Project 8 : Strategy Evaluation

Saad Rasool
srasool7@gatech.edu

Abstract—The research below represents 3 main learning objectives: trading solution, trading policy comparison and foundation for continued learning. Basically synthesizes the investing and machine learning concepts and integrates many of the technical components developed in prior projects. Another component is evaluation of the performance of a manual strategy with that of an AI-learner to better understand its behavior.

Introduction

In the research below, 3 out 5 indicators developed in project 6 were used, these indicators are ema crossover (12 and 26), momentum (5 days) and Bollinger bands %. A manual strategy and a classification based learner was developed and compared below.

Indicators Overview:

Indicators 1: Ema Crossover

Ema (exponential moving average) crossover indicator is very similar to golden and death cross indicator. The main idea behind using ema is for mover short term trading strategy. For our ema indicator we are using ema of 12 and 26 days. Ema is mostly used as it puts more weight to recent price changes hence the response is much faster then sma (simple moving averages)

We can generate buy or sell signals when the 12 day ema cross the 26 day ema line. If the 12 day ema is higher than 26 day ema and cross over to be lower then that would be a sell signal. On the other hand is the 12 day ema is lower than 26 day ema , and crosses over to be higher then that would be a buy signal.

In the manual strategy a buy signal was created when curr_ema <= 0.98 and the difference between current ema ratio and yesterday's ema ratio was < 0. A sell signal was created when current ema ratio was >= 1.02 and the difference between current ema ratio and yesterday's ema ratio >=0.

In the strategy learner the ema ratio was provided to the random forest to train the model.

Indicators: 2 Momentum

Momentum indicator demonstrates the speed at which is the price is changing, the indicator is used for more short term trades. A buy signal can be generated as the momentum of the prices is increasing and sell indicator can be generating is the momentum of the stock is decreasing. The indicator, should be used with other trading indicator. Using this indicator is like following others and what the market is doing. I think including this indicator with others mentioned above could significantly help develop a reasonable trading strategy. Momentum does require a window, in our case we are using 5 days as I wanted to include some short term trading strategies.

For Manual Strategy: A buy signal was created when the current momentum < 1 and a sell indicator was created if the current momentum was >=1.02

In the strategy learner the momentum value was provided to train the random forest classifier

Indicators 3: Bollinger bands percent

Bollinger bands is a long term trading strategy, it consists of a simple moving avg line, along with the upper and lower bound. The indicator does give us buy and sell signals. If the price of the stock continually touches the upper bound, it could mean the stock is overvalued and should be sold. On the other hand if the price touches the lower bound it could indicate that the stock is undervalued and could be a buy signal. Bollinger bands along with some of the short term trading indicators mentioned above, could produce reasonable results to maximize returns. Upper and lower bands are calculated based on the volatility of the stock price, via using standard deviation.

For Manual Strategy: A buy signal was created when the bollinger band percent < 0.0 and a sell signal was created when > 1.01

For the strategy learner the bollinger band percent was provided to train the random forest classifier.

Combining all 3 of these indicator allowed us to develop manual strategy and strategy learner.

Manual Strategy:

