Project 8:
CS7646

Perry Francois-Edwards
pdfe3@gatech.edu

1 INTRODUCTION

This Project introduces stock price indicators, machine learning algorithms and a market simulator to produce normalized portfolio returns from the JPM security and it also looks at outside factors that can affect results. The project has three parts: 1) Build a Manual Strategy, using at least 3 indicators, to outperform the benchmark. 2) Build a Strategy Learner, using the same indicator code and a model, which in this case is a Classification Learner, to outperform and compare to the Benchmark and Manual Strategy. 3) Use two metrics to determine the effects of impact on the Strategy Learner. I proclaim that the Strategy Learner will produce results better than the benchmark and Manual Strategy and the Manual Strategy will produce results only better than the benchmark. The Strategies should be able to work well with all stock market securities as well. Also note a seed was used for the Random Tree Learner to produce consistent graphs for the report.

2 INDICATOR OVERVIEW

The three indicators used are Bollinger Bands Percent (BBP), Percentage Price Indicator (PPI) and Commodity Channel Index (CCI). The implementation of the indicators to the Manual Strategy and Strategy Learner I used was quite simple. I added all the indicators to produce one vector which provides a strong indication of market predictions that will take place based on the strengths of the indicators themselves. This combined vector was then sent to the Manual Strategy and the Strategy Learner to optimize based on the Learners themselves.

The Manual Strategy and Strategy Learner did not individually optimize the indicators lookback windows in the vector. When doing calculations, it was felt that they were already optimized to their maximum benefit in Project 6.

2.1 Combined Indicator

The implementation of the combined vector was to plainly add them together. The thought process is that this indicator would move accurately and fluidly along the stock market while being able to accurately predict results.

2.2 Manual Strategy

The Manual Strategy uses the PPI initially to determine upward or downward trends as a momentum indicator. It then set-up the perfect indications of trends and depending on the trend, the overall combined indicator vector was implemented. The combined indicator vector then utilized a combined metric for different entry and exit points to determine when to BUY and SELL.

2.3 Strategy Learner

The Strategy Learner worked in a similar manner, and described in more detail in its section, but it utilized the combined indicator vector as the dependent, X, dataset in its model prediction. It set the basis for building a trades dataframe as well based on the predicted Y.

3 MANUAL STRATEGY

The Manual Strategy implemented a combined indicator vector and made assumptions on when to trade based on the combined vector and momentum.

3.1 Describe

For this strategy, as discussed previously, the PPI will be used and as a momentum indicator it is used to determine the upward or downward trend. It's the initial indicator in the Manual Strategy. After the trend was determined, the combined indicator vector was used.

When the combined indicator vector went below -100, which is the combined bottom indication for the BBP and CCI, I would set BUY markers. When the combined indicator vector goes above 200, which is the combined top indication for BBP and CCI, I would set a SELL marker. This is the basis and there are additional markers for the opposite when in the same upward or downward trend. For the manual strategy, this formula is effective to produce accurate results, but it does have some flaws when the indicators are erratic.

I believe this is an effective strategy for an in-sample dataset, but I would want to add more than five indicators to the strategy.

3.2 Compare

The figures below show the comparison of the Benchmark versus the Manual Strategy for both in-sample and out-of-sample data.

