CS7646

Gianfranco Dutka
gdutka3@gatech.edu

***Abstract*—** The purpose of this paper is to investigate the difference in performance of two different trading strategies and implementations. The first strategy will be a manual strategy that will use a few different statistical indicators to determine whether the price will go up or down. The second strategy will use the same indicators but will attempt to predict the change in price using a supervised learning model known as a random forest. The performance of these strategies will be compared against one another and against a benchmark that will be used as a point of reference for all the strategies mentioned above.

1 INTRODUCTION

Automated trading strategies have been an important development in the last several decades for most large hedge funds. The purpose of this experiment is to determine the impact supervised learning can have on an automatic strategy that uses basic statistical indicators as a way of predicting stock prices.

The experiments in this project describes three different strategies that will be compared. The first will be known as the ManualStrategy and it will place long or short positions based on three statistical indicators. The first indicator is the simple moving average ratio, the second is the bollinger band percentage, and the third is the exponential moving average convergence/divergence.

The second automated strategy will be known as the StrategyLearner. It will be similar to the first manual strategy but will use the assistance of a supervised learning algorithm in addition to the indicators mentioned above to determine when to place the long/short positions. The supervised learning algorithm that will be used is known as a random forest. A random forest is very similar to the random decision tree from project 3 with the exception that it will be slightly modified into a classification tree which will aim to predict whether the price will go up/down or stay the same.

Finally, all strategies will be compared against a benchmark which will be a simple buy and hold strategy. This strategy will just hold 1000 shares of a certain stock for the entire time-period we are investigating. The last experiment that will be conducted will rerun multiple trials of the strategy learner and slowly increase the impact value at training time. The results will then be analyzed to determine how the impact changes the performance of the model.

2 INDICATOR OVERVIEW

The first and most simple indicator we will be using is the simple moving average price ratio. The formula for the simple moving average is the following:

$$SMA = \frac{P_1 + P_2 + ... + P_n}{n}$$

Where n is the number of days in a given window and P_n is the nth day in the window. Taking the sum of every day in the window up to P_n and dividing it by the number of days n will give us the average for that window[1]. The window used to determine the sma will be 20 trading days. To get the ratio, the sma would be divided by the price level at the same date.

The second indicator used in the manual and strategy learners is the Bollinger Band percentage. Originally in project 6 the bollinger bands were used by themselves, however when coming up with the best buy/sell point, a standardized result using the bollinger bands alone proved difficult. Since it was not straightforward to standardize all the lines used within the indicator, the original bollinger bands used in project 6 were changed to the bollinger band percentage. Normally, each of the lines follows two standard deviations above and below the simple moving average. For our experiment this was also changed so that each bollinger band lies 1.5 standard deviations above and below the 10 day simple-moving average. This seemed to be more reliable when automating the manual strategy. After much tweaking of the indicators.py file the formula then becomes the following:

$$upperBand = SMA_{10} + 1.5 * s$$
$$lowerBand = SMA_{10} - 1.5 * s$$

The final indicator that was used was the exponential moving average convergence/divergence. This indicator was left untouched from project 6, however, it was standardized in the following way. The difference between the macd and mac lines were taken after shifting the macd line forward one day. If there was a line crossover and the macd line crossed the signal line a 1 was assigned to the indicators dataframe. If a crossover happened in the opposite direction a -1 was assigned instead. If 1 would denote when to take a long position, and a -1 would denote when to take a short position instead. The formula for the macd stayed the same between project 6 and project 8[2]:

$$macd = 12 - dayEMA - 26 - dayEma$$
$$macd_{signal} = P_i * k + EMA(y) * (1 - k)$$
$$where\, k = 2 \div (N + 1)$$

3 MANUAL STRATEGY

In order to come up with an effective buy and sell strategy the indicators for the in-sample price window had to be calculated. The start and end dates for the in-sample period corresponded to January 1, 2008 and December 31 2009. Once the indicators were calculated, thresholds for each indicator had to be determined in order to convert them into a buy/sell signal. This was done differently for each indicator and required looking at the price movement of 'JPM' over the in-sample period against each indicator. For the sma-ratio, the threshold for a buy and sell signal was 0.955 and 1.055 respectively. If the sma-ratio fell below 0.955 that would cause the automation algorithm to assign a 1 to the signals dataframe. The opposite would happen if the sma-ratio went above 1.055. For bollinger bands percentage the threshold was -0.03 for the buy signal and 1.03 for the sell signal. The buy and sell thresholds for the macd were more difficult to calculate and required iterating over each in-sample date and determining whether a crossover happened. If a crossover occurred a 1 for buy and a -1 for sell was assigned to the signals dataframe. Once the signals dataframe was completed the strategy iterated over each sample. If any of the signals from the indicators indicated a buy signal, a long position was executed and stored within a "trades" dataframe. If any of the signals indicated a price decrease a short position was executed and stored into the "trades" dataframe.

