CS7646 Project 8

Petros Venieris
pvenieris3@gatech.edu

1 INTRODUCTION

In the realm of financial markets, the pursuit of systematic, data-driven strategies has become increasingly important as traders and institutions seek to optimize their decision-making processes. This report explores the development, testing, and evaluation of trading strategies using both manual and algorithmic approaches. The focus is on leveraging historical stock price data to generate actionable trading signals while adhering to a realistic market simulation that includes the effects of commissions and market impact.

Can an algorithmic strategy learner outperform manual trading strategies and benchmark approaches, such as "buy and hold"? The hypothesis is that by employing a strategy learner, which adaptively refines its decisions based on historical data, it is possible to achieve superior returns compared to manual strategies and passive benchmarks..

2 INDICATORS

The first indicator, Bollinger Bands %B, measures the position of a stock's price relative to its Bollinger Bands. Bollinger Bands are calculated using a moving average and a standard deviation multiplier, forming an upper and lower boundary around the price. The %B value ranges between 0 and 1, where 0 indicates the price is at the lower band, 1 indicates it is at the upper band, and 0.5 represents the midline. This indicator helps identify overbought or oversold conditions in the market. For the Manual Strategy, specific thresholds such as %B below 0.2 for oversold and above 0.8 for overbought are manually defined. In contrast, the Strategy Learner adapts its thresholds dynamically during training to maximize returns based on historical data patterns.

The second indicator is the Relative Strength Index (RSI), which evaluates the strength of recent price movements by comparing the magnitude of gains to losses over a specified period. RSI values range from 0 to 100, with levels below 30 typically signaling oversold conditions and levels above 70 indicating overbought conditions. In the Manual Strategy, these thresholds are fixed, providing clear decision points for buying and selling. For the Strategy Learner, the RSI is treated as a continuous feature, allowing the algorithm to identify nuanced relationships between RSI levels and optimal trading actions. The lookback period, commonly set to 14 days, is an adjustable parameter optimized during strategy training.

The third indicator, the Simple Moving Average (SMA), measures the average price of a stock over a given period. This indicator is often used in conjunction with a short-term and a long-term SMA to identify trends through crossovers. For example, a bullish signal occurs when the short-term SMA crosses above the long-term SMA, while a bearish signal occurs when the reverse happens. In the Manual Strategy, the short-term and long-term periods are predefined, such as 20 days for the short-term and 50 days for the long-term SMA. For the Strategy Learner, these parameters are optimized to align with the stock's historical behavior, and the learner identifies the most profitable SMA relationships during training.

3 MANUAL STRATEGY

The Manual Strategy combines the selected indicators—Bollinger Bands %B, Relative Strength Index (RSI), and Simple Moving Average (SMA)—to generate buy, sell, or hold signals. The combination is designed to capture different aspects of market behavior: %B identifies overbought or oversold conditions, RSI quantifies the momentum of price changes, and SMA evaluates the overall trend. By integrating these indicators, the strategy aims to identify high-probability trading opportunities while minimizing false signals.

The overall signal is created using a weighted decision framework. For instance, a buy signal is triggered when %B is below 0.2, indicating oversold conditions, RSI is below 30, reflecting weak momentum, and the short-term SMA is below the long-term SMA, indicating a potential reversal from a downward trend. Similarly, a sell signal occurs when %B exceeds 0.8, RSI is above 70, and the short-term SMA crosses below the long-term SMA, suggesting a weakening uptrend. Hold signals are generated when these conditions are neutral or contradictory, avoiding unnecessary trades during unclear market conditions.

Entry and exit decisions are based on the alignment of these indicators. For entry, the strategy waits for clear confirmation of oversold conditions (for buying) or overbought conditions (for selling). For exit, it relies on the reversal of these conditions, such as %B returning to neutral (around 0.5) or RSI moving back within a 40–60 range, which often indicates the cessation of extreme momentum. The inclusion of SMA crossovers adds a trend-following component, reducing the risk of reacting prematurely to transient market fluctuations.

