Fall 2021 – Project 8: Strategy Evaluation

Michael Lukacsko
mlukacsko03@gatech.edu

***Introduction*—**The purpose of this project is to implement and evaluate two different trading strategies and compare their performance versus each other as well as a benchmark strategy. The first strategy, a manual strategy, uses trading indicator values to develop trading rules based on manually calculated values. The second strategy, called the strategy learner, uses the same trading indicator values and rules as the manual strategy, however, it uses an ensemble learner to perform trades. The ensemble learner uses a bag learner consisting of 20 random tree learners to learn and employ the trading rules. Each strategy, using its respective trading rules, creates a trading DataFrame with buy, sell, and hold flags. The resulting DataFrame is then used to calculate cumulative returns, average daily returns, standard deviation of daily returns, sharpe ratio, and the final portfolio value to compare its performance to the other strategies.

1 INDICATOR OVERVIEW

Three trading indicators are used in this project to analyze JP Morgan Chase & Co (JPM) historical stock data over a lookback window of 14 days and create trading rules. The intent of the trading rules is to provide trade signals. These trade signals are then employed in the manual strategy and strategy learner to identify ideal buy, sell, and hold opportunities. The three trading indicators are explained in the following section. Additionally, included is a plot containing visual representation of the indicators used in the manual strategy and strategy learner. The data represents indicator values using JPM's stock price between 1/1/2010 and 12/31/2011.

1.1 Bollinger Bands

Bollinger Bands are a common trading indicator that leverage a stocks simple moving average (SMA) and standard deviation to represent upper and lower bands. Another feature of Bolling Bands is Percent B (%B). %B is a visual representation of the closing price as it relates to the upper and lower bands. In short, if the stock price is equal to the upper band, %B would be equal to 100. Inversely, if the stock price is equal to the lower band, %B would be equal to zero. B% can be calculated as **B% = (Price – Lower band) / (Upper band – Lower band).**

1.2 Simple Moving Average (SMA)

The SMA is calculated using the stock price rolling mean. This is easily calculated as **SMA = (A1 + A2 ... + An) / n** where 'An' is the stock's price at period 'n' and 'n' is the number of periods. For this project the SMA was divided by the comparable price. Hence, the formula used is **PSMA = Price/SMA**. The result is a representation of the SMA as a ratio between 0 and 1.

1.3 Momentum

The measurement of momentum for this project compares the current stock price with the stock price 'n' number of periods ago. Momentum can be calculated as **Momentum = (Price – Price n periods ago)**. The resulting value was subtracted by 1 so that the ratio would be represented on a zero-line with its values falling above and/or below zero.

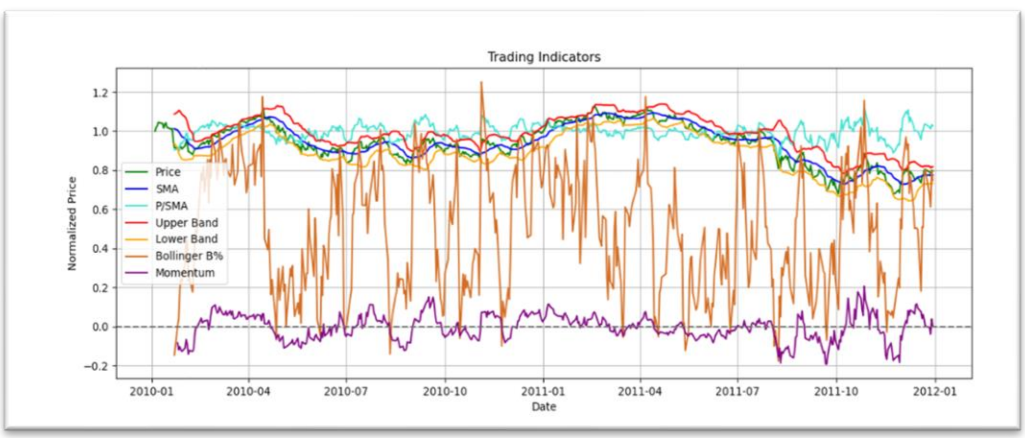

Figure 1 — Indicators used for Manual and Strategy Learners – In sample data.

