Project 3 Assess Learners:
CS7646

Abel Aguilar
aaguilar61@gatech.edu

***Abstract*—In this Project we looked at different types of learners and how effective they are predicting stock data. We look at how a Decision Tree learner can be overfit and how we can attempt to mimize the overfitting by changing leaf sizes, using a Random Tree approach, or bagging Decision Tree learners in a Baglearner. We explore this with the different Experiments shown below.**

1 INTRODUCTION

Decision Trees can be a very effective method of predicting data, however they have a tendency to overfit to specific data sets if the designer is not careful. If a tree becomes overfit it can work very well on the data set that it was trained on but its performance takes a hit when it is tested on new data. In the following experiments we look at when a Decision Tree becomes overfit and what we can do to avoid this when possible. We look at different variables that can be changed such as changing leaf sizes to implementing Bootstrap Aggregation to see if this can avoid some overfitting. We then compare Decision Trees to Random Trees (picking a random factor to split on) to see if they have the same issues. Then we compare the trees for the pros and cons of each.

2 METHODS

Three Experiments are done to attempt to solve the issue of overfitting. All three experiments use Data/Istanbul.csv to train and test our learners. In the first experiment we look at the effect leaf size has on Decision Trees, specifically what is the optimal leaf size for this training data. We run through 1-50 leaf sizes and see where overfitting occurs. In this experiment we use RMSE (Root Mean Square Error) and Correlation as metrics to judge the most effective leaf size. In experiment two we look at the effect Bootstrap Aggregation and leaf size has on overfitting. In this experiment we went through 1-50 leaf sizes but made each learner a bag learners with 20 bags (sampled with replacement) of Decision Trees each. Here we also use RMSE and Correlation to judge our results. And lastly in experiment three we compare the pros and cons of Decision Trees vs Random

Trees using Mean Absolute Error and time to train/time to query to see which learner is better. We compare each learner at 1-50 leaf sizes to determine which is the better overall learner.

3 DISCUSSION

Below are the experimental results:

3.1 Experiment 1

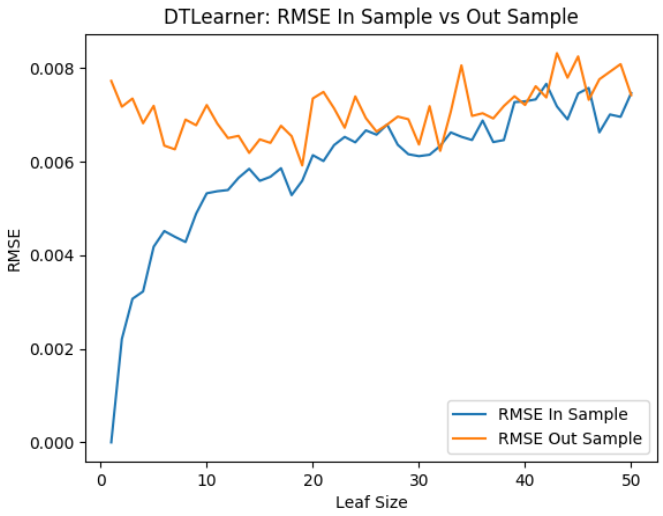

Figure 1—Decision Tree Learner RMSE for In Sample and Out of Sample Data. Leaf Size of 1 - 50

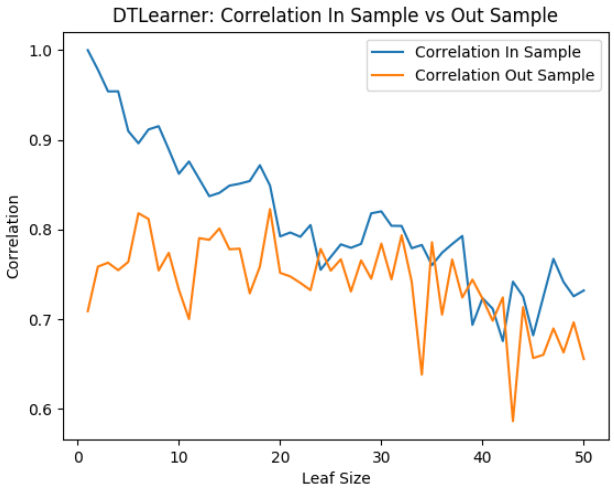

Figure 2—Decision Tree Learner Correlation for In Sample and Out of Sample Data. Leaf Size of 1 - 50