<u>Describe:</u>

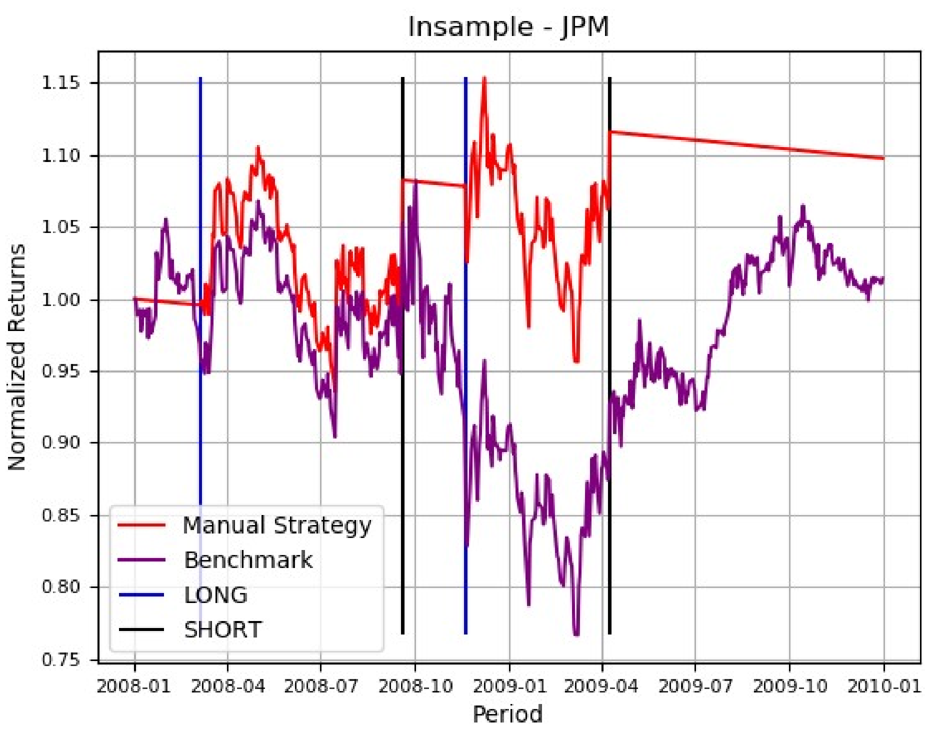
The above combination of indicators were used, the overall signal of buy and sell was generated by the following code:

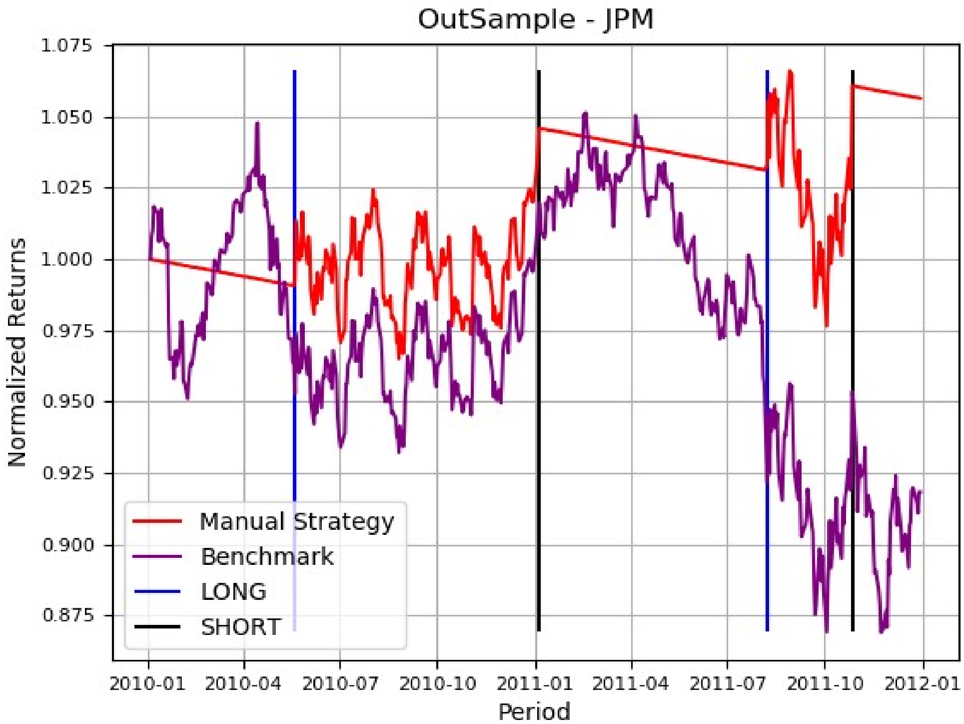
buy signal: if curr_bollinger% < 0.0 and curr_momentum < 1 and ema_diff < 0 and curr_ema < 0.98

sell signal: if curr_bollinger% > 1.01 and curr_momentum >=1.02 and ema_diff >= 0 and curr_ema >= 1.02

These thresholds were generated by trail and errors for the best outcome and investigating the trends. This is a very conservative strategy, but I still believe there is further improvements that can be made, which can further validate the strategy.

