Project 8: Strategy Evaluation

Peter Dawoud pdawoud3@gatech.edu

Abstract—This paper will combine everything developed this semester to build a machine-learning-based trading system. The paper starts by introducing the technical indicators used in that system and then introduces a manual strategy that aims to show these indicators in action. Finally, the paper presents the ML-based strategy learner and provides experiments to compare it to the manual strategy and to show the impact of trading impact.

1. Introduction

This project will explore combining supervised classification learning using Random Forests with technical indicators to develop a trading strategy. There are four phases in this project. First, the paper will discuss the four technical indicators used by briefly describing how they are implemented and used in both a manual strategy and an ML-based classification strategy learner. In the second phase, the paper will present how the indicators are combined to create a trading signal. Then compare the performance of the manual strategy to a benchmark portfolio of buying and holding 1000 shares of $JPM. In the third phase, the paper will present the strategy learner. In this phase, the paper will describe how the problem is framed, what hyperparameters are used, and whether and how data is discretized. The final phase of this paper consists of two experiments: the first aims to compare the performance of the strategy learner to the manual strategy for both in-sample and out-of-sample test cases.

2. Indicators used

1. Bollinger Bands:
   Bollinger Bands is a technical analysis tool used to identify overbought and oversold conditions in the market. It is implemented by computing a simple moving average over a window of time and the standard deviation over the same window. It then compares the SMA to the Bollinger bands which are equal to an N standard deviations added and subtracted to the SMA. If the price is above the upper band, the security is overbought. While a price below the lower band means that the security is oversold.

The 2 parameters are the size of the window which is 20 for this project and the number of standard deviations which is 2.

In this specific project, %B is used which is calculated by subtracting the lower band from the price and then dividing the result by the difference between the upper and the lower band. This is just to have one column for the indicator.

2. MACD

The Moving Average Convergence Divergence (MACD) is a popular momentum and trend-following indicator. It helps traders identify potential buy and sell signals based on the relationship between two moving averages of a security's price. MACD utilizes the exponential moving average and is calculated by subtracting the MACD line from the signal line. The MACD line is the difference between a fast and a slow EMAs, while the signal line is an EMA over a window of time. When the MACD line is over the Signal line it indicates a sell signal while the opposite indicates a buy signal.

The parameters in question for MACD are the windows for the fast EMA, the slow EMA, and the signal EMA. The values 6, 13, and 5 are used respectively.

3. RSI

$$RS_{step\ one} = \frac{Average\ gain\ for\ the\ first\ window}{Average\ loss\ for\ the\ first\ window}$$

$$RS_{step\ two} = \frac{previous\ average\ gain \times (period\ length - 1) + current\ gain}{previous\ average\ loss \times (period\ length - 1) + current\ loss}$$

$$RSI = 100 - \frac{100}{1 + RS}$$

The Relative Strength Index is one of the types of momentum oscillators used in technical analysis to determine the magnitude of never-ending price changes of a trading instrument. It is computed using the formulas above. An RSI above 70 indicates overbought security, while an RSI below 30 indicates an oversold security. The only parameter in question is the window size which is set to 14 in this project

4. ROC

$$ROC = \frac{Closing\ price_p}{Closing\ price_{p-n}} \times 100$$

The Rate of Change (ROC) is a momentum oscillator used to measure the percentage change in price over a specified period. This is used to determine momentum whether it is positive or negative. The window is set to be 5 days in this project.

3. Manual strategy

Description: The strategy used here is so naïve and so simple yet effective as the evidence will show. The four indicators vote to either short or long the security. Once a row has three votes to either long or short it is considered for initiating a trade. Here is how each indicator votes:

1. Bollinger Bands B%: if the value is more than 1, it votes to buy. While it votes to sell if the value is less than zero
2. MACD: if the MACD – signal line is more than zero it votes to buy. However, if it is below zero it votes to sell.
3. RSI: It votes to buy if it is above 70% and to sell if it is below 30%.
4. ROC: Just like MACD it votes to buy above zero and to sell below zero.

