

MC3-P3

Dominique Paster

11/28/16

Technical Indicators

I chose to use four technical indicators to decide when to buy and sell IBM stock. The four indicators are price/ Simple Moving Average (SMA), momentum, Bollinger Bands, and Moving Average Convergence Divergence (MACD). I use a 20-day window for all of the indicators with the exception of MACD. 20 days gave the best results from the tests ran.

Price/SMA

To calculate the price/SMA, I take all the prices of a stock and divide them by the SMA of the stock with a 20-day window. Calculate the SMA by doing a 20-day rolling mean of the prices value. The value given from the calculation is then subtracted by one to keep the values mostly in between -0.5 and 0.5. A chart is shown below for IBM. The chart shows the price and SMA starting at 0 and the normalized price from that starting value. It also shows the price/SMA in red starting at 0.

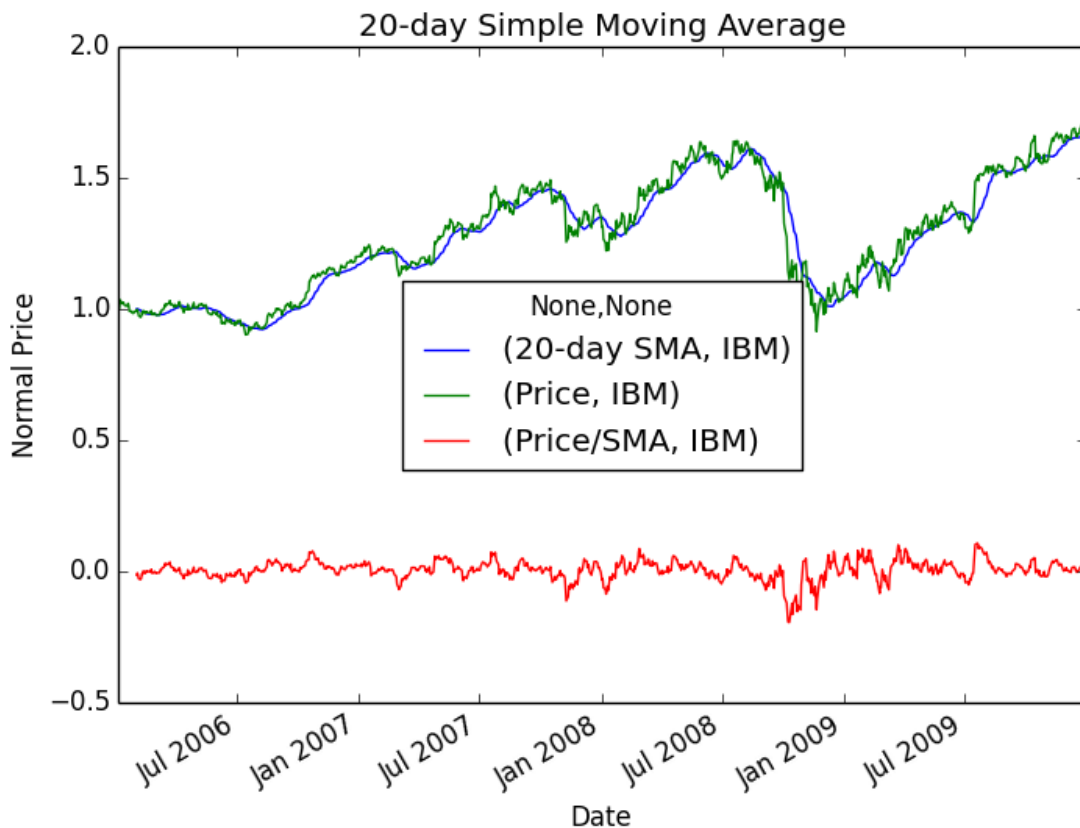


Figure 1. 20-day simple moving average with price and price/SMA during the training period.

Momentum

To calculate the momentum for a 20-day period, I shift the price values by 20 and then divide that by the original price values. This will give you a trend of where the price is moving over the next 20 days. In the chart below the price normalized and shifted down to zero are shown along with the momentum values.