<u>Compare</u>:





As the we can see from the plots above, in the in sample and out of sample data when comparing manual strategy with the benchmark., the manual strategy does out perform the benchmark. We can also observe that there are not significant number of trades that we were made and this is due to threshold set for buy and sell signal. Like as I mentioned there is further improvement to the strategy that could have been implemented.

Metric	Manual Strategy – Insample	Manual Strategy – Outsample
Cumulative Returns	0.098	0.06
STDEV	0.012	0.006
Mean Daily Return	0.003	0.00013

Strategy Learner

For the implementation of the strategy learner , I decided to use the Rllearner which is a classifier based learner, random forest along with the bag learner.

<u>RTLearner Hyperparameters:</u>

lookforward_period = 7 days, which is used during the training period of the learner. It allows the algorithm to look forward certain amount of days to develop sell , buy or do nothing signals during the training period

price_threshold = 0.02, basically the gain in price to sell the stock

leaf = 20 , leaf size of the random forest learner

bag_size , bag size of 20 was used, meaning 20 RT Learners were generated, trained and used for prediction

These hyperparameters were determined during the training period of the code and were kept the same in the testing period. The goal of the hyperparameters optimization was to obtain the highest amount of portfolio value.

First we had to convert the regression RTLearner to Classifier by changing the mean to mode in the bag and RTLearner code. This was done in order to obtain buy , sell or do nothing signals (1, 0 , -1)

The most important aspect of this process was to create a dataframe with the indicators selected in our case ema_ratio, momentum and bollingerband percent as X_Train and use the price of the stock along with the look_forward period to determine the signals Y_train. Then we trained the RTLearner with the implementation of bag learner to determine a classifier algorithm that could predict when to buy or sell based on the training data set provided.

The trained algorithm was then used for the out-sample data and for other stocks.

The X_train data was already between 0 and 1, as most of them were ration or percentage, hence no standardization was used or implemented.

Experiment 1 Manual Strategy / Strategy Learner:

Describe:

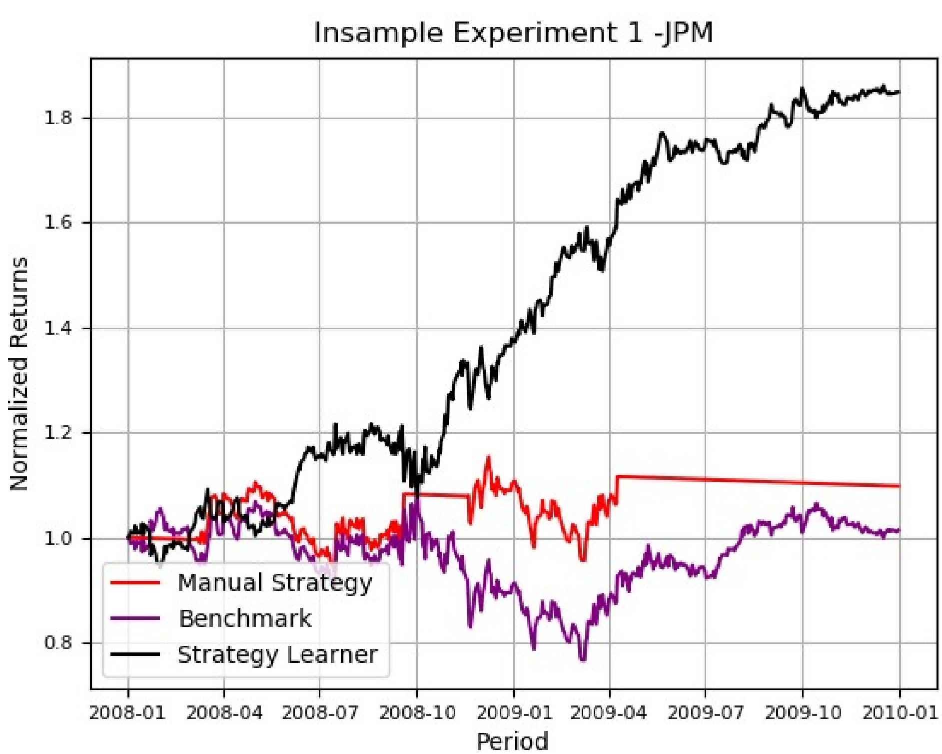
The main objective of this experiment was to compare the manual and strategy learner, with in-sample and out-sample data for JPM stock.