For Experiment One we can see that overfitting does occur with respect to leafsize. In both the RMSE graph and Correlation graph we can see that at leaf size of 50 we have a high RMSE and low correlation. As we begin to decrease leaf size we begin to see improvement in both RMSE and correlation. This trend continues until leaf size 15. From leaf size 15 to 1 we begin to see our graphs separate and see that in sample RMSE goes down drastically while out of sample RMSE begins to increase again. We see a similar trend in correlation. In sample correlation goes up and out of sample correlation goes down. This suggests that overfitting occurs between the leaf sizes of 1 and 15.

3.2 Experiment 2

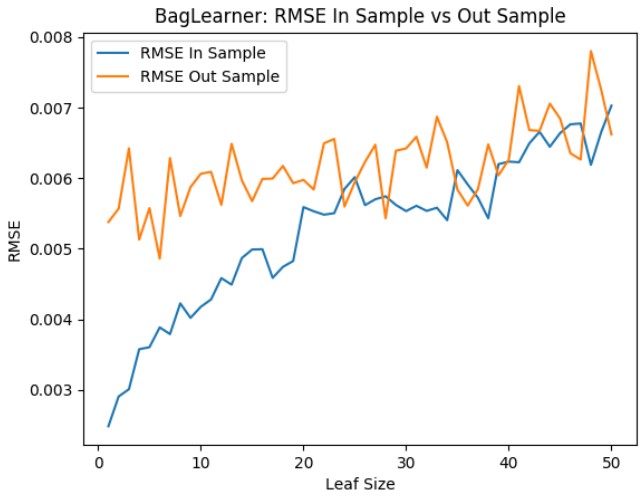

Figure 3—Bootstrap Aggregation Learner (BagLearner) of Decision Tree with 20 Bags. RMSE for In Sample and Out of Sample Data. Leaf Size of 1 - 50

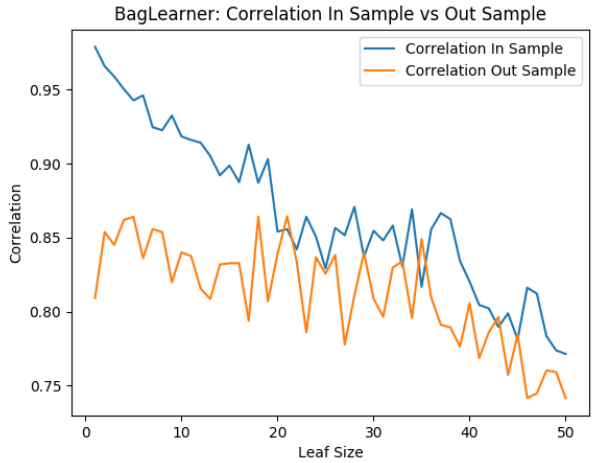

Figure 4—Bootstrap Aggregation Learner (BagLearner) of Decision Tree with 20 Bags. Correlation for In Sample and Out of Sample Data. Leaf Size of 1 - 50

For Experiment Two we can see that bagging is reducing overfitting. In some ways bagging had eliminated overfitting. We can see that from leaf size 1-15 RMSE and correlation for out of sample data has flatten out and stop improving. While the performance does not decrease between those leaf sizes it also does not increase for out of sample data. In sample data, as expected, continues to increase in those leaf sizes. This shows that leaf size can be reduced with bagging.