2 MANUAL STRATEGY

The manual trading strategy in this project uses trading rules, based on the returned indicator values, to flag buy, sell, and hold signals using numerical flags. A flag of 1 signals a buy, -1 signal a sell, and a 0 represents a hold. As a constraint, the manual strategy is limited to holding 1000, 0, or -1000 shares of JPM at any time and has a starting balance of $100,000.

2.1 Manual Strategy Implementation

As previously described, the three indicators above are combined to create trading rules based on their respective value. The trading rules consider whether the current position is in a long, short, or out position to buy, sell, or hold shares. This is indicated by 1, -1, or 0 in the code. If a long position is identified and recorded as a 1, a signal to buy 1000 shares would be signaled by our rules. If, while in a long position, the rules identify a sell, a sell order for 2000 shares would be issued. The use of each indicator to identify trade signals is described below.

1. ***Bollinger Band %B Indicator***

   The %B indicator used to create trading rules looks for a B% of .8 and .2 to signal selling and buying opportunities. I used these values to signal an uptrend beginning when above .80 and a downtrend when below .20. As such, when an uptrend is identified, a sell order is issued. When a downtrend is identified, a buy order is issued.

2. ***Simple Moving Average (SMA) Indicator***

   The Price/SMA indicator used to create trading rules assumes that if the price of the JPM moves substantially from the SMA, a buying or selling opportunity is present. Using the Price/SMA ratio, a value above 1.0 signals a selling opportunity because the stock price will most likely trend down toward the SMA. If the Price/SMA is less than 0.5, a signal to buy is issued. This is because the stock price would be below the SMA and will most likely be trending upward

3. Momentum

The momentum indicator used to create trading rules looks at whether the rate of change calculated by this indicator is positive or negative. A positive value for momentum is indicative of a buying opportunity and a negative value of momentum indicates a signal to sell. A value of -0.2 and 0.2 is used to flag selling or buying signals.

My belief is that the indicators and values outlined above create effective trading rules that, when used in the manual strategy, produce better returns on in-sample data when compared to the benchmark strategy. Figure 2 below proves this to be the case.

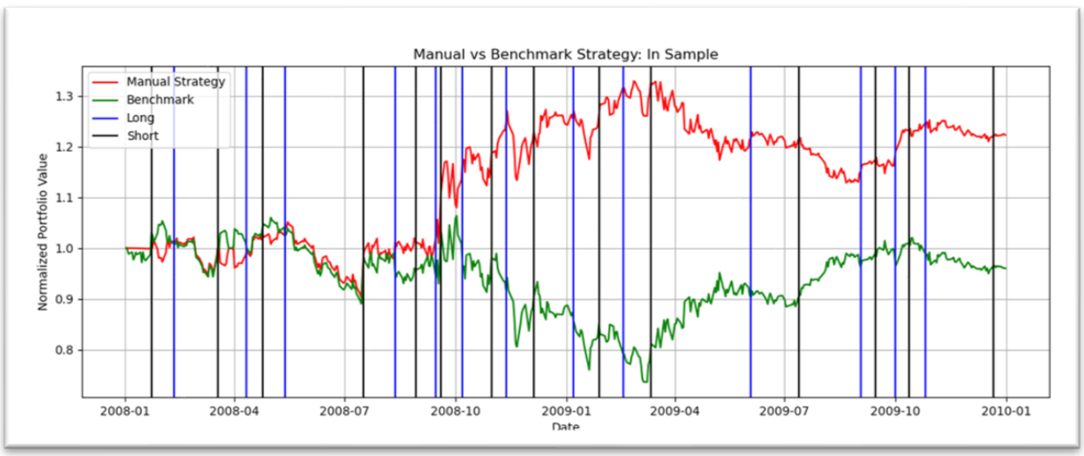

Figure 2—In sample trading results for Manual vs Benchmark trading strategies.

