# Project 8:
# Strategy Evaluation Report

Kamran Munawar
kmunawar3@gatech.edu

## 1 INDICATOR OVERVIEW:

For this assignment, I am utilizing the following indicators to develop and evaluate the trading strategies:

### 1.1 Momentum

The Momentum Indicator is a straightforward yet powerful tool used to measure the rate of change in a stock's price over a specific time period. By comparing the current price of a stock to its price from a previous period, this indicator helps determine whether the stock is gaining or losing momentum. A positive momentum value indicates that the stock price is increasing relative to the look-back period, while a negative momentum value signals a decline in the stock price over the same time frame.

The formula for calculating momentum is as follows:

$$\text{Momentum}(t) = P(t) - P(t-N)$$

In both the Strategy Learner and the Manual Strategy, a 14-day look-back window is employed to calculate the momentum

### 1.2 Simple Moving Average (SMA)

SMA is a popular technical indicator used to smooth price data by averaging a stock's price over a set period. It helps to identify trends and minimize the impact of short-term fluctuations. The formula for the SMA is given as

$$\text{SMA}(t) = \frac{P(t) + P(t-1) + \cdots + P(t-(N-1))}{N}$$

A buy signal occurs when the price crosses above the SMA, indicating a possible uptrend. Conversely, a sell signal occurs when the price crosses below the SMA, suggesting a potential downtrend.In both the Strategy Learner and the Manual Strategy, a 20-day look-back window is utilized to calculate the Simple Moving Average (SMA) and the Price over SMA ratio. This ratio is used to assess whether the price is relatively high or low compared to its recent average

### 1.3 The Commodity Channel Index (CCI)

The Commodity Channel Index (CCI) is a momentum-based technical indicator used to identify cyclical trends in asset prices. It compares the current price level to its historical average over a specific period, quantifying how far the price deviates from its average. The result is then normalized by dividing it by the Mean Absolute Deviation (MAD). CCI is particularly useful for detecting overbought and oversold conditions and identifying potential price reversals.

$$\text{CCI} = \frac{P_{\text{typical}} - \text{SMA}}{0.015 \times \text{MAD}}$$

Where $P_{\text{typical}} = \frac{(\text{High}+\text{Low}+\text{Close})}{3}$ is the typical price. When the CCI rises above +100, it signals that the asset may be overbought, suggesting a potential sell opportunity. Conversely, when the CCI falls below -100, it indicates that the asset is likely oversold. In both the Manual Strategy and the Strategy Learner, a 20 day look back window is employed for calculating the Commodity Channel Index (CCI)

## 2 MANUAL STRATEGY

In Manual Strategy we try to combining multiple indicators like SMA, CCI, and Momentum into a single trading strategy enhances the decision making process by leveraging the strengths of each indicator while mitigating their weaknesses. for example SMA Helps to avoid trading against the trend but is is also Lagging indicator and slow to react to sudden price movements. CCI effective in detecting extremes and potential reversals but it can also produce false signals in strong trending markets.Momentum reacts quickly to changes in price direction but is prone to noise in volatile or sideways markets. This strategy ensures only high-confidence signals (alignment across all three indicators) lead to trading actions. Mixed signals result in a hold decision, reducing the risk of acting on ambiguous market conditions.

A buy signal is generated when the price is significantly below the SMA (Price/SMA < 0.95), the CCI indicates an oversold condition (CCI < -100), and Momentum shows downward movement (Momentum < -0.05). Conversely, a sell signal is triggered when the price is significantly above the SMA (Price/SMA > 1.05), the CCI indicates an overbought condition (CCI > 100), and Momentum shows upward movement (Momentum > 0.05). In cases where the indicators provide mixed signals, the strategy defaults to a hold decision, minimizing the risk of acting on ambiguous or conflicting market conditions.

Fig1 showing comparison of performance of manual strategy versus the benchmark for the in-sample.The Manual Strategy significantly outperformed the benchmark, leveraging frequent buy and sell signals to capitalize on the volatile market conditions during 2008 and 2009. The manual portfolio showed steady growth, achieving substantial cumulative returns, while the benchmark remained largely stagnant with minimal growth. The use of technical indicators like SMA, CCI, and Momentum allowed the strategy to time trades effectively, avoid prolonged downturns, and outperform the passive buy and hold approach of the benchmark. The high frequency of trades during this period further underscores the strategy's ability to navigate the heightened volatility and exploit trading opportunities.