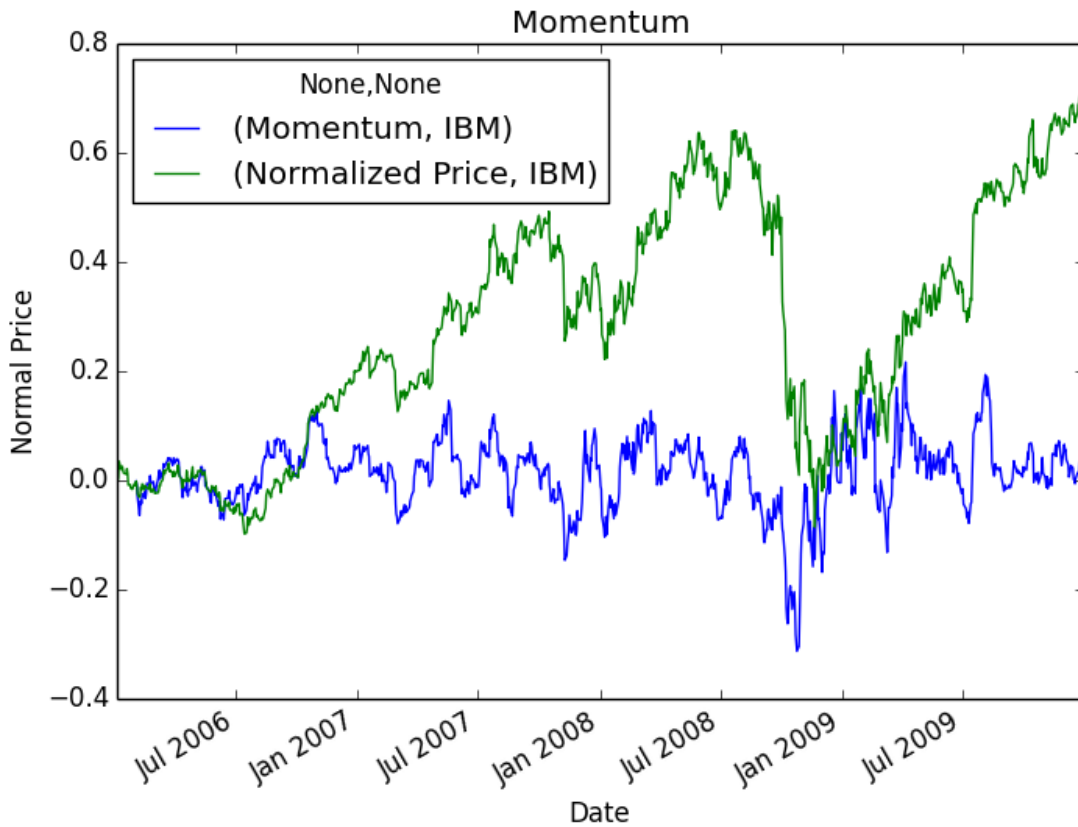


Figure 2. Momentum and normalized price during the training period.

Bollinger Bands

To calculate the value for Bollinger Bands, I start by calculating the moving 20-day standard deviation. I then take the prices and subtract the 20-day moving average calculated earlier for use with calculation of the SMA. Then this value is taken and divided by the standard deviation (calculated earlier) multiplied by two. The values are roughly between -2 and 2. Graphically I prefer to use the an upper and lower band which I calculate by taking the moving average and subtracting the standard deviation multiplied by two and inversely the upper band is calculated by adding the standard deviation multiplied by two to the moving average. This makes it easier to determine when the price deviates from the normal curve visually. The chart below shows price, SMA, upper and lower Bollinger bands.