2.2 Manual Strategy vs Benchmark Comparison

Table 1 — In sample and out of sample Manual Strategy vs Benchmark returns.

Dates	Strategy	Cumulative Ret.	Avg. Daily Ret.	Standard Deviation	Sharpe Ratio	Final Portfolio Value
In-Sample	Manual	0.22268	0.00049	0.01385	0.56661	$122,255.75
	Benchmark	-0.03991	0.00007	0.01747	0.06450	$95,814.40
Out-Of-Sample	Manual	0.03860	0.00011	0.00804	0.21236	$103,849.35
	Benchmark	-0.13537	-0.00025	0.00878	-0.45302	$86,278.20

The above chart shows how much more efficient the manual strategy is when compared to the benchmark strategy. More specifically, the manual strategy had a normalized return above 1.0 while the benchmark strategy was below 1.0 for both in sample and out of sample trading data. This means that the manual strategy increased in value while the benchmark strategy decreased in value for both periods. Final portfolio value for in-sample data netted a return on investment (ROI) of over 22% while the ROI of the benchmark strategy over the same time netted a ROI of about -4%. The same can be seen looking at the out of sample data. The manual strategy netted a ROI of 3.8% and the benchmark strategy netted a ROI of -13.5%.

2.3 Manual Strategy Evaluation

Using figure 3 below as a reference, there are periods in the out of sample data where the benchmark strategy performed better than the manual strategy implementation. Nevertheless, the manual strategies cumulative return, average daily return, sharpe ratio, and final portfolio values are all better when compared to the corresponding values of the benchmark strategy.

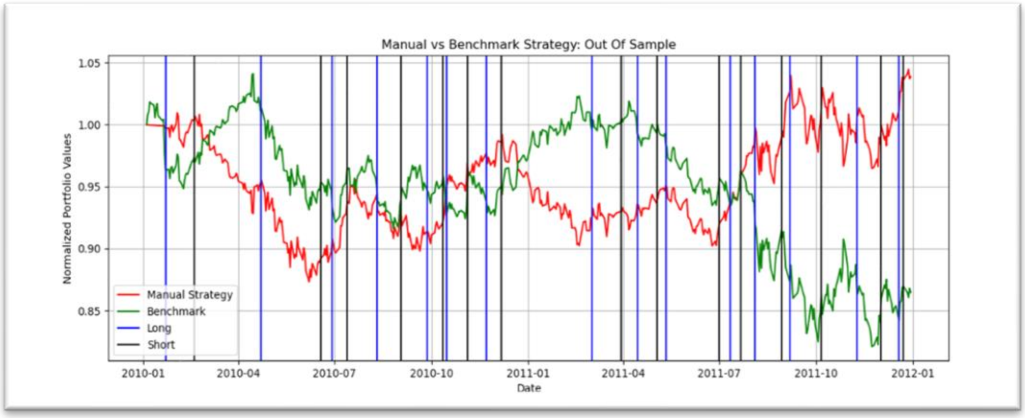

Figure 3—Out of sample trading results for Manual vs Benchmark Strategy

3 STRATEGY LEARNER

3.1 Strategy Learner Implementation

To frame this trading problem as a learning problem, an ensemble learner was implemented using bagging. To accomplish this, a Bag Learner was employed with a bag size of 20 and leaf size of 1. The Bag Learner uses 20 random tree learners (RTLearners) for this problem. To train the learners, the following strategy is followed:

1. The StrategyLearner stores the JPM stock price in a DataFrame
2. Using the JPM normalized stock price, the three trading indicators (Price/SMA, Bollinger Band B%, and Momentum) are calculated and stored in another DataFrame to be used as the X-training data.
3. Y-training data is created using the 10 day future normalized stock price to flag a buy, sell, or hold using 1, -1, and 0 respectively.
4. The add_evidence() method is called, and the learner learns the trading strategy. As such, the learner can then forecast future price changes and make decisions based on its learnings.