Fig2 showing comparison of performance of manual strategy versus the benchmark for the out-sample period.The Manual Strategy continued to demonstrate superior performance by maintaining a portfolio value consistently higher than the benchmark. Although market conditions were relatively calmer compared to the in-sample period, the manual strategy effectively timed buy and sell decisions using its indicators, avoiding major declines observed in the benchmark portfolio. The strategy showed a more cautious approach with fewer trades but still managed to outperform the benchmark, which declined steadily due to its inability to adapt to fluctuating market conditions.

The performance of the Manual Strategy in the out-sample period (2010–2011) reflects its ability to adapt to unseen data and market conditions.Compared to the in-sample period, the out-of-sample period had fewer trades, indicating a more cautious approach. The strategy executed trades only when all indicators aligned, ensuring high-quality signals. The benchmark portfolio, being a passive strategy, steadily declined throughout the out-of-sample period. This decline highlights its vulnerability to market fluctuations and inability to adjust to changing conditions. In contrast, the Manual Strategy demonstrated adaptability, consistently maintaining a higher portfolio value and outperforming the benchmark.

The differences between in-sample and out-sample performance in a trading strategy can be attributed to several factors, including market conditions, data fitting risks, and the inherent limitations of using historical data to predict future outcomes. In-sample period (2008-2009) includes the 2008 financial crisis, char-