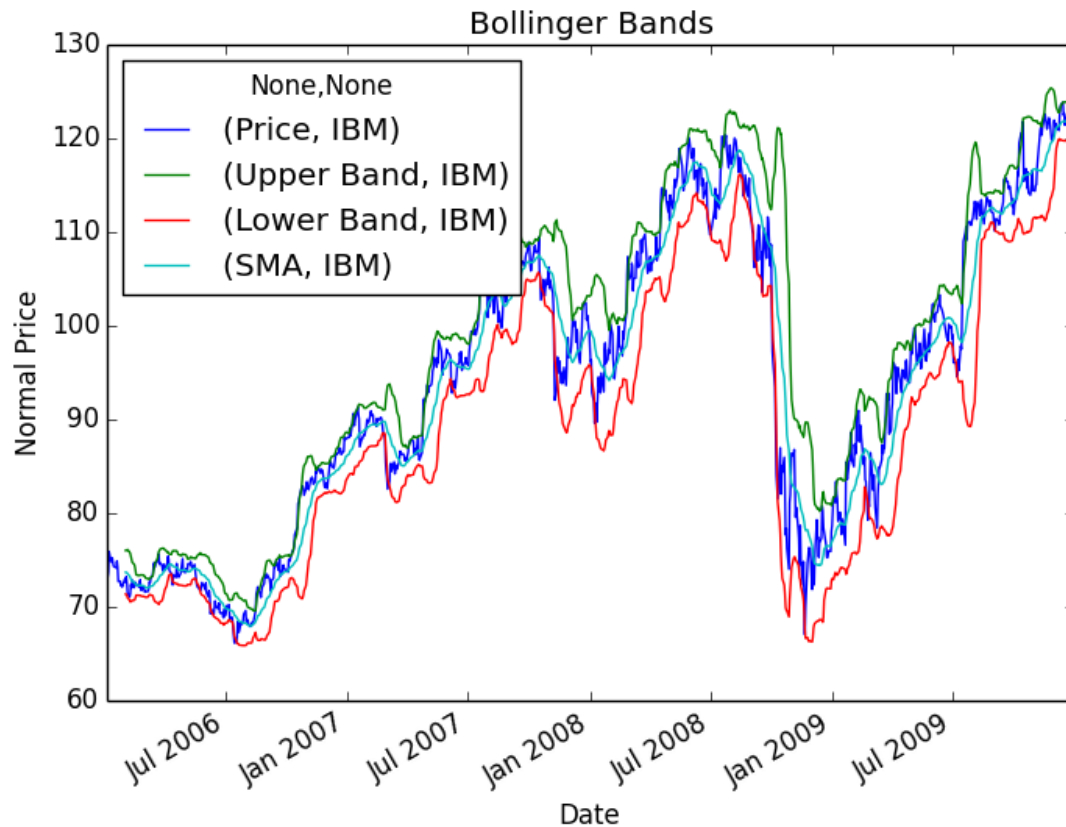


Figure 3. Bollinger bands, SMA, and price during the training period.

MACD

To calculate MACD, I found multiple weighted moving averages for the prices data. There is a weighted average function in the python library but I also did some research to learn more about how an exponential weighted average is calculated. It takes the mean but puts more emphasis on the last calculated number. I use the function because it was easier to understand and follow. First calculate the 12-day weighted average and 26-day weighted average. Then subtract the 26-day weighted average from the 12-day weighted average to get the known MACD. Then, take the 9-day weighted average of the MACD to gain a signal line to determine when to buy or sell. It is known to buy or sell when these lines cross each other. There are multiple crosses shown in the chart below.