Once the learner is trained, it is tested using the following strategy:

1. The JPM stock price and indicator values are stored as X-testing data.
2. A query() method is called by the StrategyLearner to query the ensemble learner and return a DataFrame that will be used in turn to create a

trades_df DataFrame that will be populated with the buy, sell, and hold flags equal to 1, -1, or 0.

3. Because of the constraint that only 1000, 0, or -1000 shares can be possessed at any time, I use the following logic to trade up to 1000 or 2000 shares at a time:
   a. If and out position is identified by a flag equal to 0, this means a purchase of 1000 shares is valid.
   b. If the flag returned is a 1 or -1, a long or short position has been identified and this means a buy/sell order of 2000 or -2000 shares is valid.

The only data adjusted in this problem is the normalization of JPMs stock price to start at 1.0. While this is not a required adjustment of data, it is considered best practice and makes analyzing results easier to perform.

4 EXPERIMENT 1: MANUAL STRATEGY / STRATEGY LEARNER

In experiment 1, the benchmark strategy, manual strategy, and strategy learner are implemented to determine which performs the best using JPM's in sample historical data. The same trading constraints outlined above are used here, however, a trading commission of $9.95 is used along with an impact of 0.005. Additionally, the same indicators and trading rules are used by both the manual strategy and strategy learner to determine the best opportunity to buy, sell, or hold. My belief is that the strategy learner outlined above will perform better than the manual strategy, and the manual strategy will perform better than the benchmark strategy. I believe this will be the case because the manual strategy will rely only on the three indicators while the strategy learner will use ensemble learning with the three learners to deliver results with less overfitting and, as a result, better performance. To determine which strategy performs the best, I call a compute_stats() method to calculate and return the cumulative return, average daily return, standard deviation, and sharpe ratio. Also used to decide the most optimal strategy is the final portfolio value which is calculated by taking the value of the last element in the df_port_val DataFrame returned by compute_portvals().

4.1 Experiment 1 Parameters

In sample historical trading data for JPM 1/1/2008 – 12/31/2009, commission = $9.95, impact = 0.005. Starting cash = $100,000 and allowable positions = 1000 shares long, 1000 shares short, 0 shares. Only orders to buy/sell is allowed and there is no limit on leverage.

4.2 Experiment 1 Results

Figure 4—Benchmark Strategy vs Manual Strategy vs Strategy Learner

Table 2 — In sample results of Benchmark vs Manual Strategy vs Strategy Learner trades

Dates	Strategy	Cumulative Ret.	Avg. Daily Ret.	Standard Deviation	Sharpe Ratio
In-Sample	Benchmark	-0.039913746	0.000071	0.017468	0.064496
	Manual Strategy	0.222679157	0.000494	0.013852	0.566609
	Strategy Learner	1.166549931	0.001599	0.011385	2.230368

Reviewing table 2 above, the cumulative return of each strategy makes clear that the benchmark performed the worst, the manual strategy better than the benchmark, and the strategy learner better than the manual strategy. Also notable is each strategy sharpe ratio, or return adjusted for the risk. The strategy learner has a sharpe ratio over 2 which is considered excellent, while the manual strategy has a sharpe ration between 0.5 and 1 which is just good. The benchmark strategies sharpe ratio under .1 is not good. While these results and the visualization in figure 4 above show that this experiments outcome proves my hypothesis to be true, it should be mentioned that these results might not occur 100% of the time. This is because the RTLearners random tree uses randomization when selecting the best feature to split on. As a result, it would be possible for the manual strategy to perform better than the strategy learner, however, I would expect such an occurrence to be rare.