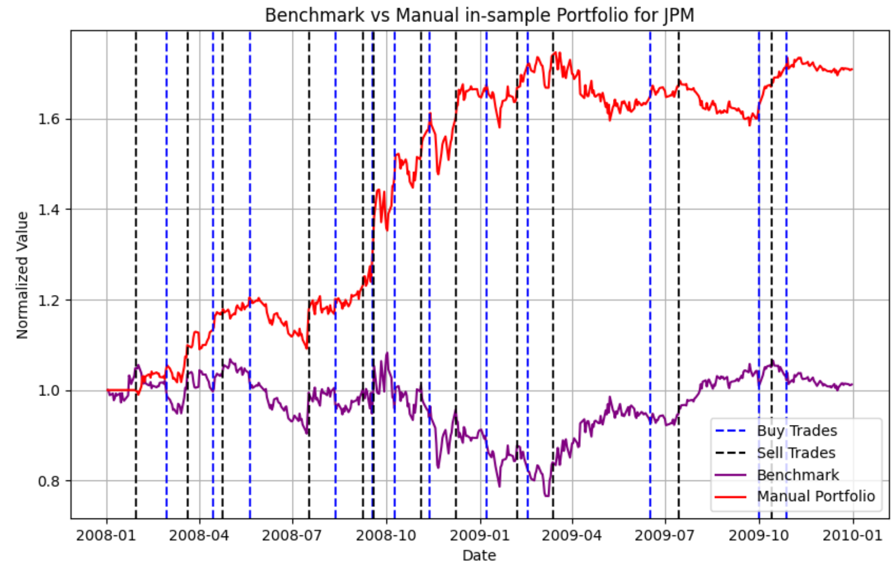


*Figure 1*—manual vs benchmark in-sample

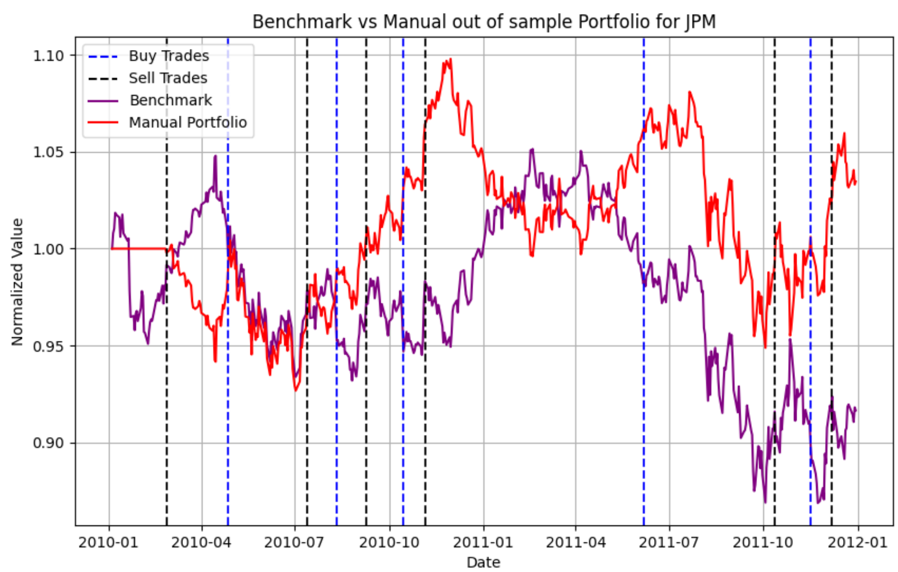


*Figure 2*—manual vs benchmark in-sample

acterized by extreme market volatility and significant price swings. These conditions created more opportunities for the Manual Strategy to exploit overbought and oversold conditions detected by indicators. out-sample period (2010-2012) The market was relatively calmer during this period, with less extreme volatility and more stable trends. These conditions reduced the number of trading opportunities for the Manual Strategy. As a result, the strategy executed fewer trades, which limited its profit potential.

The Manual Strategy was tuned using historical data from this period, allowing it to "fit" the specific patterns and trends present in the training data. This often results in higher performance during in-sample testing because the strategy is effectively optimized to exploit these patterns. The out-sample period represents unseen data, where the strategy's ability to generalize is tested. Market behavior in 2010–2011 might not fully align with the patterns and conditions observed during the in-sample period. For instance Volatility and trading ranges may have differed.Indicator thresholds (overbought/oversold levels) may not have been as effective in identifying profitable trading opportunities.

## 3 STRATEGY LEARNER

To frame the trading problem as a learning problem, the Strategy Learner uses a reinforcement learning (RL) framework, specifically a Q Learner, to learn an optimal trading policy based on historical market data.

### 3.1 Learning Environment

- State Space: The state of the market is represented by a combination of indicators: Momentum, SMA, and CCI. These indicators are calculated for the stock prices over time
- Action: the learner can take one of three possible actions 0:Hold, no trading activity. 1: Buy, go long by 1000 shares or switch from short to long. 2: Sell, go short by 1000 shares or switch from long to short.
- Reward Function: The reward is designed to encourage profitable trades and discourage losses. The learner calculates the reward for each action based on the change in prices and the market impact cost. For example Buy: Reward equal to price change in favor of the long position. Sell: Reward equal to price change in favor of the short position. Hold: Reward equal to 0, as no trade is

made.

### 3.2 add_evidence method

The add_evidence method is used to train the learner on historical data Historical stock prices are fetched, and technical indicators (SMA, CCI, Momentum) are computed.These indicators are discretized into states using discretize_states method. The learner iterates over the training data multiple times (e.g., 50 epochs), simulating the decision making process day by day. For each day the current state (based on indicators) is passed to the Q-Learner.Reward is calculated based on holding , price difference from previous day. Rewards are adjusted for transaction costs and market impact to ensure realistic trading scenarios.The next action is determined based on the current day reward and state.The Q Learner updates its Q table using the Q Learning formula to improve its policy.

### 3.3 testPolicy method

Once trained, the strategy is tested using the testPolicy method.The same indicators (SMA, CCI, Momentum) are calculated for the test period, and states are discretized.The Q Learner applies its learned policy to the test data.For each day, the current state is passed to the learner.The learner selects an action (buy, sell, or hold) based on its Q table. Trades are recorded, ensuring they adhere to the constraints of maximum holdings (-1000, 0, +1000 shares).The strategy transitions between positions (e.g., from long to short) as needed, with appropriate trade sizes (+1000, -1000, or ±2000 shares).

### 3.4 Hyperparameters

The Q Learner in the StrategyLearner has several key hyperparameters that control its behavior and performance. These hyperparameters include num_states(=1000) determines the size of the Q table, representing the total number of possible discrete states the learner can encounter. num_actions(=3) defines the number of possible actions the learner can take Hold, Buy, or Sell. Learning rate(alpha=0.5) determines how much the Q value is updated during learning. Higher values cause the learner to give more weight to new information, while lower values make it rely more on prior knowledge.Discount factor (gamma = 0.3) Controls the importance of future rewards compared to immediate rewards. A value close to 1 emphasizes long term rewards, while a value close to 0 focuses on short term rewards.Random action rate (rar =0.8) The probability of taking a random action instead of the action suggested by the Q table. Encourages exploration during training.Random action decay rate(radr = 0.765) Controls the rate at which the random action rate (rar) decreases over time, transitioning the learner from exploration to exploitation.DynaQ Simulations (dyna = 0) Specifies the number of simulated experiences the learner generates in each training iteration to augment real world experiences.