3.3 Experiment 3

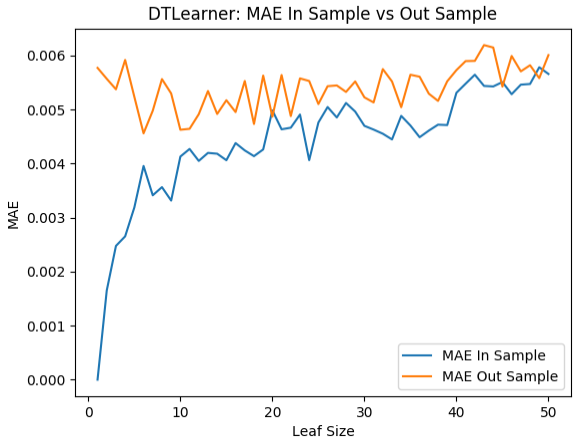

Figure 5—Decision Tree Learner MAE (Mean Absolute Error) for In Sample and Out of Sample Data. Leaf Size of 1 - 50

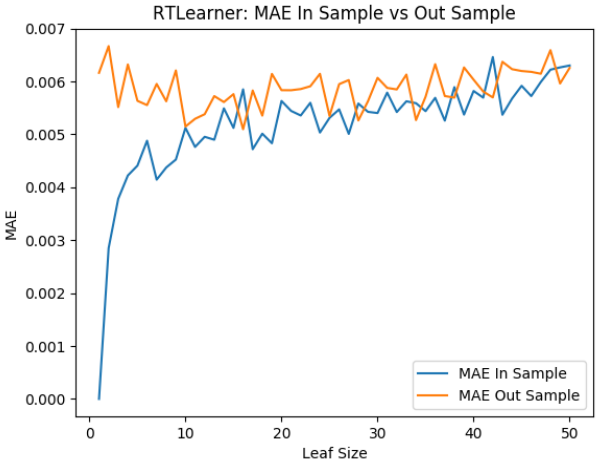

Figure 6—Random Tree Learner MAE (Mean Absolute Error) for In Sample and Out of Sample Data. Leaf Size of 1 - 50

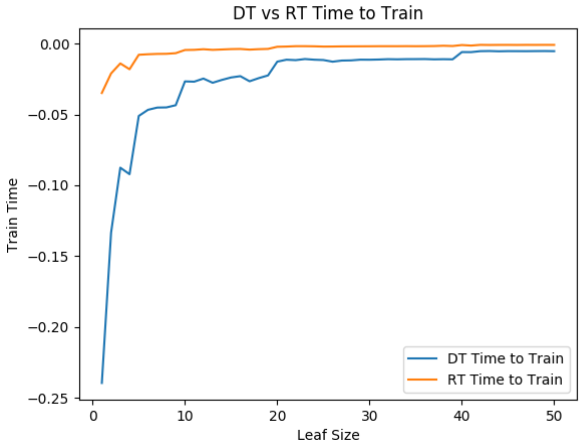

Figure 7—Decision Tree Learner vs Random Tree Learner Time to train. Leaf Size of 1 - 50

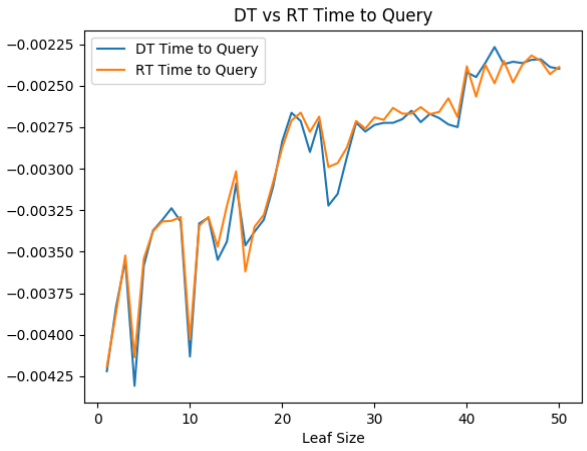

Figure 8—Decision Tree Learner vs Random Tree Learner Time to query. Leaf Size of 1 - 50

In Experiment Three we compare a Decision Tree Learner to a Random Tree Learner. Using Mean Absolute Error we can see that both our Decision Tree and our Random Tree both perform very similarly even overfitting at similar leaf sizes (a size of 8). Our Decision Tree very slightly out performs our Random Tree but the difference is very small. This mostly likely is the result of using correlation to determine which factor to split on. Next we see that our Random Tree consistently outperforms our Decision Tree in time to train, specially in lower leaf sizes. And both perform the same (as expected) in time to query. Overall because the performance gain of a Decision Tree is so small compared to a Random Tree and the time to train improvement from Decision to Random is so much greater, it is clear that Random Trees seem to be the better choice.