Manual Strategy Metrics:

Cumulative Return: 0.1555

Stdev of Daily Returns: 0.0054

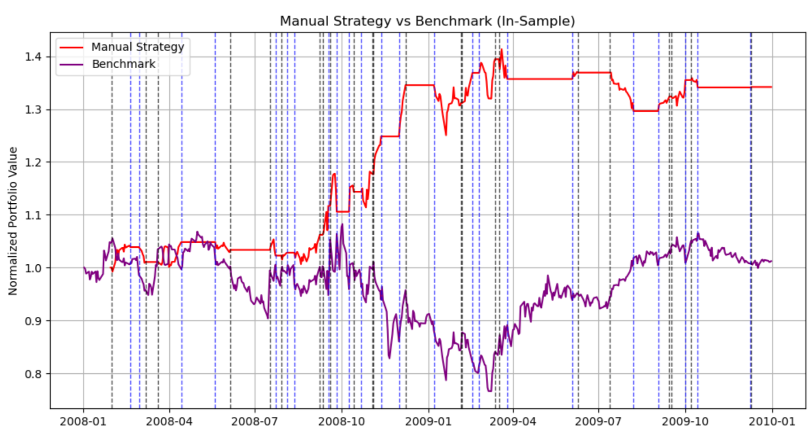
Mean of Daily Returns: 0.0003

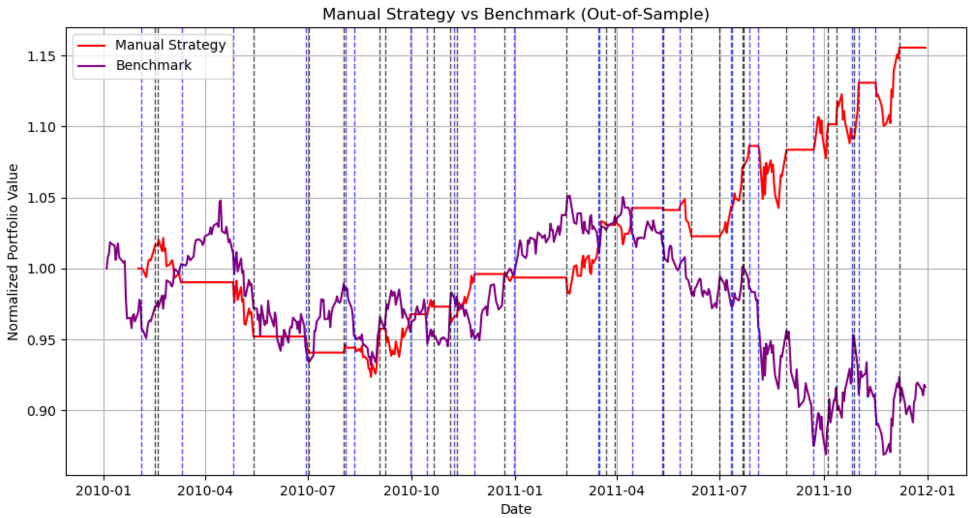
Benchmark Metrics:

Cumulative Return: -0.0834

Stdev of Daily Returns: 0.0085

Mean of Daily Returns: -0.0001





We can see that our Manual strategy outperforms our benchmark on every metric, on the out sample period it takes some more time to start notice the positive performing. This strategy is effective because it combines short-term momentum with broader trend analysis, balancing responsiveness with stability. However, it assumes that past patterns in indicator behavior are indicative of future performance, which may not always hold true. This limitation is particularly relevant in highly volatile or anomalous market conditions where traditional indicators may fail. The observed differences arise from changes in market conditions. The in-sample period may have had more consistent trends and momentum that aligned well with the indicators, while the out-of-sample period exhibited more unpredictable behavior. Additionally, the static thresholds and rules of the Manual Strategy, while effective during training, limited its adaptability to new patterns.

