaf Size Repeated for a different seed using RT Learner

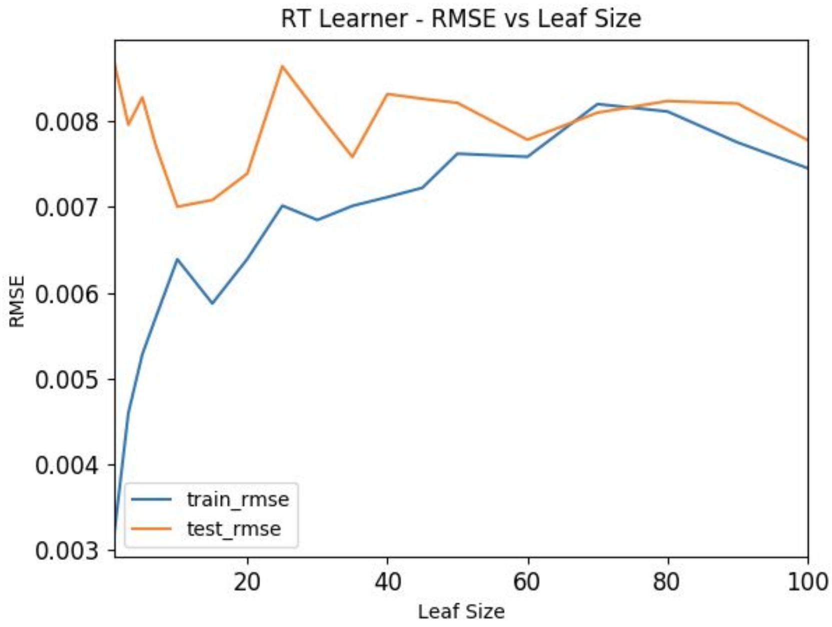
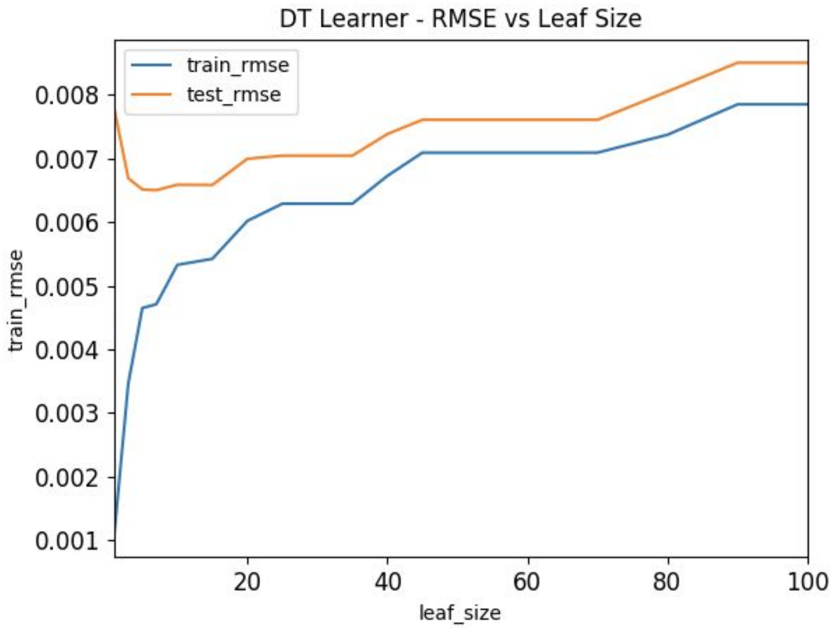


Figure 8 - RMSE vs Leaf Size Repeated for a different seed using DT Learner



However, decision trees take longer to train and predict. On my machine it takes an average of 2.9 seconds for the entire decision tree experiments to run (averaged over 10 runs). Whereas Random Tree experiments only take 1.6 seconds time. This is because Decision trees have increased computational complexity coming from the choosing the best feature at each recursive call. If there are N variables then we have to calculate correlations N times in the worst case. Calculating the median also takes a lot of time (at least Ologn if the array has to be sorted first). However calculating the random value is O(1) so the complexity is reduced for random trees.

Conclusion

For this data, Linear regression models perform best (measured by RMSE) compared to random and decision trees on out of sample test data. However,we can improve decision and random tree performance by choosing an optimal leaf size that can reduce overfitting and improve model performance.We can also greatly reduce test error for both tree methods by using bagging. Decision tree however tends to perform better than random trees even with bagging. Using both bagging and optimal leaf size at the same time however doesn’t reduce overfitting any more as opposed to only using bagging for decision trees.