5 EXPERIMENT 2: STRATEGY LEARNER

Experiment 2 looks at the effects varying impact values have on the strategy learner using in sample data. Because impact is directly associated with the cost a trader incurs based on buying and selling assets, it is my expectation that the strategy learners will perform worse with higher impact values.

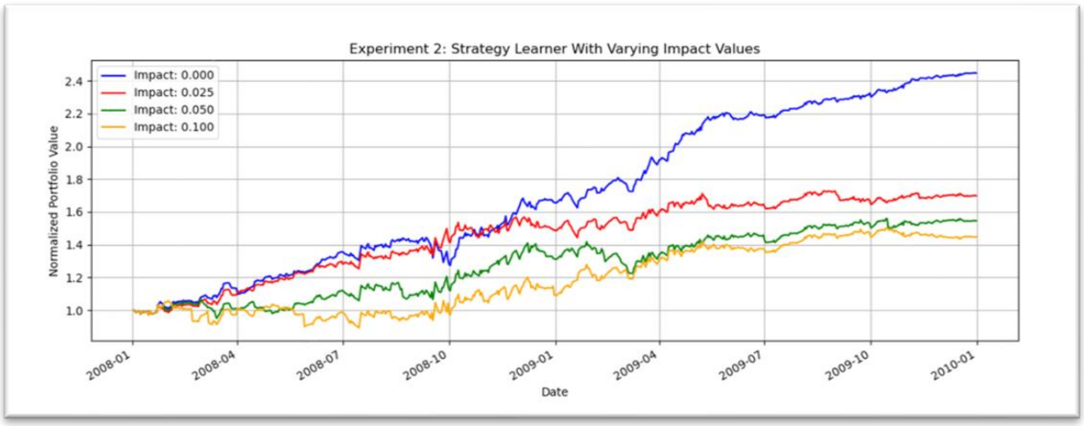

Figure 5— Experiment 2 results using Strategy Learner and varying values of impact.

Table 3 —Portfolio stats from Experiment 2 using Strategy Learner and varying values of impact

Dates	Impact	Cumulative Ret.	Avg. Daily Ret.	Standard Deviation	Sharpe Ratio	Total Trades
In-Sample	0.000	1.85430	0.00213	0.00954	3.54069	74
	0.025	0.68234	0.00112	0.01298	1.36563	32
	0.050	0.48104	0.00089	0.01484	0.95128	26
	0.100	0.49016	0.00093	0.01653	0.89200	11

5.1 Experiment 2 Parameters:

In sample historical trading data for JPM 1/1/2008 – 12/31/2009, commission = $0.00, impact = 0.000, 0.025, 0.050, 0.100. Starting cash = $100,000 and allowable positions = 1000 shares long, 1000 shares short, 0 shares. Only orders to buy/sell is allowed and there is no limit on leverage.

5.2 Experiment 2 Results

The visual results in figure 5 and portfolio statistics in table 3 above make clear two points:

1. There is a correlation between the value of the impact and how the strategy learner performs. This observation validates the hypothesis that higher impact values will cause the strategy learners to perform worse. It is clear to see, based on table 3, that the cumulative return decreases as the value of impact increases. Also notable, but not necessarily unexpected, is that the sharpe ratio decreases as well. The strategy learner with the impact value of 0.000 has a sharpe ratio above 3.5 which is very, very good. As the impact values increase, cumulative return, sharpe ratio, and the other portfolio statics drop as well. One observation I would follow up on and test in more detail is the comparison of results from the strategy learner with impact values of 0.050 and 0.100. These two strategy learners performed similarly, but surprisingly the higher of the two impacts performed better. This is even though the strategy learner with an impact value of 0.100 had more than half as few trades overall.
2. There is a correlation between the number of trades our strategy learner will make and the value of the impact. This was not something I had considered as part of my hypothesis, but it makes sense. As impact increases, the strategy learner will not trade as often because a lower cumulative return for each trade should be expected. The return per trade is accounted for in the add_evidence() method and, as a result, less trades should be expected as the return per trade decreases.