4 STRATEGY LEARNER

The Strategy Learner was designed to determine optimal trading actions based on a machine learning model. The process began by framing the problem of predicting stock price movements as a supervised learning task. The objective was to predict the direction of stock prices (whether they would go up, down, or remain the same) based on various technical indicators. This was treated as a classification problem, where the model's task was to classify each day’s trading action as either "buy," "sell," or "hold."

The first step in setting up the learning problem was to collect relevant stock price data. In this case, the stock symbol chosen was "JPM" and data was gathered for two time periods: an in-sample period from 2008 to 2009, and an out-of-sample period from 2010 to 2011. For each of these periods, stock prices were used to calculate technical indicators, which would serve as input features for the model. The indicators used included the Simple Moving Average (SMA), Bollinger Bands, and the Moving Average Convergence Divergence (MACD). These indicators were selected because they are commonly used in technical analysis and have proven to be valuable for detecting trends, volatility, and momentum in stock prices.

The second step involved generating labels for the model. Since the task is to predict price movement, the labels were constructed based on the future direction of the stock price. A lookahead window of 5 days was chosen, meaning that the model would predict whether the stock price would increase or decrease in the next 5 days. If the price was expected to rise, the label would be "1" (buy), if it was expected to fall, the label would be "-1" (sell), and if there was no significant movement expected, the label would be "0" (hold). This approach allowed the model to learn from historical price patterns and make predictions that would inform the trader's decision-making process.

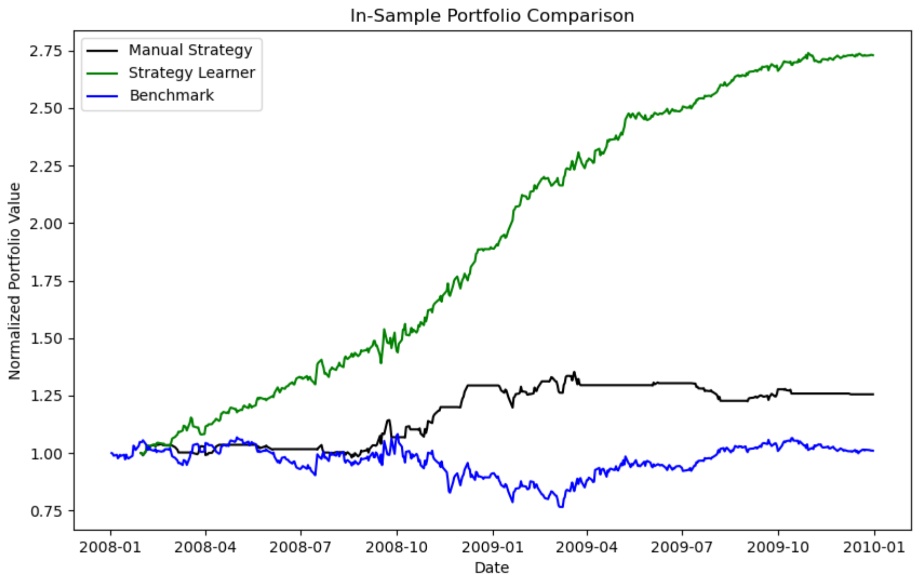
Hyperparameters were an essential part of the learning process. The BagLearner method, which was used for the learning task, is an ensemble method that builds multiple weaker models (in this case, RTLearners) and combines their predictions to improve accuracy and robustness. The primary hyperparameters for the model included the number of bags (or individual models) used in the ensemble, which was set to 20. This was chosen based on experimentation with different values to balance model complexity and performance. The leaf_size for each RTLearner was set to 5, which determines the maximum number of samples that can be used in a terminal node of the decision tree. This parameter helps control overfitting, and a smaller value like 5 was chosen to avoid overly complex models.