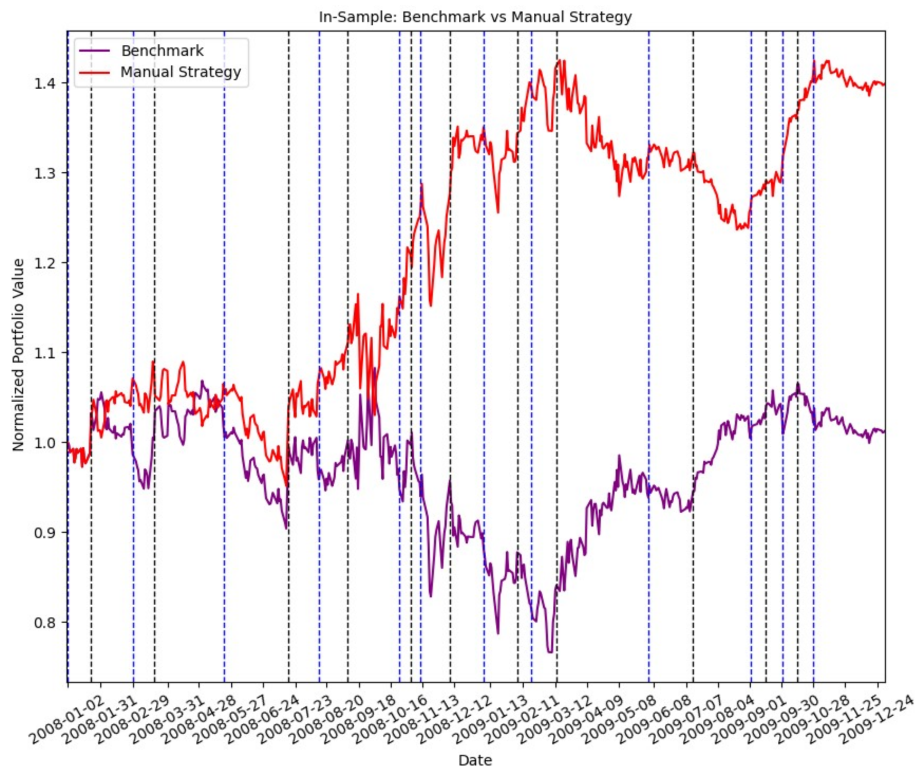

Figure 1—The In-Sample: Benchmark vs Manual Strategy. The blue dashed lines indicate LONG entry points and the black lines indicate SHORT entry points.

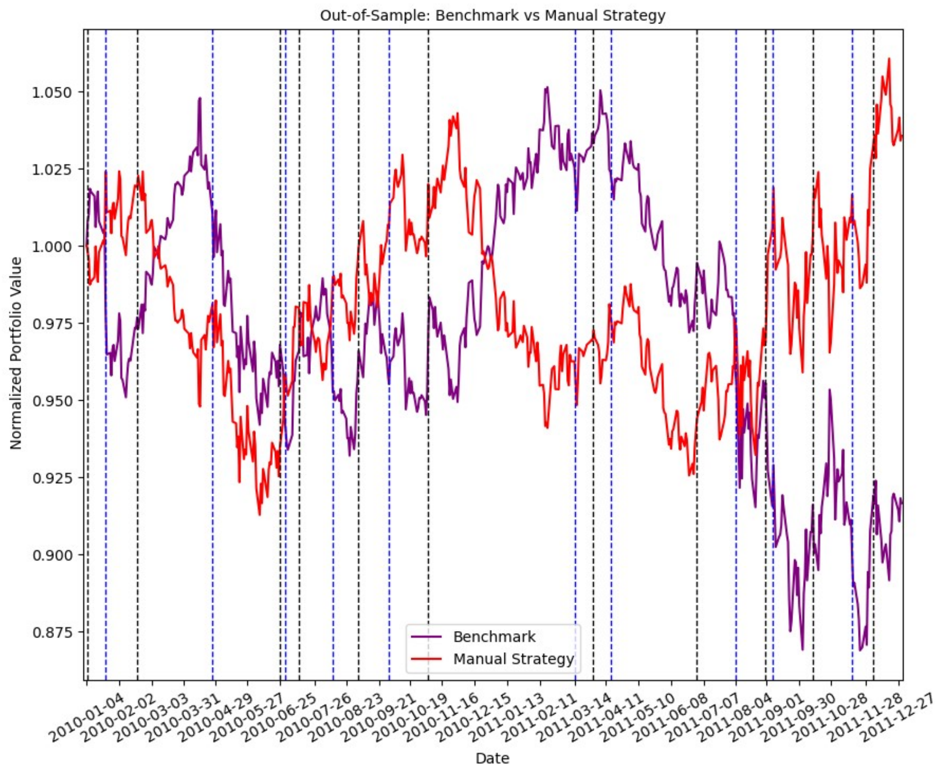

Figure 2—The Out-of-Sample: Benchmark vs Manual Strategy. The blue dashed lines indicate LONG entry points and the black lines indicate SHORT entry points.

As you can see from the figures, the in-sample data provided far greater results than the out-of-sample results. However, both produced higher cumulative returns.

The table below includes the in-sample cumulative returns, standard deviation of the average daily returns and the mean of the average daily returns for the benchmark and Manual Strategy portfolios.

Table 1—The statistics for the Benchmark and TOS data.