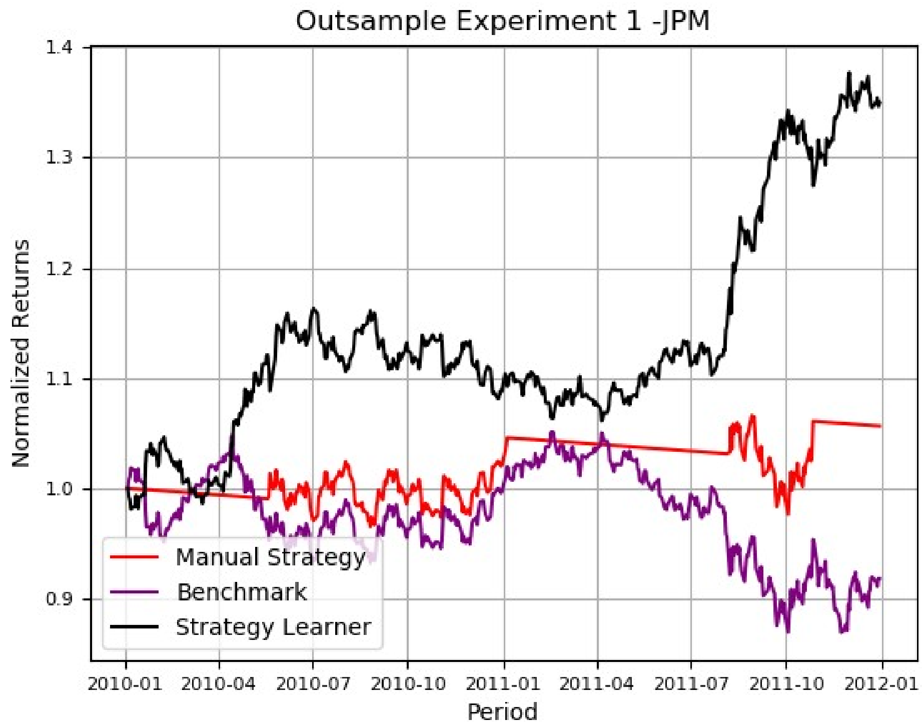
Initial Hypothesis: The strategy learner would out perform manual learner, the main reason behind this was that when developing are strategy learner we were able generate buy/sell or do nothing signal based on the price of the JPM stock, to create the training dataset for RTLearner. Where in the manual learner we had to use the indicator to develop the strategy, hence the strategy is very dependent on the type of indicators used.

Assumptions:

One of the main assumption made in this project is that the past historical data of the stock would have a similar representation as in the future. Which is not 100% true. Another assumption made was that our manual strategy was developed based on the JPM stock, and might not be 100% valid for other stocks.

