Assignment 8 - Strategy Evaluation

Vasu Patel
vpatel436@gatech.edu

***Abstract*—** This project involves creating a manual strategy using at least 3 technical analysis indicators (Bollinger band percentage, simple moving average ratio, and moving average convergence/divergence) from the indicator evaluation project, and a classification based strategy learner using the bag learner and random tree learner classes from the assess learners project to compare the performance of each learner when given the stock prices for JPM from a set time range.

1 Indicators

1.1 Bollinger Band Percentage

The Bollinger Band Percentage indicator derives from Bollinger Bands, an indicator which is used to measure volatility in stock prices by using two band lines representing the simple moving average plus two standard deviations of the price, and the simple moving average minus two standard deviations of the price. The Bollinger Band Percentage is calculated using the upper and lower bands, which measures how overbought or oversold a stock is on a particular day.

%B = (Price - Lower Band)/(Upper Band - Lower Band)

A reading of 1.0 and above indicates the price is above the upper band, while a reading of 0 and below indicates the price is below the lower band. In the manual strategy, a buying signal is indicated when the BBP is less than 0.2, while a selling signal is indicated when the BBP is greater than 0.8. These values are chosen to make sure the price is close enough to the upper or lower band value to make it safer to trade the stock since it is likely that the future price will decrease or increase respectively.

1.2 Price to Simple Moving Average Ratio

The Simple Moving Average(SMA) indicator is used to measure the average price of a stock over a specified period of time, which is calculated by specifying a window size and then computing the mean of the stock prices within each window frame using the *.rolling()* function.

$$SMA = \frac{A_1 + A_2 + ... + A_n}{n}$$

The SMA tells traders the direction of the market trend, while the price to SMA ratio, which is calculated by dividing the stock price dataframe by the SMA, tells traders if a stock is likely overvalued if the ratio is usually above 1, and likely undervalued if the ratio is below 1. In the manual strategy, the price to SMA ratio indicates a buy signal when the ratio is less than 0.975, while a ratio above 1.025 indicates a sell signal. These thresholds are chosen to make sure that the price trend is moving far enough up or down where trading is more safe and profitable.

1.3 Moving Average Convergence/Divergence

The Moving Average Convergence/Divergence(MACD) is a momentum indicator that uses the Exponential Moving Average(EMA), an indicator that tries to determine price trends based on recent stock prices, to determine buy and sell signals by showing the relationship between two EMA lines of a stock in comparison with a signal line. The MACD values are calculated by using two EMA calculations, with one using a short window period of 12 and the other using a long window period of 26. The long and short EMA values are calculated using the *.ewm()* function on the JPM stock price dataframe with the specified window size as the parameter in the function, and then taking the mean of the functions to get the resulting long and short EMA values. Afterwards, the MACD values are retrieved by subtracted the long EMA value dataframe from the short EMA dataframe. The MACD signal line is calculated by using the *.ewm()* function on the calculated MACD dataframe, but with a specified window size of 9, and then taking the mean of the result. A buy signal is indicated when the current day MACD value is greater than the current MACD signal value plus 0.05, and a sell signal is indicated when the MACD value is less than the MACD signal minus 0.05. The 0.05 is accounted into the MACD signal value to make sure that the MACD line is trending well above or below the signal line and not making an abrupt change in the opposite direction after crossing it, which may lead to unprofitable trading decisions.

2 Manual Strategy

2.1 Trading Rules

The manual strategy takes into consideration two technical indicator rule requirements in order to determine whether to buy or sell stock. The main indicator for making a decision on a particular day is the current price to SMA ratio since it determines how a stock is trending. However, this alone does not fully determine if it is a good choice to buy or sell stock on the said day since the trend may abruptly change, and therefore is combined with the BBP and MACD rules to make a better prediction of the future trend of the stock, creating an effective strategy to increase portfolio value. The SMA ratio value is first checked to determine whether the stock price is trending upwards or downwards, and then the BBP value along with the MACD and MACD signal value rules are checked. If the BBP or MACD rule is satisfied, a signal is created for trading, otherwise the action would be to do nothing for that day. If the SMA ratio value is less than 0.975 and either the BBP value is less than 0.2 or the MACD value is greater than 0.05 plus the MACD signal value, the trading signal will be equal to 1, meaning a buy signal is generated. On the other hand, if the SMA value is greater than 1.025 and either the BBP value is greater than 0.8 or the MACD value is less than the MACD signal value minus 0.05, the trade signal will be -1, indicating a sell signal. Since the valid stock holding positions are 1000, 0, and -1000, if a buy signal is generated, 1000 shares will be bought if the current holdings is 0, 2000 shares if the holdings are -1000, and 0 if the holdings are 1000 shares. If the sell signal is generated instead, 1000 shares will be sold if the current holding is 0, 2000 if the holdings are 1000, and 0 if the holdings are -1000.