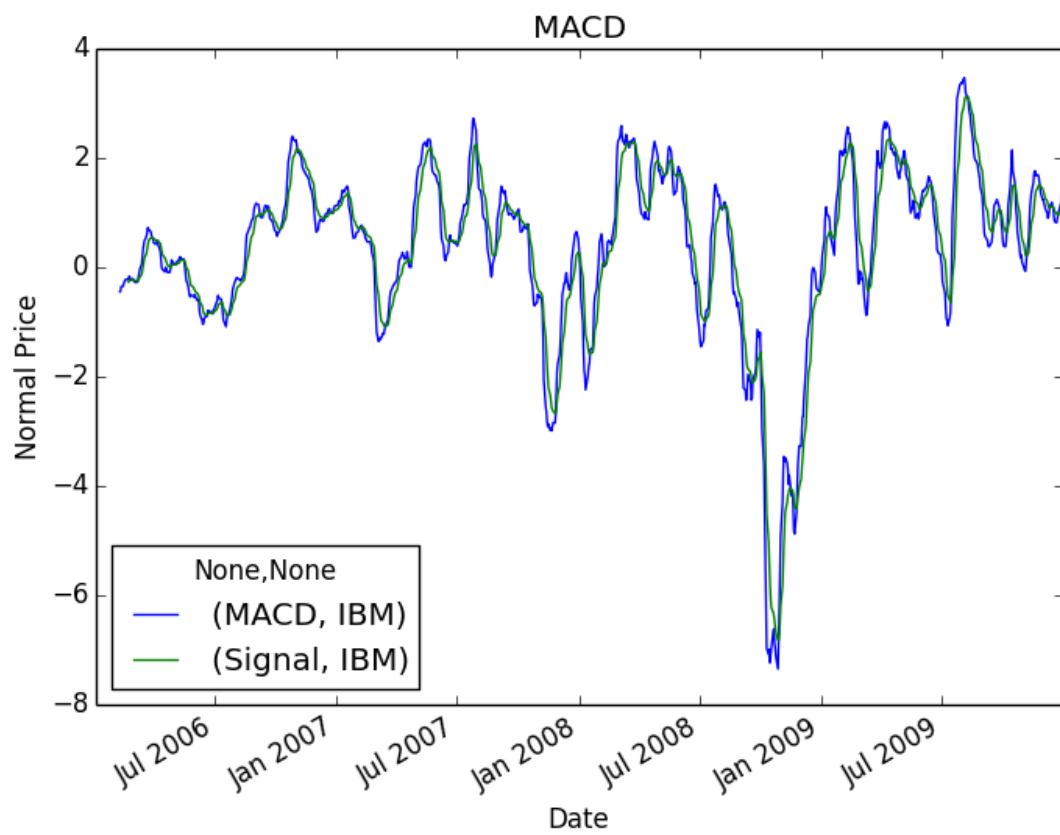


Figure 4. MACD and signal line during the training period.

Manual Rule-Based Trader

The created rule set has two components to determine whether a trade occurs or not. The first component uses Bollinger Bands, price/SMA, and momentum. After doing some research, Bollinger Bands was a strong indicator to decide when to buy or sell. Price/SMA and momentum were weaker indicators so they are combined in the first component. If the Bollinger Band value is greater than 1 this signals a sell or less than -1 for a buy signal. These trends will occur when the current price is above or below two standard deviations. I am doing this more dangerously than described in the lecture which suggests that the trader should wait until the price enters within two standard deviations. If this value signals a buy or sell, then the program will enter the second component. The first component also checks momentum and price/SMA. For a buy to occur the momentum has to be greater than 0 which shows a positive trend is occurring and the price/SMA is less than 0 which shows that the current value is below the mean value for 20 days. For a sell to occur the momentum has to be less than 0 which shows a negative trend is occurring and the price/SMA is greater than 0 which shows that the current value is above the mean value for 20 days. After either of these conditions pass then the second phase needs to be passed for a buy or sell to occur. The second phase only uses the MACD and signal line to determine a buy or sell. I subtract the signal line from the MACD both shown in Figure 4. If this number goes from positive to negative or from negative to positive over three days (including current day), then a buy or sell can occur. This has given me good results as shown in Figure 5 below. The rule-based trader outperforms a benchmark which buys on the first day and sells on the last day.