Results for JPM Stock

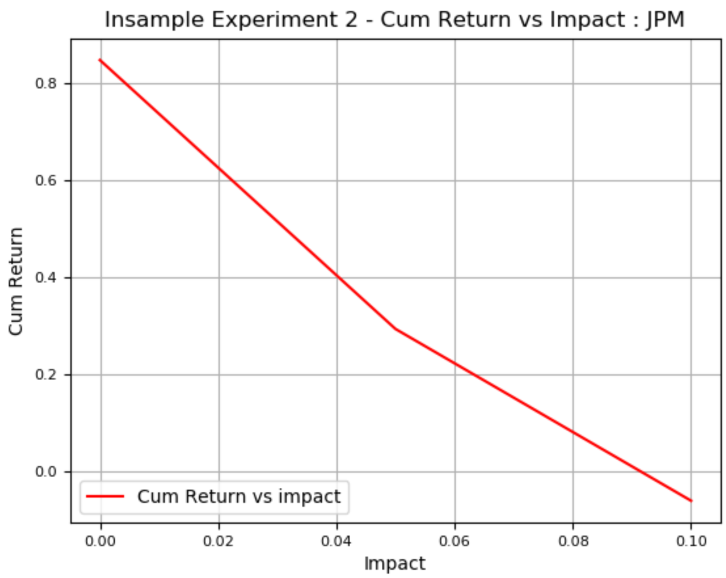


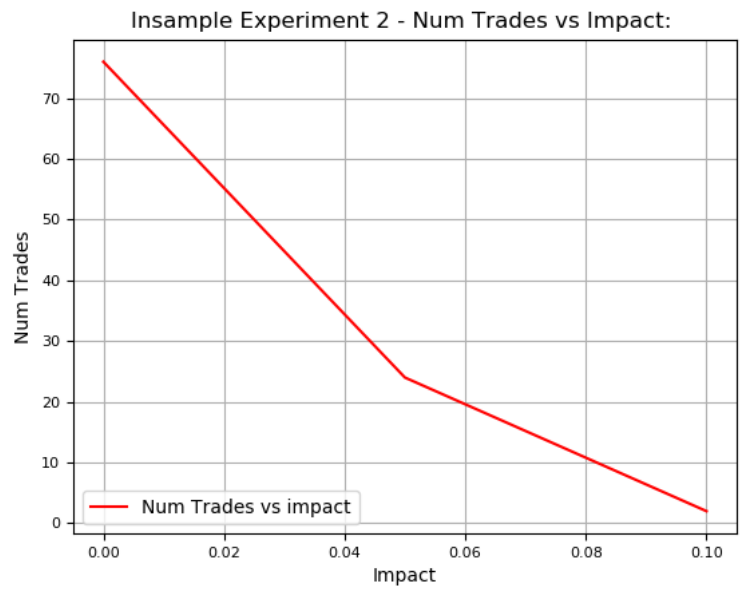


As we can observe from the above in-sample and out-sample plots, the strategy learner out performs the manual and benchmark. This does make sense for the following 2 reasons, one that the training of the RTLearner was more robust than the manual strategy, meaning that the data used was actual based on the future price in the in-sample period. Second reason that the manual strategy could have been improved if other indicators were used that reflected the price changes. The in-sample results would be very similar very time , just based on the criteria used for training, looking ahead at the price of the stock and then training it would always out perform out of sample and manual learner. Another reason is that when using the price to generate buy/sell or do nothing for the indicators, you are actually using the most optimized method, as price is what determines when to buy/sell or do nothing.

Experiment 2:

In experiment 2, the objective was to understand how impact would effect the performance of the learners. The initial hypothesis made was increasing the impact would have a negative impact of the portfolio value. The 2 metrics used to validate our hypothesis are, cumulative returns and number of trades. See the plots below





As since in the plots cumulative return and number of trader both decrease when impact is increased. In this case the values of impact used were 0.0, 0.05 and 0.1.

To reproduce these results, strategy learner was used in this case RT Learner with bag learner as described above. Each value of impact was used to calculate the df trades for the stock, which was used to calculated cumulative return. Number of trades is essentially the count od df_trades dataframe where ever there was a buy/sell signal.

Conclusion:

This was an amazing project, that helped us understand and learn the value of machine learning in trading and to validate the results and performance. This is definitely a drop in the ocean and would help us develop better and more robust strategies in the future.