2.2 Manual Strategy vs Benchmark Strategy Comparison

In both the manual and benchmark strategies, the starting value of the portfolio is $100,000, where both strategies focus on trading JPM stock for both the in-sample and out-of-sample periods. The benchmark strategy simply consists of buying 1000 JPM stocks on the first day of the time range, and holding that amount until the end of the time period, whereas the manual strategy makes trades based on the rules developed using the technical indicators described above. In each figure, the vertical blue lines represent long entry points for the manual strategy, while the vertical black lines represent short entry points.

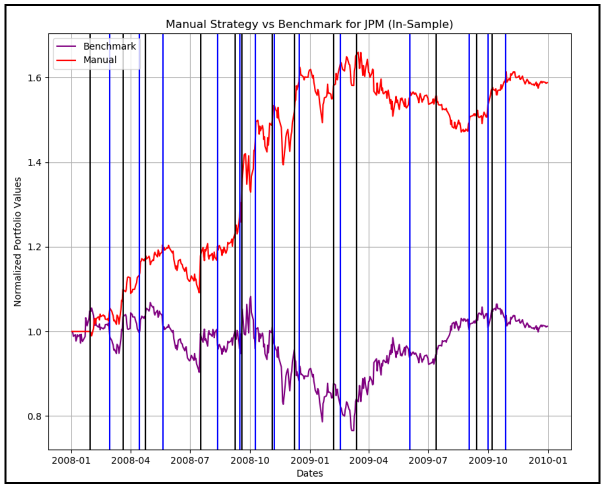

Figure 1— Manual vs Benchmark Strategy portfolio values for in-sample period (2008, 1, 1) - (2009, 12, 31).

The figure above shows the comparison of the performance between the manual strategy and the benchmark strategy with regards to portfolio value over the in-sample time range of January 1, 2008 through December 31, 2009. As seen in the figure, the manual strategy performs a lot better at maximizing portfolio value than the benchmark strategy throughout the in-sample time range, with the ending portfolio value of the manual strategy being $158,820.35, and the ending portfolio of the benchmark strategy being $101,027.70. The manual strategy portfolio value is shown to be consistently higher and trending up for the most part, while the benchmark portfolio value seems to rise up and down, until it starts trending mostly downwards until it picks back up around April of 2009, but barely rises higher than its starting portfolio value. The table below shows the cumulative return, average daily return, and standard deviation of daily return for both strategies.

Portfolio Statistic	Manual	Benchmark
Cumulative Return	0.588204	0.012325
Average Daily Return	0.000982	0.000169
STD of Daily Return	0.011280	0.017041

Overall, the manual strategy yields a higher cumulative return, average daily return, and standard deviation of the daily return than the benchmark strategy, showing that it is a much better strategy to use over the other for the in-sample period.

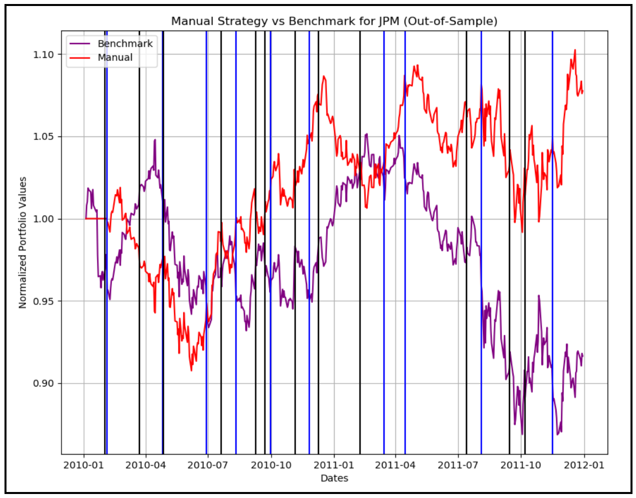

Figure 2— Manual vs Benchmark Strategy portfolio values for out-of-sample period (2010, 1, 1) - (2011, 12, 31).