Statistic	**Benchmark**	**Manual Strategy**
Cumulative Return	*.012300*	*.398668*
Standard Deviation of daily returns	*.017004*	*.013649*
Mean of daily returns	*.000168*	*.000759*

The table below includes the out-of-sample cumulative returns, standard deviation of the average daily returns and the mean of the average daily returns for the benchmark and Manual Strategy portfolios.

Table 2—The statistics for the Benchmark and TOS data.

Statistic	**Benchmark**	**Manual Strategy**
Cumulative Return	*-.083400*	*.035665*
Standard Deviation of daily returns	*.008481*	*.008123*
Mean of daily returns	*-.000137*	*.000103*

3.3 Evaluate

The out-of-sample Manual Strategy used the same combined indicator vector method as the in-sample Manual Strategy. Based on my final results, the out-of-sample portfolio values were far less than expected and a less robust strategy than what was expected for JPM.

3.4 Explain

I believe these outcomes happened because the strategy is not directly correlated to the out-of-sample learner. When trying to 'tweak' the approach, based on my indicators and methods, one data sample would increase and the other would decrease. It was difficult to find the perfect balance between the two. This also occurs because of the amount of trades that may occur in one dataset versus the other.

4 STRATEGY LEARNER

The Strategy Learner implemented a combined indicator vector while using the Random Forest, Bag Learner and Random Tree (RT) Learner, to predict the final trades.

4.1 Describe Learner

When creating the Strategy Learner, I took the Classification Learner approach of using the Random Forest Learner. The RF Learner was built from a Bag Learner that utilizes the Random Tree (RT) Learner. This method of model learning was chosen so that I could classify my data to predict a trades dataframe that would have a better performance.

Within the Strategy Learner, the Bag Learner was initialized with a leaf size of 5 and 20 bags. When training my Strategy Learner, this is where the combined indicator vector was implemented as well as the classification reference.

The classification section builds the training predicted values of the code and uses an expected return, based on a future day's return, YBUY and YSELL values. In my code, I set these values as the following below:

Table 3—The Classification values for the Strategy Learner.

Name	**N, Lookahead**	**YBUY, Buy Predictor**	**YSELL, Sell predictor**
Classification Values	*10*	*.05*	*-.3*

Based on the market returns and these classifications, the predicted Y values were found. The look ahead value was set to 10 because I wanted to look ahead at shorter time periods to review the market in the short term. I did not want to set predictions too far in the future where the expected return would cause the model to hold a position for longer than needed and produce negative results. I set the YBUY marker low because if there is a potential positive result to take advantage of, the model would buy it. I set the YSELL values in this manner to take advantage of SHORT positions in the market during these time periods. Along with knowing what values I wanted, these values were determined from a tremendous amount of trial and error.

When using this learner and the indicators, the lookback windows remained the same as the Manual Strategy. I felt the indicators were optimized well from Project 6.

I did not standardize my data because I did not see it necessary to provide successive results, but that may have hindered performance in the later experiments.

5 EXPERIMENT 1

Experiment 1 compares the normalized portfolio values of the Benchmark, the Manual Strategy and the Strategy Learner using in-sample and out-of-sample JPM data.

The initial hypothesis is that the Manual Strategy will outperform the Benchmark and the Strategy Learner will outperform the Manual Strategy. The Manual

Strategy and Strategy Learner will also have an applied commission of $9.95 and impact of .005 for each trade.

5.1 Describe, interpret, and summarize

The in-sample normalized portfolio returns proved to follow the expected hypothesis. The highest portfolio return was for the Strategy Learner and the lowest happens to be the benchmark.

I would expect this type of result every time, especially for the Strategy Learner, with in-sample data because the predicted results are based on the actual results and built to maximum profits. So, in-sample is expected to always provide positive results whereas out-of-sample, even though it is built in a similar manner (without learning), can cause a lot more error with the Strategy Learner.