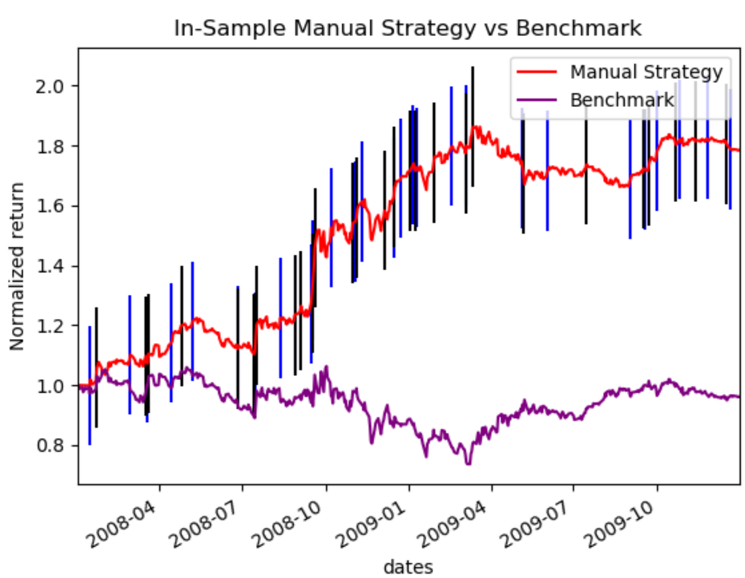

Figure 1— Figure 1: shows the normalized return between the manual strategy and the benchmark portfolio using a simple buy and hold strategy.

To calculate the returns for the manualStrategy positions, the trades dataframe was passed through the market simulator for the in-sample dates. The market simulator returned the portfolio values for the manual strategy and benchmark portfolio corresponding to just holding 1000 shares of "JPM" for a year. Figure 1 displays the returns of the manual strategy and the benchmark portfolio side-by–side. We can see that the manual strategy had a much better cumulative return of about %78 percent when compared with the benchmark which returned a bit below the starting value years prior.

The returns change drastically when going from the in-sample period to the out-of-sample period although the end-result is the same. The manual strategy, through the use of the 3 indicators mentioned above, performs much better than the benchmark even in the out-of-sample period. Despite the better performance when compared to the benchmark, the out-of-sample period does perform worse than the same strategy executed in-sample. This could largely be attributed to the in-sample period not being representative with the out-of-sample data. Meaning that a lot of the insight provided by the in-sample data might not transfer to the out-of-sample data causing the strategy to have significantly degraded performance. Figure 2 shows this change. Although the performance for the out-of-sample prices degrades performance, the returns for the in-sample period are still about %10 as opposed to the benchmark which is giving negative returns.