Performance Comparison In-Sample

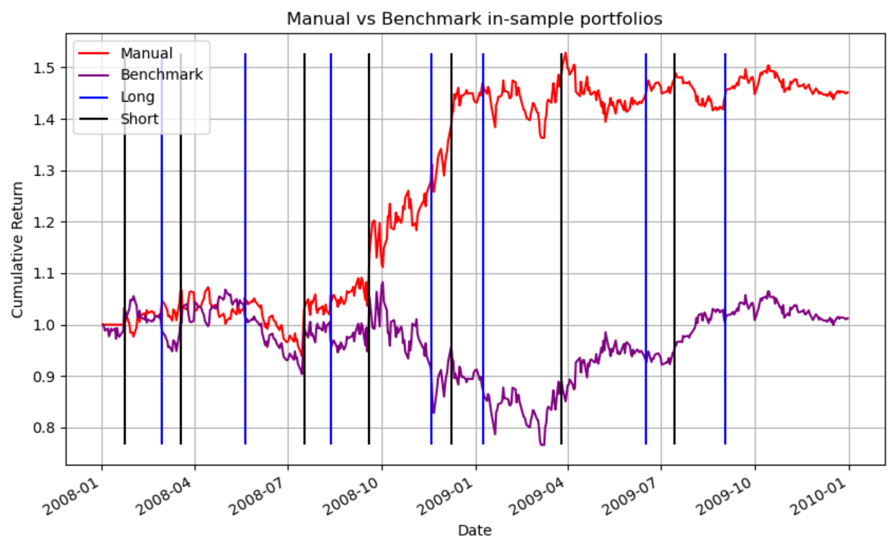

Figure 1

Figure 1 shows that for the in-sample dataset, the manual strategy outperforms the benchmark strategy for more details. Figure 1 also shows the short and long entry points as black and blue vertical lines respectively.

The following table shows that the manual strategy outperforms the benchmark portfolio on every metric.

	Manual Strategy	*Benchmark*
Cumulative Return	0.45118	0.01230
Standard Deviation	0.01302	0.01698
Average Daily Return	0.00082	0.00016
Sharpe Ratio	1.00396	0.15708

Performance Comparison Out-of-Sample

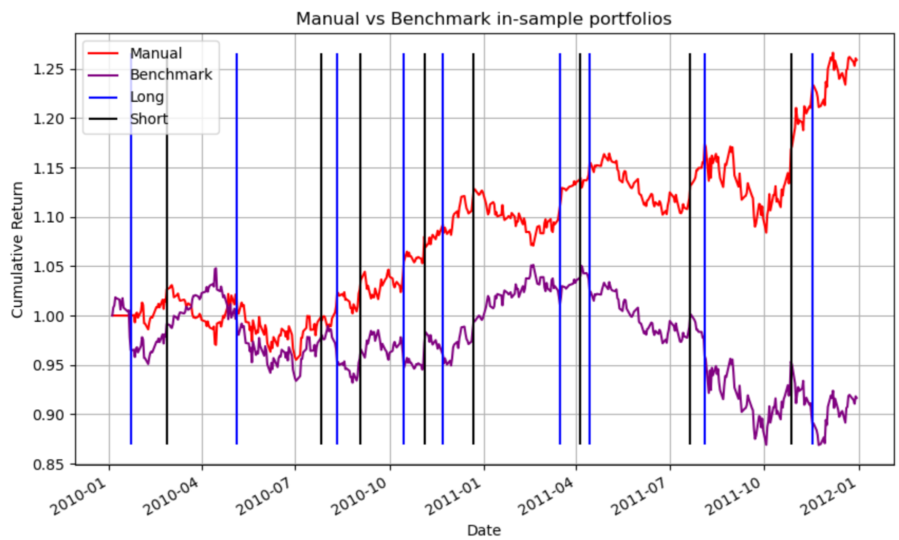

Figure 2

	Manual Strategy	*Benchmark*
Cumulative Return	0.25887	-0.08340
Standard Deviation	0.00726	0.00847
Average Daily Return	0.00048	-0.00013
Sharpe Ratio	1.05774	-0.25706

One more time figure 2 and the second table show that the voting manual strategy is effective resulting in higher cumulative return, lower volatility, better average daily return, and a higher Sharpe ratio. Judging by the Sharpe ratio, it's clear that the manual strategy outperformed in the out-of-sample dataset over the in-sample one although the cumulative return is smaller.