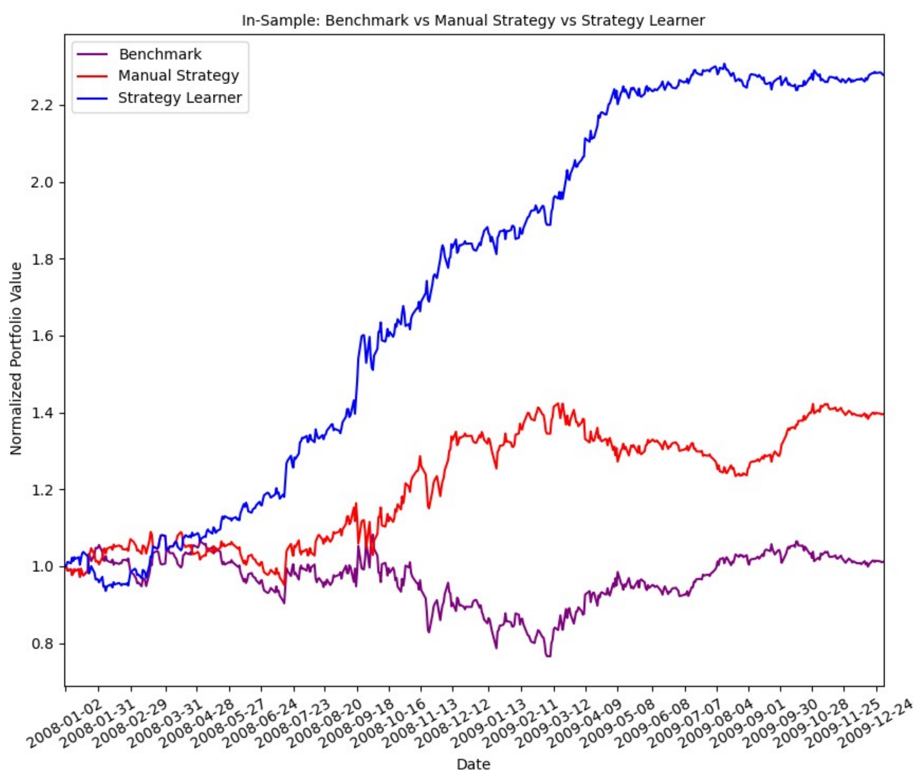

Figure 3—The In-Sample: Benchmark vs Manual Strategy vs Strategy Learner.

The out-of-sample normalized portfolio returns did not prove to follow the expected hypothesis. As shown earlier, the manual strategy performed better than the benchmark, but the strategy learner did not perform better than the benchmark, even though it was close.

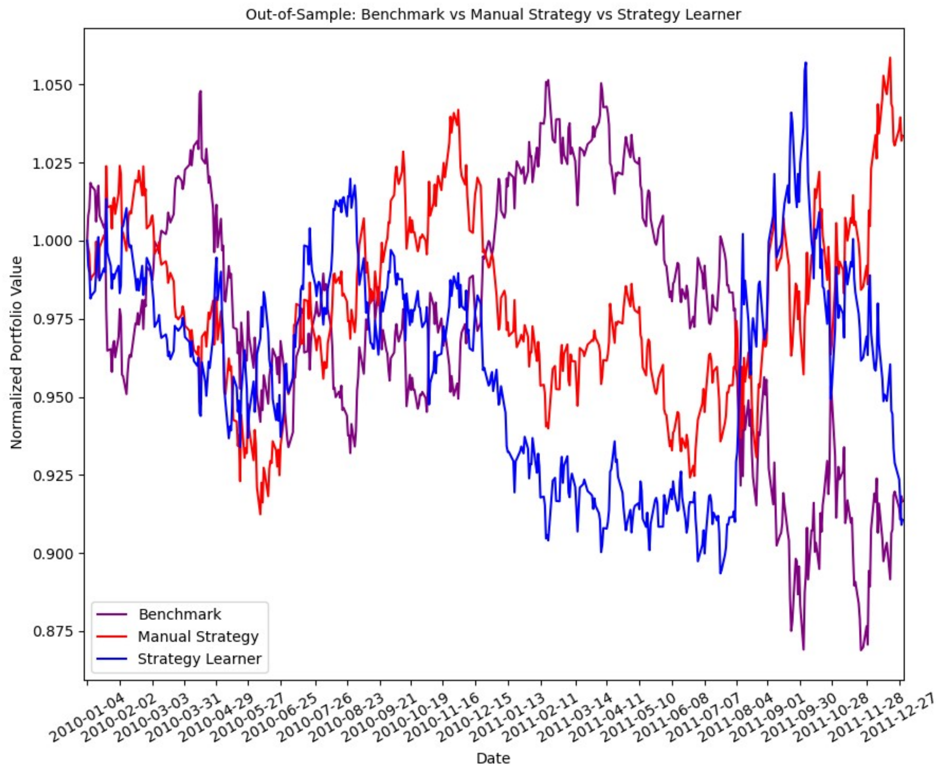

Figure 4—The Out-of-Sample: Benchmark vs Manual Strategy vs Strategy Learner.