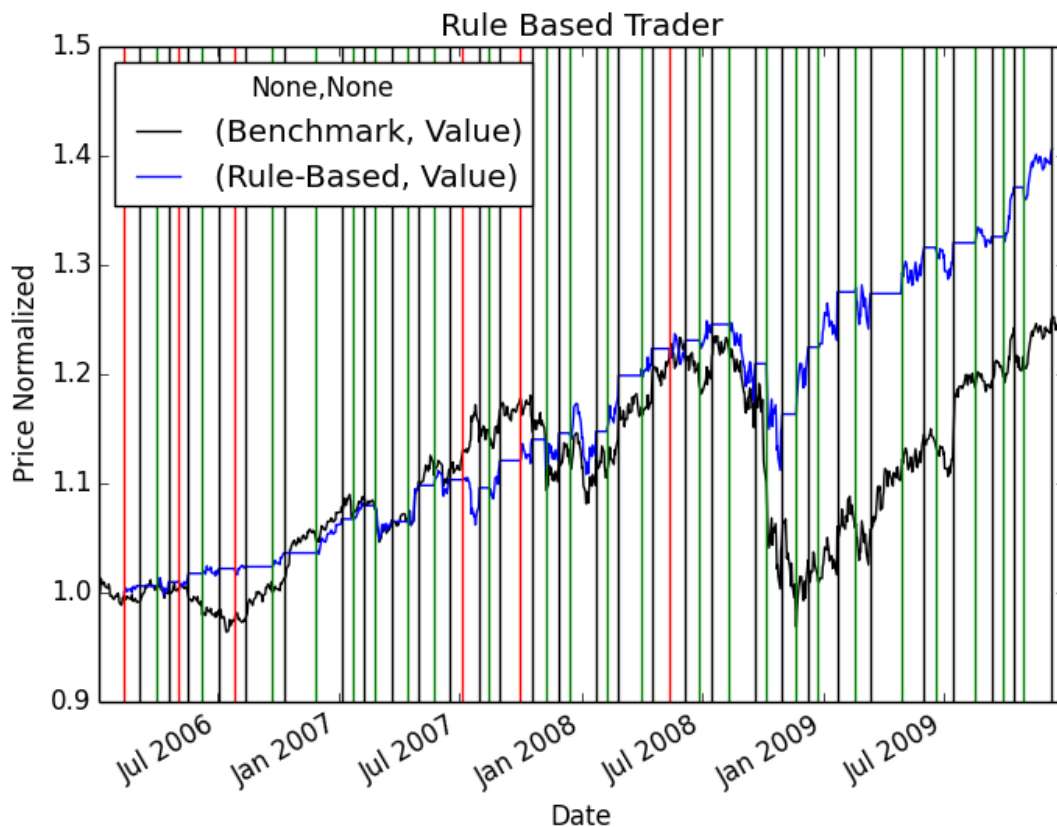


Figure 5. Rule Based trader compared to the benchmark during the training period.

Machine Learning Trader

I started by getting the indicator values from the Technical Indicators section above and concatenating them along the Y axis to form a matrix. I then take the prices array from the list and shift it to the left by ten and divide it by itself to get predicted y return for ten days. I then buy if the return is greater the 3% and sell if the return is less the -2%. If the return does not meet either condition the program holds the stock. I then turn this into an array and pass it as training data to my random tree classifier. The learner has a leaf size of 5. The learner was modified to use the mode instead of the mean when the number of items is less than or equal to leaf size. This is to ensure that every leaf node will have a 1,0, -1 as the value. The classifier trains and query for this data and the results are then turned into an orders file. The order method accepts all valid orders (-1 sell,1 buy) and guarantees that a position is held for 10 days. If the order has a value that is not 1 or -1 it is considered a hold. The results of the order file on the market are shown below in Figure 6. As expected the Machine Learning trader outperforms both the rule-based trader and benchmark. I think part of this is due to the fact that it uses future data to make decisions. I tried changing the leaf size to reduce overfitting along with adding a seed to better understand the results. I also tried using a bagging learner to reduce overfitting to the training set but gave worse results as I changed the parameters. This could be due to I am testing with the training sample for this data.

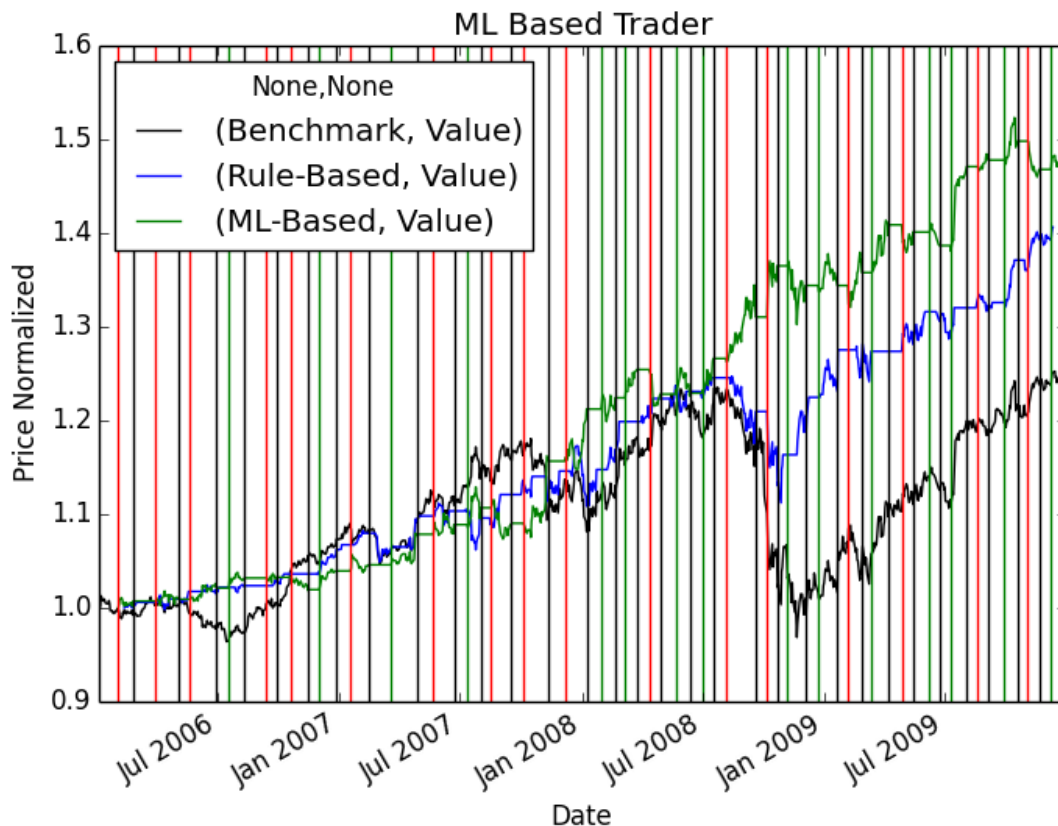


Figure 6. ML Based trader compare to the rule-based trader and benchmark during the training period.