Since the strategy doesn't learn, there is not that much difference to discuss when it comes to in-sample versus out-of-sample. This just shows how this voting strategy works on 2 different datasets and from the provided information. It is not a bad strategy.

4. Strategy learner

Problem Preparation and Description: The problem is prepared in the code using `'_prepare_signal'` method. This method fetches the historical data for the stock in question using its ticker symbol. Then, it prepares the four indicator signals as mentioned above. The next step is calculating the future return by shifting the prices to for several days. In this project, 7 days were found to produce the best results. Future gain is calculated depending on the impact value the thresholds are determined to convert the future gains to long and short signals. These short and long signals will be the labels the random forest learns from the In-Sample dataset and predict over both In-Sample and Out-Of-Sample datasets. These signals are then used to predict buy and sell entry points when trying to produce the trades dataset.

Model Hyperparameters: When training and using the model to predict, an important problem that arises is the best hyperparameter values possible. I used a confusion matrix to reach the Bag size and minimum leaf size, which improves the model's performance over in-sample and out-of-sample. In this Project, the bag size was set to 80 for each of the size 260 rows, and the minimum leaf size was set to 26. The performance will be shown in Experiment 1.

Data Standardization: since the decision trees, the building block of random forests, use each column's data separately and create splitting conditions specific to each column, no data standardization was required for this project.

5. Experiment 1

Description: This experiment builds to strategy objects using the 2 classes `StrategyLearner` and `ManualStrategy`. Then it trains the Strategy learner over the in-sample data class from January 1st, 2008 to December 31st, 2009. Then the 2 classes are used to test the policies for both datasets. The results are then plotted on a graph vs the benchmark portfolio. In most cases, learning policy outperforms manual strategy. However, due to the random nature of random forests, this is not always guaranteed. Figures 3 and 4 show a case where the strategy learner outperforms the manual strategy for both in-sample and out-of-sample data.