Figure 2 as shown above shows the comparison of the manual strategy against the benchmark strategy with regards to portfolio value, but this time using the out-of-sample time period of January 1, 2010 through December 31, 2011. Within this time frame, the manual strategy portfolio does not always yield a higher value than the benchmark portfolio value. At around March 5, 2010, the benchmark portfolio value, which is $99,755.70, surpassed the manual portfolio value of $99,091.70 and continued to do so until around August of 2010, where the manual strategy started to beat the benchmark strategy. The benchmark strategy beats the manual strategy again around February 10, 2011, but is short lived since the manual strategy regains a higher portfolio value after March 4, 2011 and continues to do so until the end of the time period. The ending portfolio values for the manual and benchmark strategies are $107,759.80 and $91,445.70 respectively. Although the manual strategy does better than the benchmark strategy again, it did not perform as well as it did during the in-sample period. The technical indicator rules may not have predicted future buy and sell signals as well as it did for the in-sample period, thus making trades that negatively impacted the portfolio value during some points within the out-of-sample time range. As a result, although it beat the benchmark strategy, the manual strategy does not significantly raise the portfolio value from its starting value as it did during the in-sample period. The table below shows the portfolio statistics for the out-of-sample period for both strategies.

Portfolio Statistic	Manual	Benchmark
Cumulative Return	0.077598	-0.083579
Average Daily Return	0.000177	-0.000137
STD of Daily Return	0.007581	0.008500

The manual strategy again outperforms the benchmark strategy in terms of cumulative return and average daily return, but has a slightly lower standard deviation of the daily return. As seen in figure 2, since the benchmark strategy portfolio value has been mostly decreasing throughout the out-of-sample time period and ends up being lower than its starting value at the end, it explains why its cumulative and average daily returns are negative rather than positive.

3. Strategy Learner

The strategy learner built is a classification based learner, meaning it uses a random tree based bag learner to predict values given a set of training data to learn from. The leaf size and bag size given to the bag learner for this strategy are size 5, and 20 bags. In the *add_evidence()* function, the training data needs to be created in order to train the bag learner to predict signal values later on. In order to construct the x training data, the Bollinger band percentage, simple moving average, MACD, and MACD signal values are calculated for the time range specified in the *add_evidence()* parameters. The technical indicator values are then converted into their own separate dataframes by using the *pd.DataFrame()* function in order to store all the values afterwards as a single variable using the *pd.concat()* function. The columns are named with their respective indicator name, and the *.replace(np.nan, 0)* function is used to replace nan values with 0 in any column that has such values. The daily returns are calculated next using the JPM stock prices while taking into account the lookback days. After this calculation, experimentation was done to determine a YBUY and YSELL value that works best for the strategy learner, which is determined to be 0.035 and -0.035 respectively while accounting for the specified impact value as well. With the YBUY and YSELL values determined, the y data for each day is calculated by comparing the daily return values with the YBUY and YSELL values. If the return is greater than YBUY, the y column for that day would equal 1, meaning a buy signal is generated. However, if the return is less than YSELL, the y value will be -1, indicating a sell signal. Otherwise, if none of the previous cases are true, the y value will simply be 0, meaning no action will be taken. With the x and y data being generated, the bag learner's *.add_evidence()* function is called to add the said generated x and y data for the learner to learn. Since the training data has been fed to the learner, the test data is now generated in the strategy learner's *testPolicy()* function in the same manner as in the *add_evidence()* function. This time, the y values are now generated through the bag learner by calling its *.query()* function with the x test data as the parameter. With the y values now generated, they are looped through and compared to see if each value is greater than 0 which indicates a signal of 1 or less than 0 which indicates a signal of -1, otherwise the signal will be 0. The signal value along with the share value is then checked to see what type of trade will be executed, along with how many shares will be traded on that particular day based on the allowable holdings of 1000, 0, and -1000, similarly to the manual strategy. Due to using a classification based strategy learner, there was no need for data adjustment since the dependent data in this case are the stock prices, which is used to calculate all the indicator values to use as the x data.