4 SUMMARY

Through these experiments we found that Decision Trees begin to overfit between leaf sizes 1-15 for the data in Data/Istanbul.csv. This effect can be reduced and almost eliminated with the use of bagging. We also found that Random Tree is a better overall choice for this data set than a Decision Tree.

5 REFERENCES

1. Am1rr3zAAm1rr3zA 6, ijmarshallijmarshall 3, Aminu KanoAminu Kano 2, & KeikuKeiku 7. (1960, June 1). Create a two-dimensional array with two one-dimensional arrays. Stack Overflow. Retrieved February 12, 2023, from https://stackoverflow.com/questions/17710672/create-a-two-dimensional-array-with-two-one-dimensional-arrays
2. Building the future with software. (n.d.). How to select random rows from a NumPy array in Python. Kite. Retrieved February 12, 2023, from https://www.adamsmith.haus/python/answers/how-to-select-random-rows-from-a-numpy-array-in-python
3. DanDan 97911 gold badge77 silver badges33 bronze badges, kennytmkennytm 504k104104 gold badges10681068 silver badges996996 bronze badges, KatrielKatriel 118k1919 gold badges134134 silver badges167167 bronze badges, PhilippPhilipp 47.2k1212 gold badges8484 silver badges108108 bronze badges, Arman AynaszyanArman Aynaszyan 3122 bronze badges, schwaterschwater 10511 silver badge11 bronze badge, & SergioSergio 2922 bronze badges. (1957, July 1). Why does Corrcoef return a matrix? Stack Overflow. Retrieved February 12, 2023, from https://stackoverflow.com/questions/3425439/why-does-corrcoef-return-a-matrix
4. GeeksforGeeks. (2020, December 4). Numpy.append() in Python. GeeksforGeeks. Retrieved February 12, 2023, from https://www.geeksforgeeks.org/numpy-append-python/
5. GeeksforGeeks. (2020, September 2). Compute Pearson product-moment correlation coefficients of two given NumPy arrays. GeeksforGeeks. Retrieved February 12, 2023, from https://www.geeksforgeeks.org/compute-pearson-product-moment-correlation-coefficients-of-two-given-numpy-arrays/
6. GeeksforGeeks. (2021, November 28). How to calculate mean absolute error in python? GeeksforGeeks. Retrieved February 12, 2023, from https://www.geeksforgeeks.org/how-to-calculate-mean-absolute-error-in-python/

7. GeeksforGeeks. (2022, May 13). Python - iterate over columns in NumPy. GeeksforGeeks. Retrieved February 12, 2023, from https://www.geeksforgeeks.org/python-iterate-over-columns-in-numpy/
8. GeeksforGeeks. (2022, September 1). How to check the execution time of python script ? GeeksforGeeks. Retrieved February 12, 2023, from https://www.geeksforgeeks.org/how-to-check-the-execution-time-of-python-script/
9. Panjeh. (2020, June 22). How to get average of rows, columns in a Numpy array. Medium. Retrieved February 12, 2023, from https://panjeh.medium.com/how-to-get-average-of-rows-columns-in-a-numpy-array-8f305dd92624
10. Vishal. (2022, October 1). Python random randrange() and randint() to generate random integer number within a range. PYnative. Retrieved February 12, 2023, from https://pynative.com/python-random-randrange/#:~:text=within%20a%20range-,Use%20a%20random.,4%2C%206%2C%208).
11. What is the preferred method to check for an empty array in NumPy. TutorialsPoint. (n.d.). Retrieved February 12, 2023, from https://www.tutorialspoint.com/what-is-the-preferred-method-to-check-for-an-empty-array-in-numpy#:~:text=Use%20the%20numpy.&text=any()%20function%20determines%20if%20any%20array%20members%20along%20the,otherwise%20it%20is%20not%20empty.