Evidence and discussion

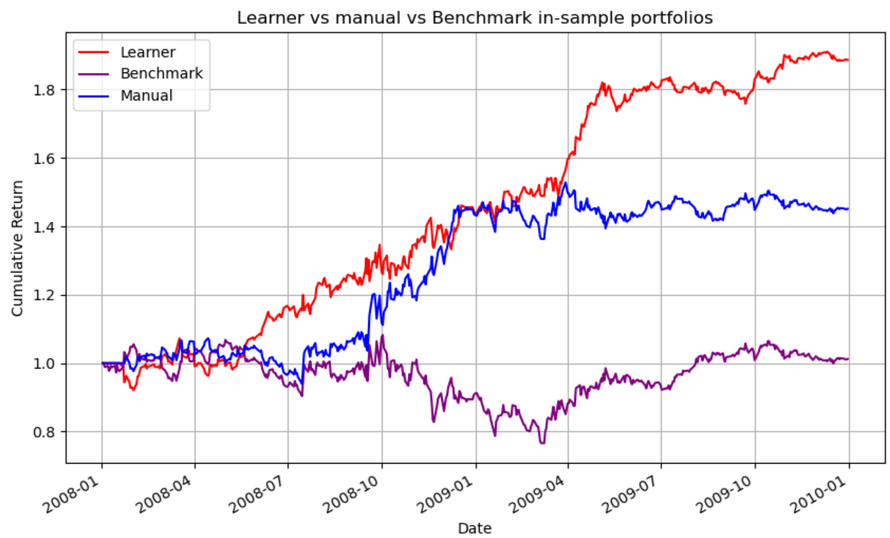

Figure 3

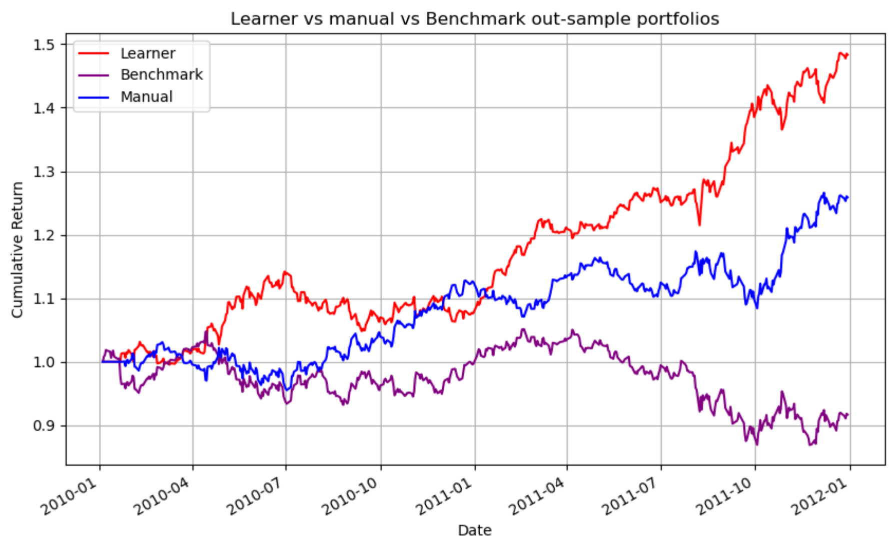

Figure 4

The way strategy learner achieves this high return is by trading a lot. Although strategy learners may make some bad trades, on average they will make more good trades than bad ones. However, as the cost of trading goes up it starts to get challenging for Strategy learners to perform well. This is the motivation behind Experiment 2.

6. Experiment 2

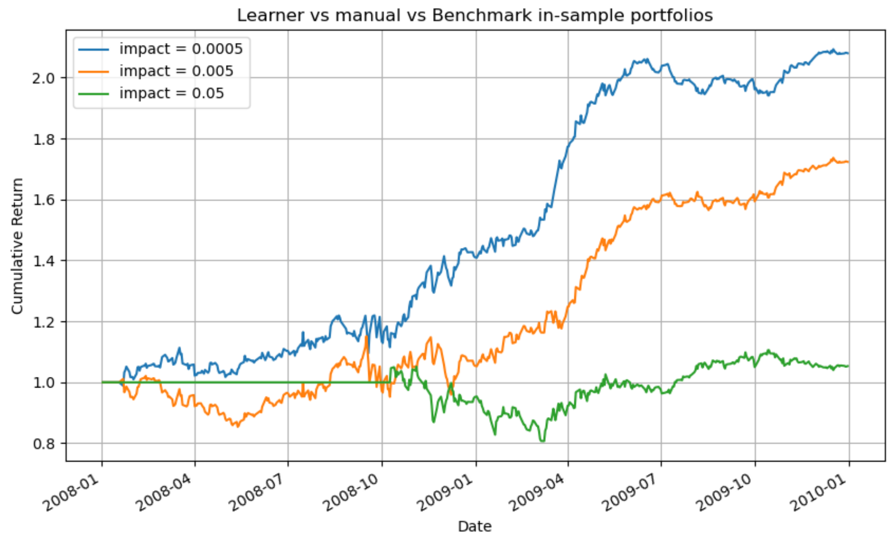

Figure 5

	0.0005	*0.005*	*0.5*
Long Entrances	68	61	1
Short Entrances	69	61	0
Cumulative return	1.08007	0.72348	-0.00013

Description: The strategy learner presented in this project automatically adjusts the buy and sell signal in the training data depending on the impact. For high impact, it only flags rows with large price changes as buy or sell signals leading it to build a more conservative model to decrease trading costs. This leads to a smaller number of trades. In the presented run of the experiment, we can see that for low impact the model resulted in more trading activity leading to higher gains.

In this experiment, we try out three impact values 0.0005, 0.005, and 0.05 to see their impact on trading activity and cumulative return.

The above table shows long entrances, short entrances, and cumulative return for the three impact values.

7. Conclusion

A random frost Strategy learner can be a strong tool in an environment where higher-frequency trading is not severely penalized. With that said, we can see that we have strong indicators that were able to generate good returns even with a very naïve manual strategy.