### 3.5 Discretized data

The discretize_states function converts continuous indicator values into discrete states, enabling the Strategy Learner to operate effectively in a finite state action space. The function takes a list of indicators and a parameter for the number of discrete buckets (disc_buckets), defaulting to 10. Each indicator is processed individually and any missing data (NaNs) is replaced with zeros. The function then discretizes each indicator using quantilebased bucketing (qcut), which divides the range of values into equally populated buckets. This approach ensures that each bucket contains roughly the same number of data points, making it robust to non-uniform distributions.

Once all indicators are discretized, the function creates a composite state by combining them into a single integer value. This is achieved by assigning weights to each discretized indicator, with weights determined by the positional power of the number of buckets The final combined state is calculated as the dot product of the discretized values and their weights, resulting in a unique integer representation for each combination of indicator states.

## 4 EXPERIMENT 1 (MANUAL STRATEGY / STRATEGY LEARNER)

This experiment is designed to compare the performance of two trading strategies,Manual Strategy and Strategy Learner against a Benchmark during both in-sample (2008–2009) and out-sample (2010–2011) periods for JPM stock. The goal is to assess the effectiveness of these strategies in terms of cumulative returns, risk management, and adaptability to market conditions. In sample parameters are parameters are symbol = "JPM", sd = January 1 2008 , ed = December 31 2009, sv = 100,000 and commission = 9.95, impact = 0.005 and out sample parameters are parameters are symbol = "JPM", sd = January 1 2010 , ed = December 31 2011, sv = 100,000 and commission = 9.95, impact = 0.005

Fig 3 showing Benchmark vs Manual Strategy vs Strategy learner in-sample

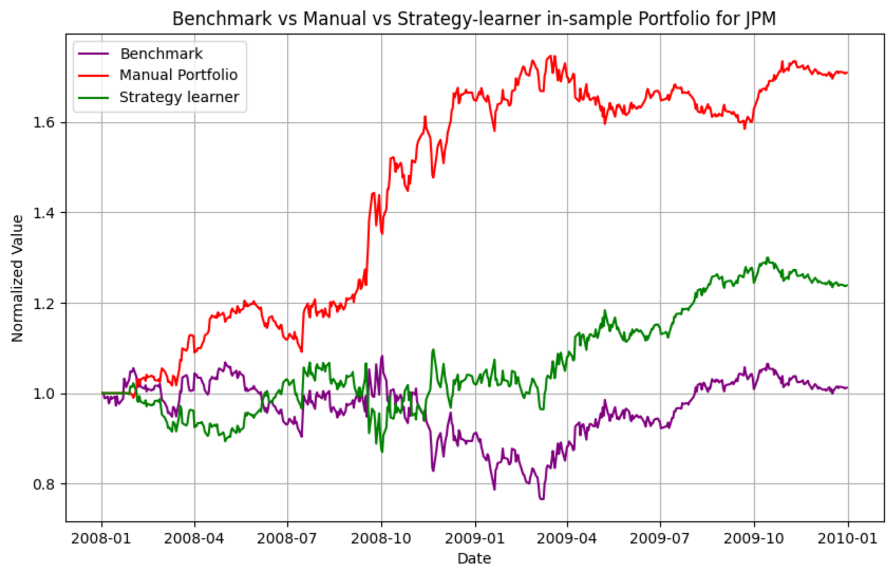


***Figure 3***—Benchmark vs Manual Strategy vs Strategy learner in-sample JPM portfolio

JPM portfolio .The Benchmark (purple line) represents a simple buy-and-hold strategy that starts and ends around normalized values near 1, exhibiting low volatility and consistent performance. The Manual Portfolio (red line) significantly outperforms the Benchmark, with a peak around 1.6, indicating that the manual strategy captures favorable market movements effectively. However, its fluctuations suggest higher risk due to increased volatility. The Strategy Learner (green line) shows a more consistent upward trajectory compared to the Manual Portfolio, with notable gains and reduced risk, ending above 1.4. This suggests that the Strategy Learner efficiently adapts to market dynamics, balancing returns and risk better than the manual approach while clearly outperforming the Benchmark. The results highlight the superior adaptability of the Strategy Learner while demonstrating the limitations of static strategies like the Benchmark.