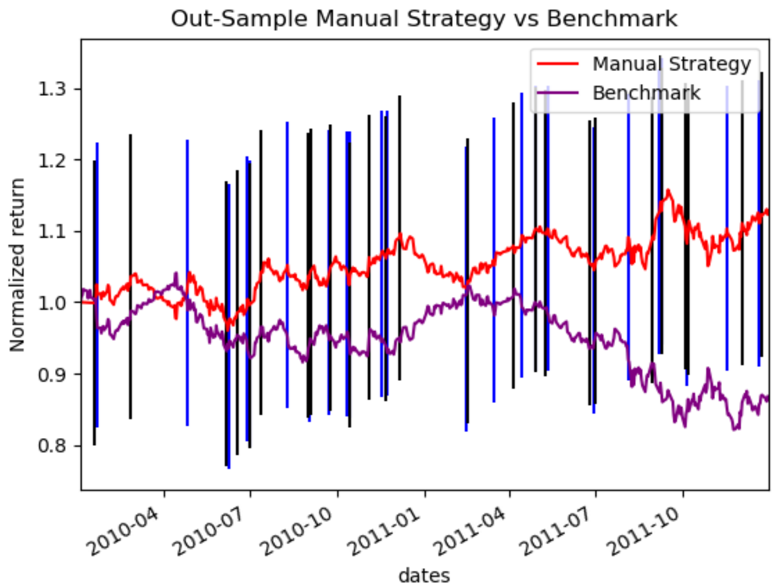

Figure 2— Figure 2 shows the commodity channel index alongside the normalized price of JPM. The buy and sell signals are denoted in the dashed black lines in the plot with the cci line.

```
******** in sample manual strategy ************
Sharpe Ratio of portfolio: [1.73080959]
Cumulative Return of portfolio: [0.78564942]
Standard Deviation of portfolio: 0.011119918395908411
Average Daily Return of portfolio: [0.00121241]
Final Portfolio Value: 178547.17499999685

***********************************************

******** in sample benchmark ************
Sharpe Ratio of portfolio: [0.06795797]
Cumulative Return of portfolio: [-0.03785905]
Standard Deviation of portfolio: 0.017435058892808206
Average Daily Return of portfolio: [7.46386107e-05]
Final Portfolio Value: 96186.01500000147

***********************************************

******** Out-sample manual strategy ************
Sharpe Ratio of portfolio: [0.54526196]
Cumulative Return of portfolio: [0.12411345]
Standard Deviation of portfolio: 0.007613964630680384
Average Daily Return of portfolio: [0.00026153]
Final Portfolio Value: 112400.15999999785

***********************************************

******** out-sample benchmark ************
Sharpe Ratio of portfolio: [-0.44647809]
Cumulative Return of portfolio: [-0.13348906]
Standard Deviation of portfolio: 0.008763152341131456
Average Daily Return of portfolio: [-0.00024647]
Final Portfolio Value: 86624.76500000147

***********************************************
```

Figure 3— Figure 3 shows the sharpe ratio, cumulative return, standard deviation and average return for the in-sample, out-of-sample JPM data for the manual strategy.

Looking at figures 1 and 2 one can see the trades that allowed the manual strategy to excel when compared to the benchmark. The blue vertical lines represent long positions when it was predicted that prices would go up. The black vertical lines represent changing strategies to a short position.

Figure 3 also sheds some more detail on the other statistics between both strategies. We can confirm by looking at the strategy statistics that the volatility for both the in-sample and out-of-sample manual strategy is lower than the simple buy and hold strategy despite having to make more trades. In addition, we see that the manual strategy has a sharpe ratio of 1.73 and 0.54 for the in-sample and out-of-sample data. The benchmark strategy of just buying and holding has a negative return for both the in-sample and out-of-sample period.

4 STRATEGY LEARNER

The strategy learner used a random forest paired with the same indicators used in the manual strategy in an attempt to determine when to place the long/short positions. Since the random forest is a supervised learning approach, the learner had to be "taught" how to determine a strategy. This was done by calculating the 15 day return from every date within our in-sample period used when training the data. If this return exceeded some positive or negative threshold the appropriate target value was selected for the date where a 1 corresponded to a long position and a -1 corresponded to a short position.

Coming up with the appropriate target values for each of the in-sample dates and for each of the indicator values allows the random forest learner to create the appropriate model that would correctly allow the strategy learner to predict whether values would go up or down. The appropriate long/short strategy would then be executed. Changes to the random decision tree from project 3 were made to turn it from a regression tree to a classification tree. After randomly picking a feature from our dataset the mode instead of the mean was used to determine whether or not the random tree would predict a long, short or neutral position.

The hyper parameters included the indicators used and the way some of them were discretized. The threshold for setting the target as a 1/-1 is also another hyper parameter. The leaf size used is another hyper parameter which allows us to avoid overfitting. The bag size used is another hyper parameter. For this experiment a bag size of 13 was selected with a leaf size of 5.

Most of the indicator data did not have to be discretized. The sma-ratio already was an appropriate floating point number and so was the Bollinger band percentage. That was not the case for the moving average convergence/divergence. To standardize the macd, the difference between the macd and the signal line were taken to come up with a singular value that the random forest could use. This singular value is representative of the Macd line being above or below the signal line. This turned out to be useful enough for the strategy learner to detect a signal from the indicator and use it for an appropriate long/short classification.