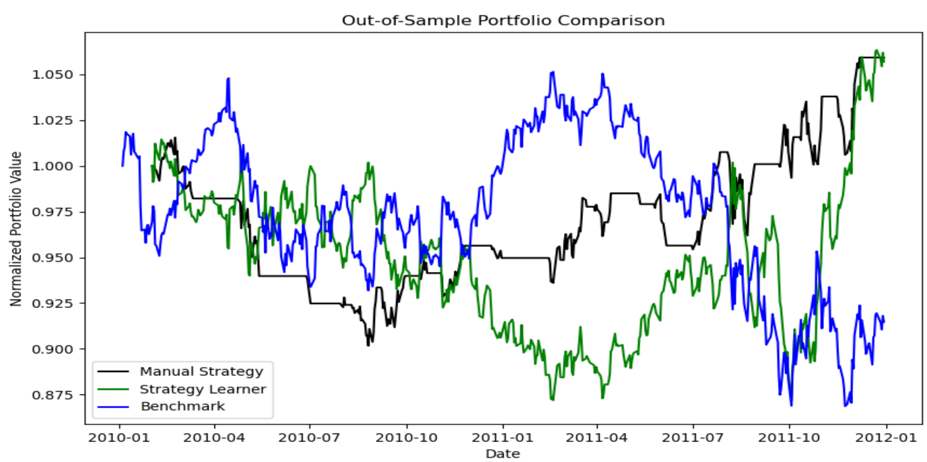
In terms of data preprocessing, the features were not standardized or normalized. This was a conscious decision because the chosen indicators—such as the SMA and MACD—are already based on price data that are on similar scales. Additionally, these indicators are designed to capture the relative movements of the stock prices, and normalizing or scaling them could distort the meaningful relationships between the indicators and the stock's price movement. Therefore, no additional data transformations were applied beyond the calculation of the indicators and the generation of labels.

This framework provided an efficient way of automatically making trading decisions, leveraging machine learning to adapt to patterns in the data and adjust trading actions accordingly.

EXPERIMENT 1

In Experiment 1, we aimed to compare the performance of a manually defined trading strategy with that of a machine learning-based strategy and the benchmark. The goal was to assess whether the Strategy Learner could outperform the manual strategy by adapting to patterns in the data. Assumptions made during the experiment included ignoring transaction costs and market impact, using a starting portfolio value of $100,000, and applying the same technical indicators for both strategies. The experiment aimed to test the hypothesis that the machine learning-based strategy would be more effective in predicting stock movements and generating higher returns compared to the rule-based manual approach.


In-Sample Portfolio Comparison
Manual Strategy
Strategy Learner
Benchmark
Normalized Portfolio Value
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
2008-01
2008-04
2008-07
2008-10
2009-01
2009-04
2009-07
2009-10
2010-01
Date


Out-of-Sample Portfolio Comparison
Manual Strategy
Strategy Learner
Benchmark
Normalized Portfolio Value
1.050
1.025
1.000
0.975
0.950
0.925
0.900
0.875
2010-01
2010-04
2010-07
2010-10
2011-01
2011-04
2011-07
2011-10
2012-01
Date