Comparative Analysis

The results of using the out of sample data were surprising. The Machine Learning (ML) based trader performed better than the benchmark and the rule-based trader. The more surprising fact is the rule-based learner performed worse than the benchmark. This could be due to the fact that it takes 34 of the 252 trading day to analyze any data. The rule based learner out performs the benchmark for almost all of the dates until the end and during October for a few days. As shown in the chart below. I believe the troublesome area is the period from mid-August to November where the results are steady. This could be caused by the rules being strict. I think also some of the results are due to the fact of the RTLearner being random and not using correlation to help pick the decision. I also believe a major difference in performance is in the difference in time from 3 years training to only run for a year worth of time. Given more time I believe my Learners will get closer performance to in sample results. Also adding correlation can improve results.

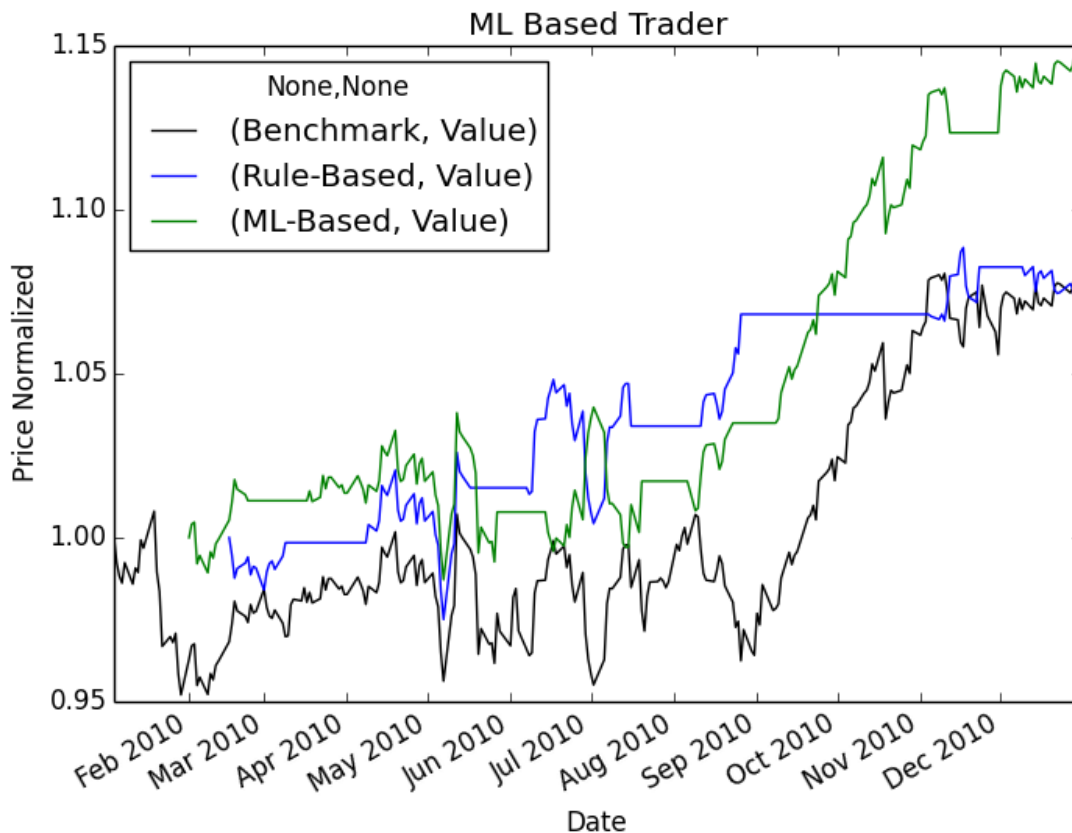


Figure 7. ML Based trader compare to the rule-based trader and benchmark during the testing period.

Table 1. Table showing results for all learners in and out of sample.

	Benchmark	Rule-Based	ML-Based
In Sample	125715	140625	149420
Out of Sample	108155	107055	114920