5 EXPERIMENT 1

Experiment 1 consisted of comparing all three strategies implemented in the above sections. The stock ticker "JPM" was used to compare the performance of the manual strategy, the strategy learner and the benchmark. All three strategies started with 10000 dollars and the in-sample period covered the dates starting from January 1, 2008 until December 31, 2009. My hypothesis for the experiment was that the random forest learner would greatly outperform all strategies by a significant amount in the in-sample and out-of-sample periods.

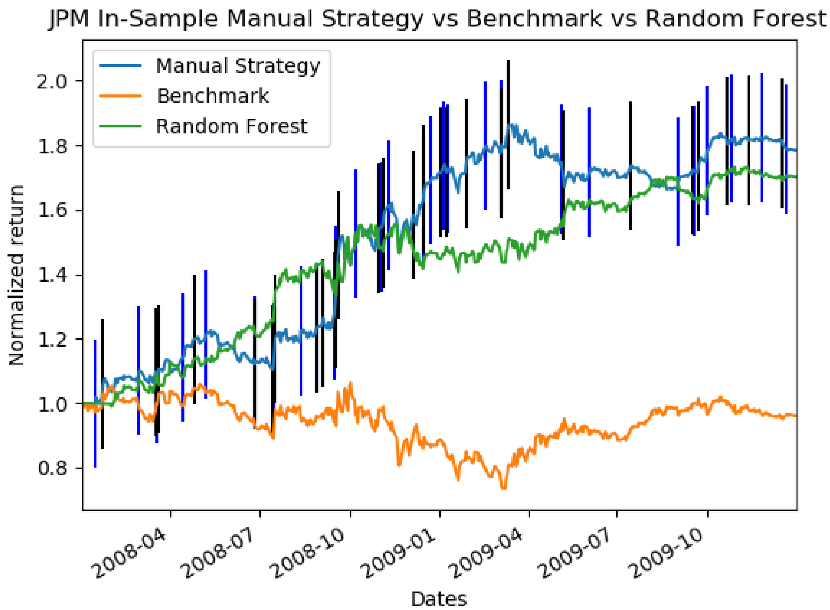

Figure 4— Figure 4 shows the performance of the manual strategy against the benchmark and the strategy learner for the in-sample period. For the given period the manual strategy slightly outperforms the random forest. Both the manual strategy and random forest greatly outperform the benchmark.

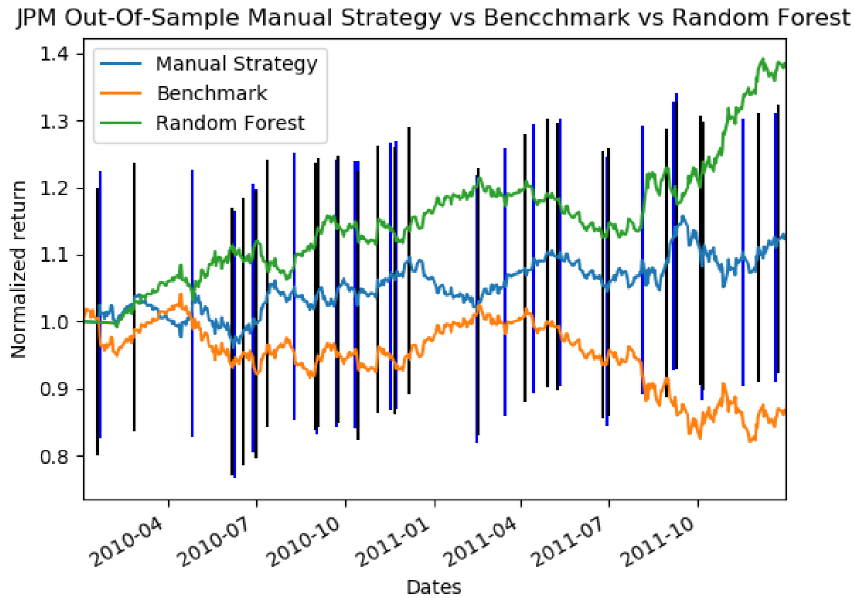

Figure 5— Figure 5 shows the performance of all three strategies against one another. For the out-of-sample period with "JPM" the random forest greatly outperforms the manual strategy and the benchmark.