4. Experiment 1

This experiment involves comparing the portfolio values of the benchmark, manual, and strategy learners over the in-sample (January 1, 2008 - December 31, 2009) and out-of-sample (January 1, 2010 - December 31, 2011) time periods. All three learners have a starting value of $100,000, and trade JPM stock for the experiment. The strategy learner has a commision and impact of 0. Additionally, when computing portfolio values, the commission and impact will be 9.95 and 0.005 respectively. Since the trading rules are tweakable to give the best performance for the in-sample time period within the manual strategy, it is expected to give a higher portfolio end value than the benchmark, and the strategy learner is predicted to have a higher value than the other two strategies overall.

The manual and strategy learner constructors are first imported and called in order to start using their methods. The strategy learner needs to call its *.add_evidence()* with the specified date range, starting value, and symbol before calling its test policy. Afterwards, the respective *.testPolicy()* functions from both learners are called to create the trades dataframe. For the benchmark strategy, its trades dataframe is generated by calling the *.benchmark_trades()* function from the manual strategy learner. Once the three trades dataframes are created, the *.compute_portvals()* function is called from the imported *marketsimcode* file to compute each learner's portfolio values using the specified parameters for starting value, impact, and commission. Before plotting each learner's portfolio values, each portfolio is normalized by dividing the portfolio dataframe by the first value within the dataframe. Afterwards, the normalized portfolio values of each learner is plotted using the *matplotlib.pyplot* library.

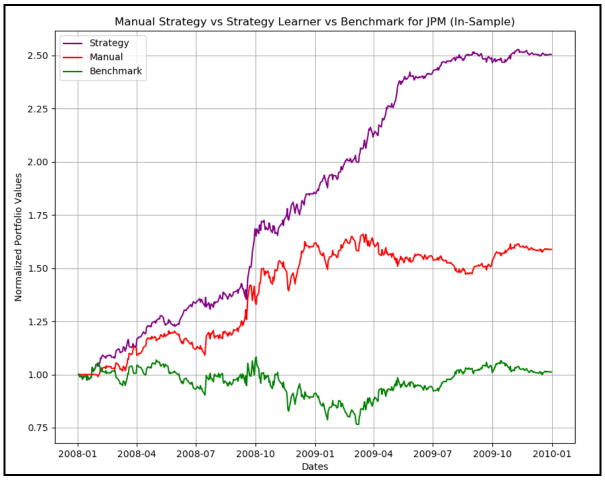

Figure 3 — Manual vs Benchmark vs Strategy Learner portfolio values for in-sample period (2008, 1, 1) - (2009, 12, 31).

Portfolio Statistic (In-Sample)	Manual	Benchmark	Strategy
Cumulative Return	0.588204	0.012325	1.504186
Average Daily Return	0.000982	0.000169	0.001869
STD of Daily Return	0.011280	0.017041	0.009590

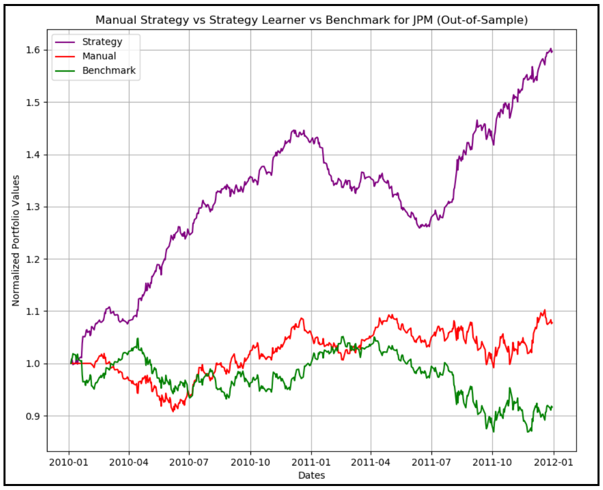

Figure 4 — Manual vs Benchmark vs Strategy Learner portfolio values for out-of-sample period (2010, 1, 1) - (2011, 12, 31).