6 EXPERIMENT 2

Experiment 2 uses two metrics, 1) Normalized Portfolio Values and 2) Root Mean Square Error (RMSE), to observe the effect of changing the impact value on the Strategy Learner when trading using in-sample data and a $0 commission.

I hypothesize that changing the value of the impact will negatively affect the in-sample results more as the impact value increases. I also hypothesize that the RMSE would increase as the impact value increases. Impact values below.

Table 4—The Classification values for the Strategy Learner.

	Impact_1	**Impact_2**	**Impact_3**	**Impact_4**
Impact Values	*0*	*.0005*	*.005*	*.05*

Based on how often and how much is being traded, these impacts could have a small or big change to the outcomes.

6.1 Results

The image below shows the effect of increasing impacts on the Strategy Learner.

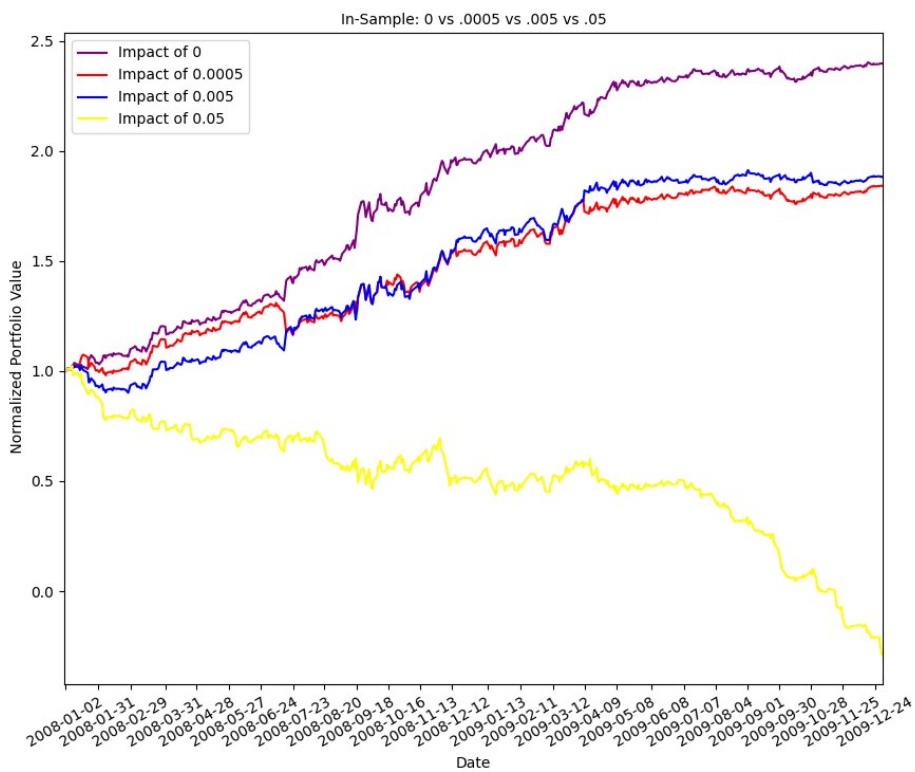

Figure 5—The In-Sample Strategy Learner: Varying Impact Values.

The table below shows the effect of the increasing impacts on the RMSE.

Table 5—The Classification values for the Strategy Learner.

	Impact_1	**Impact_2**	**Impact_3**	**Impact_4**
Impact Values	*0*	*.0005*	*.005*	*.05*
RMSE	*.363498*	*.361120*	*.363835*	*.364217*

These results were slightly expected, but at the same time unexpected. I expected Normalized Portfolio values with varying impacts to be less than the 0 impact.

However, I didn't expect them to vary this much. I thought they would be closer together, but my model must negatively impact them based on the amount of trades. I also did not expect an impact of .005 to produce a higher normalized value than a lower impact. This may suggest my model needs some standardizing.

In regard to the RMSE's, I expected them to be drastically different because of the normalized values being different, but I was proven wrong. Here, it shows all values are starkly similar. It was quite the surprise and gave confidence in the models comparison. It does slightly agree with my hypothesis, except for 'Impact_2'.