As you can see from figure 4, the performance of the random forest and the manual trading strategy were almost identical. My hypothesis turned out to be correct for the out-of-sample period but not for the in-sample period since the manual strategy outperformed the strategy learner. The manual strategy starts out with higher returns for the first half of the insample strategy however the returns ultimately fall while the random forest returns rise throughout the entire window in a more linear fashion. Essentially, the risk-adjusted return for the random forest is higher since it achieves a slightly greater return than the manual strategy with much less volatility. On the contrary the benchmark performed the worst out of the three, remaining flat throughout most of the in-sample period.

The second part of experiment1 used the out-of-sample period between January 1 2010 and December 31 2011 for the stock ticker "JPM". Here we see a stark difference in performance mainly between the manual strategy and the strategy learner.The strategy learner greatly outperformed the manual strategy by a large margin. It seems from the second part of experiment1 that the strategy learner managed to better generalize the in-sample data to extract better signals that allowed it to outperform in the out-of-sample period.

6 EXPERIMENT 2

For experiment two the in-sample JPM period was used to gauge the effect changing the impact would have on the performance of the model. My hypothesis for experiment 2 suggested that as the impact would go up, the performance of the model would be significantly degraded. Part of the reason for this is that the impact affects the buy and sell threshold used to determine the target for the model. This might worsen the predicting power of the model and its indicators.

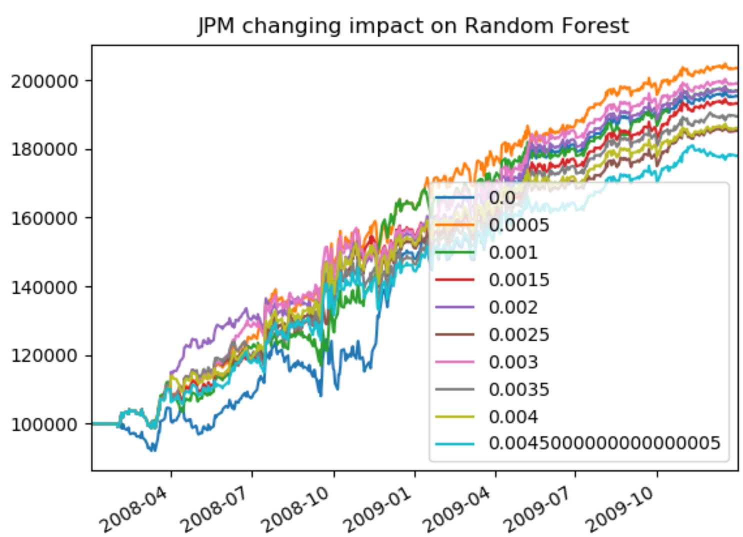

Figure 6— Figure 6 shows the effect on return as the impact increases for each trial.

As you can see from figure 6 the hypothesis was correct. The highest returning trial was in fact the trail that had the second lowest impact value. The trend seems to be that every trial after the first one has lower returns than the previous trial as we increase the impact value. The lowest return was seen by the trial with the highest impact at 0.0045. The highest return was seen by the trial with 0.0005. It was expected that the trial that had an impact of zero would perform better although that trial actually performed worse than many of the other trials that had higher impact. The reason for this effect is likely because of the way the buying and selling targets are calculated. When the targets are being calculated, the impact moves the threshold for both of these parameters up and down. If this value moves to a point that does not reflect real life, it might degrade the quality of the model once it is trained. Finally, one can see how degraded the performance becomes by looking at some other statistics after conducting the trials. The trials with the highest impact ratio had a sharpe ratio of about 1.6 with a return of 0.75. This is a sharp decrease in sharpe ratio and cumulative return since some of the trials that were run with a lower impact value have a sharpe ratio above 2.0, and returns above 1.0 for the in-sample period.

The final potential explanation for the drop in performance as the impact increases is simply from the increased cost of having to execute transactions. If this could be mitigated by improving the returns on each trade and therefore performing less trades overall, then the higher impact effect could be avoided. Unfortunately, while the returns were positive for all trials, it seems like the random forest implemented in this project had some trouble maintaining higher performance as the impact increased

7 REFERENCES

1. “Moving average.” *Wikipedia*, https://en.wikipedia.org/wiki/Moving_average. Accessed 22 October 2023.
2. “MACD.” *Wikipedia*, https://en.wikipedia.org/wiki/MACD. Accessed 22 October 2023..