Figure4 illustrates the performance of Benchmark, Manual Portfolio, and Strategy Learner during the out sample period (2010–2011). Unlike the in-sample period, the Benchmark (purple line) demonstrates relatively stable performance, fluctuating around normalized values near 1 but ending slightly below. The Manual Portfolio (red line) shows moderate success, staying consistently above the Bench-

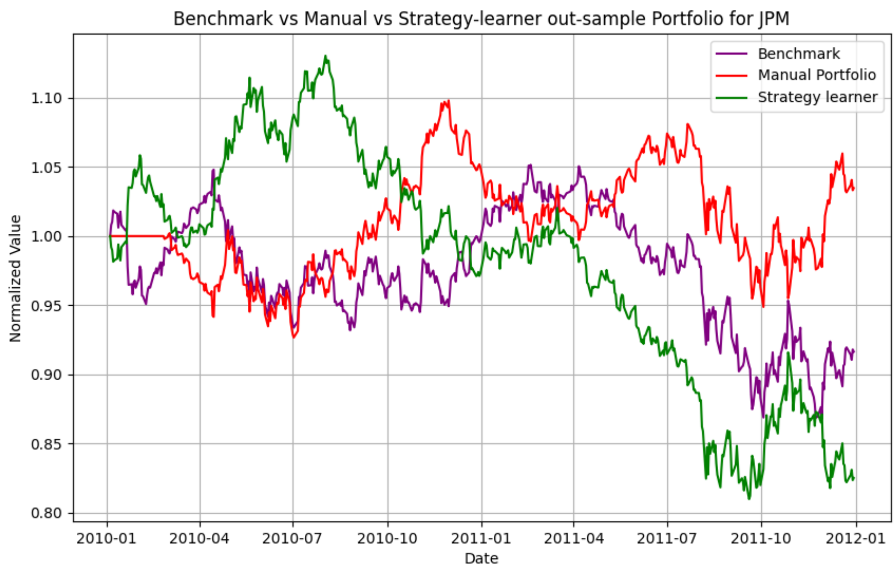


*Figure 4*—Benchmark vs Manual Strategy vs Strategy learner out-sample JPM portfolio

mark and ending near a normalized value of 1.05, indicating slight profitability with reduced volatility compared to the in sample period. However, the Strategy Learner (green line) performs poorly, experiencing significant drawdowns and ending near 0.8. This underperformance suggests that the Strategy Learner's model may have overfitted to the in-sample data and failed to generalize effectively to out sample market conditions. The graph highlights the limitations of Q learning based strategies when faced with unseen market dynamics, while simpler strategies like the Manual Portfolio demonstrate greater robustness in this scenario.

### 4.1 Experiment 2 (Strategy Learner)

### 4.2 Hypothesis

As the impact value increases, in-sample trading behavior will shift to less frequent trading, resulting in reduced cumulative returns and lower mean daily returns. This occurs because higher impact values increase the effective transaction costs, making frequent or marginal trades less profitable.

### 4.3 Rationale

When the impact value increases, each trade incurs a larger effective cost due to market impact. This disincentivizes frequent trading as the strategy learner adjusts to maximize net returns.With higher impact values, the strategy learner is likely to favor trades with higher confidence levels or larger expected gains to offset the increased transaction costs.

### 4.4 Experiment

Run Strategy learner multiple times using different values of impact. Other parameters are symbol = "JPM", sd = January 1 2008 , ed = December 31 2009, sv = 100,000 and commission = 0.00

### 4.5 Result

Table 1 showing commutative return and Mean of daily return. The data supports the hypothesis that increasing the impact value shifts the trading strategy towards fewer and more cautious trades, reducing both cumulative returns and the mean of daily returns due to the compounding effect of higher transaction costs.

*Table 1*—strategylearner ouput for different values of impact.

| impact | commutative return | Mean of Daily Returns |
|---|---|---|
| 0.00 | 0.4189 | 0.000803 |
| 0.0025 | 0.315335 | 0.000659 |
| 0.005 | 0.242771 | 0.000551 |
| 0.0075 | 0.170206 | 0.000436 |
| 0.01 | 0.097641 | 0.000315 |