Portfolio Statistic (Out-of-Sample)	Manual	Benchmark	Strategy
Cumulative Return	0.077598	-0.083579	0.596558
Average Daily Return	0.000177	-0.000137	0.000947
STD of Daily Return	0.007581	0.008500	0.005720

Figures 3 and 4 above show the outcome of the experiment using the in and out sample time periods along a table showing each learner's overall portfolio statistics. Overall, the strategy learner performed exceptionally as predicted, outperforming both the manual and benchmark strategies significantly in the end in terms of ending portfolio value and portfolio statistics in both the in and out sample time periods. It is expected that the strategy learner should outperform the manual and benchmark strategies for in-sample data every time due to making better trading decisions, as reflected by its in-sample performance.

5. Experiment 2

This experiment involves seeing how different impact values affect the performance of the strategy learner using the in-sample time range. The impact values chosen are 0, 0.08, and 0.15. The metrics used to see how the impact value affects performance are portfolio values and number of trades. It is predicted that portfolio value will decrease as well as the number of trades performed when increasing the strategy learner impact.

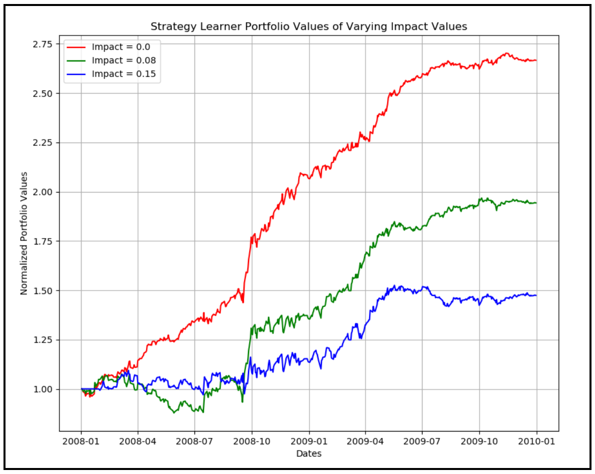

Figure 5 — Strategy Learner portfolio values using 3 impact values

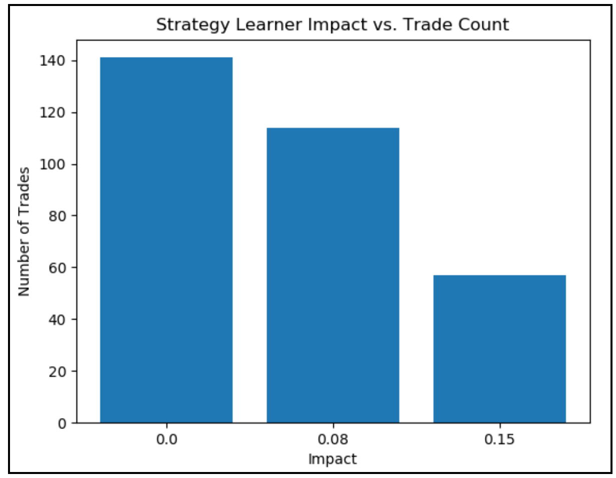

Figure 6 — Number of trades with regards to impact using the Strategy Learner.

In figure 5, the portfolio outcome of three strategy learners using the three impact values stated previously are shown. This experiment is performed by calling 3 strategy learner constructors, each having a separate impact value of 0, 0.08, and 0.15, and 0 commission. Each learner's *.add_evidence()* function is called using the JPM symbol, the in-sample time period, and a starting value of $100,000. The *.testPolicy()* function is then called using the same parameters in the add evidence function. Afterwards, each learner's resulting trades dataframe is used in the *.computer_portvals()* function to retrieve the portfolio values. The portfolio values are then normalized before plotting them. As expected, the lower the impact value is, the higher the ending portfolio value is for the respective strategy learner. It can be noted that the learner with an impact of 0.15 does outperform the learner with the 0.08 impact during the beginning of the time range until around October of 2008, so there may be instances where the higher impact learner may perform better than the lower impact learner at some points. Figure 6 shows the impact of the learners in relation to the number of trades performed. The trade count is retrieved through the created function *trade_count(trades,symbol)*, where each trades dataframe is checked to see if the shares for each day is greater than or less than 0, in which case the count is increased by 1. The impact and trade count for each learner is then plotted with the *plt.bar()* feature. The higher impact learner had the least amount of trades, while the lower impacts showed increasing trade count as predicted.