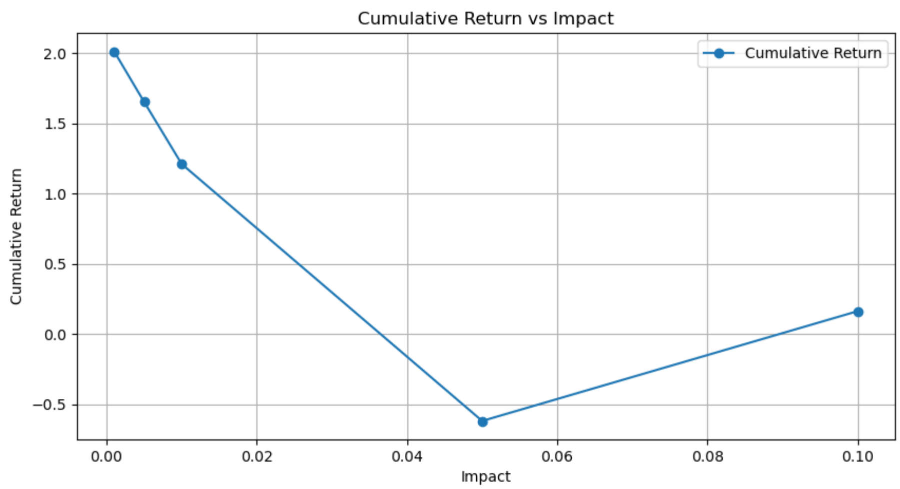
As we can see our strategy learner clearly outperform the benchmark and the manual strategy by a huge normalized value , is 2.5 times at least better than both the benchmark and the manual strategy in our insample , something that points out how good the machine learning algorithm ajdust to our data and how good our hypermarameters perform, on out of sample there is a different storry, while the benchmark is better nuch of the time we can see that in the end both our manual and stategy learner outperform the benchmark for at least 4 months. With a huge decline on benchmark and a big rise on our other 2. This showcase the overfitting on the insample because of our indicators but still we beat the market in the end and hope to keep this difference high. Yes for the in sample part of the code since we build our models based on these data it is normal to outperform this easily the benchmark , still the percentage of the strategy learner is pretty satisfying but for outsample we have to be cautious and make a more effective algorithm.

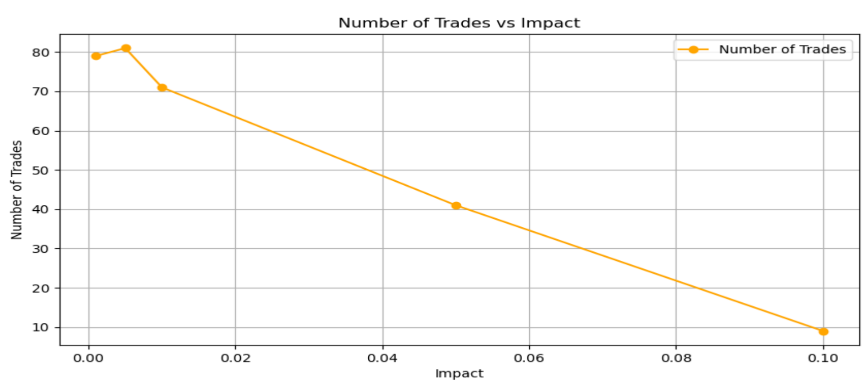
EXPERIMENT 2

In Experiment 2, the goal is to examine how varying the impact factor, which represents transaction costs or frictions associated with executing trades, affects the performance of a trading strategy. The hypothesis is that as the impact increases, the strategy's performance will be influenced in two significant ways: cumulative return and the number of trades executed. First, regarding the cumulative return, it is expected that as the impact factor increases, the cumulative return of the portfolio will decrease. This is because a higher impact represents higher transaction costs, meaning the strategy will lose more value per trade. Therefore, higher transaction costs will reduce the profitability of trades, and ultimately, the cumulative return. At low impact values, the strategy may generate higher returns since trades can be executed with less friction, but as the impact grows, the strategy will be penalized for frequent or large trades, leading to a reduction in return.

Secondly, the hypothesis suggests that the number of trades executed by the strategy will decrease as the impact factor increases. This is because higher transaction costs erode the profitability of each trade, making it less attractive for the strategy to engage in frequent buying and selling. With a higher impact, the strategy becomes more conservative, opting to execute fewer trades to avoid incurring large transaction costs.

he experiment is designed to assess these two key metrics—cumulative return and number of trades—by training the StrategyLearner with different impact values (0.001, 0.005, 0.01, 0.05, and 0.1)





The obvious result is happening , as the impact increases the number of trades and the cumulative return are declining with a huge rate , our trades are almost half by the time we get on 0,1 commission and the cumulative returns are negative at some point with barely staying on 2 after commission hits 0.5 These changes illustrate how transaction costs influence both the profitability and the trade frequency of a machine learning-based trading strategy.

1