Georgia Institute of Technology
Machine Learning For Trading
Fall 2019

Project 8: Strategy Learner

Jae Ro

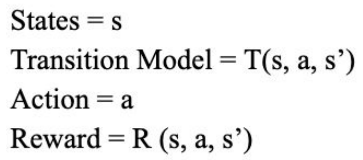
Strategy Learner - CS 7646

Introduction

In this assignment I attempt to create a trading agent that utilizes Reinforcement Learning (specifically Q-Learning) in order to optimize my returns between a given period, specifically for the security of JP Morgan Chase and Co. To do this, I will define an appropriate Markov Decision Process (MDP) using the previously specified technical indicators, implement a Q-Learning based trading agent on the defined MDP with the appropriate set of trading rules, compare in-sample results to that of the Manual Strategy used in the previous assignment, and test the effects Market Impact can have on this new trading strategy. For clarification, the stock symbol used in all of the graphs in this paper is JPM (denoting the security of JP Morgan Chase and Co.). Accordingly "In-Sample" data refers to JPM stock prices between 2008-1-1 and 2009-12-31, while "Out-of-Sample" data refers to to JPM stock prices between 2010-1-1 and 2011-12-31.

Part I: Framing the Trading Problem as a Reinforcement Learning Problem

First and foremost, in order for a reinforcement learning-based stock trading strategy can be implemented, the whole stock trading problem must be framed as a Markov Decision Process (MDP). Accordingly, the features that make up an MDP are as follows:



States = s
Transition Model = T(s, a, s')
Action = a
Reward = R (s, a, s')

Thus, an MDP refers to the environment defined by the attributes listed above that is used to model decision-making via an agent that acts according to the rules of said MDP. Furthermore, the reinforcement learning algorithm used by the Strategy Learner trading agent attempts to find the best possible action to take in each state such that it maximizes the expected value of the total reward (the optimal policy). As such, we need to define these MDP attributes correctly in the context of stock trading so that our agent can behave properly and make optimal trading decisions (Buy, Sell, Hold) on any given day. Note: due to the fact that we are using Q-learning (a model-free reinforcement learning algorithm), we do not need to define the Transition Model, and so we don't include it in the following definitions.

1.1 States

For the purposes of our trading problem, we define the state of a given day's stock price in terms of our technical indicators (Price-to-SMA Ratio, Bollinger Bands Position, MACD Crossover, and Ichimoku Cloud). These technical indicators exhibit the momentum, volatility, and price trend of a particular stock and so a given trading day's state can be thought of as a combination of the outputs of these technical indicators. The hope that this combination of technical indicators can provide a means of successfully predicting future stock prices. However, the issue is that because the values output by our technical indicators are continuous the number of states in our MDP would be infinite, making it incredibly problematic for our Q-learner to find an optimal policy. Thus, to solve this issue and make training feasible, we discretized the outputs of our technical indicators by separating the trading days into a set number of bins based on the values of each indicator. Price-to-SMA, Bollinger Band Position, Ichimoku Cloud values were each split into 10 separate bins (labeled 0-9), while MACD values were split into 4 bins (labeled 0-3). As a result, the total number of states for our MDP was 4*10*10*10 = 4000 possible states (each state representing a unique combination of the 4 technical indicators' discrete outputs, ex. 2108). The discretized values for each indicator at a given day are concatenated together to form a single string state value in the order of ("MACD" + "price-to-sma" + "bb position" + "ichimoku cloud").

1.1.a Price to Simple Moving Average (SMA) Ratio

The simple moving average, also known as the rolling mean, is the average of all of recent closing prices within a fixed number of periods (ie. last n days). Accordingly, the price-to-sma ratio tells us how far above the current price is to the SMA or how far below the current price is in relation to the SMA. The price-to-sma ratio indicator values were discretized by identifying the min and max values, creating 11 evenly spaced thresholds between the min and max values, and then separating the trading days based on their indicator values into the 10 bins between these thresholds (labeled 0-9).

```
    |-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
Min     0       1       2       3       4       5       6       7       8       9    Max
```

1.1.b Bollinger Bands Position

This indicator is a measure of volatility in stock price and is typically expressed in terms of its position within the range of [1, -1] units of standard deviation. If the Bollinger Band Position is at 1.0, then the current stock price is 2 standard deviations above the 20-day SMA, and if the Bollinger Band Position is at -1.0, then the current stock price is at 2 standard deviations below the 20-day SMA. The Bollinger Band Position values were discretized in the same manner as the price-to-sma ratio indicator above.

```
    |-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
Min     0       1       2       3       4       5       6       7       8       9    Max
```

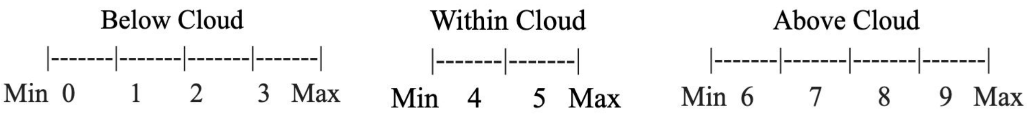
1.1.c Moving Average Convergence Divergence (MACD) Crossover Signal

The MACD is a trend-following momentum indicator that expresses the relationship between two exponential moving averages across short and long term periods. The MACD value measures momentum by subtracting the exponential moving average of the past 26 days from the exponential moving average of the past 12 days. A positive MACD value indicates a positive momentum in that the most recent window (past 12 days) has a higher value than that of the past 26 and as such is on the "rise". The opposite is true as well for measuring its momentum downward. Accordingly, the Signal Line represents the 9-day exponential moving average of the MACD. The indicator that we use is the crossover of the MACD value and the Signal Line. Thus the output states were discretized according to the following table (note: the ideal states are 1 and 2 because they indicate crossover).

MACD Value	Signal Line	State Label
negative	negative	0
negative	positive	1
positive	negative	2
positive	positive	3

1.1.d Ichimoku Kinko Hyo Cloud

The ichimoku cloud, which is the area encapsulated by the Leading Span A and the Leading Span B. This "cloud" attempts to forecast key areas of support and resistance that the stock prices may find in the future. Accordingly, prices that fall outside of the cloud are thought to be indicators of the current trend (below the cloud = down trend, above cloud = upward trend, within the cloud = trendless). As such, this indicator was discretized by initially dividing the stock prices into 3 separate data sets (below cloud, within cloud, above cloud). Within each of these partitioned datasets, the min and max values of each data set were calculated and then the trading days were then partitioned into bins as follows:



1.2 Actions

In the MDP world, the action is simply that the action that the agent can take in a current state that will move it to another state (or possibly the same state). For our problem context, the actions that our Strategy Learner can take are limited to: BUY 1000 shares, BUY 2000 shares, SELL 1000 shares, SELL 2000 shares, HOLD. However, there exist additional constraints such that the trading agent can only be in one of the following three positions: -1000 shares, 0 shares, +1000 shares. As a result, in the interest of simplicity we define the actions as merely HOLD, BUY, SELL (labeled as 0, 1, 2 respectively) and take care of the order quantity and additional constraints within the strategy algorithm rather than here in the definition of the actions.

1.3 Rewards

The reward for our MDP is the utility that the agent gains from being in state s, taking action a, and arriving in state s'. The reward for our trading strategy at a given time step (trading day) is defined as:

Reward = (1-impact) * holdings * percent change in share price

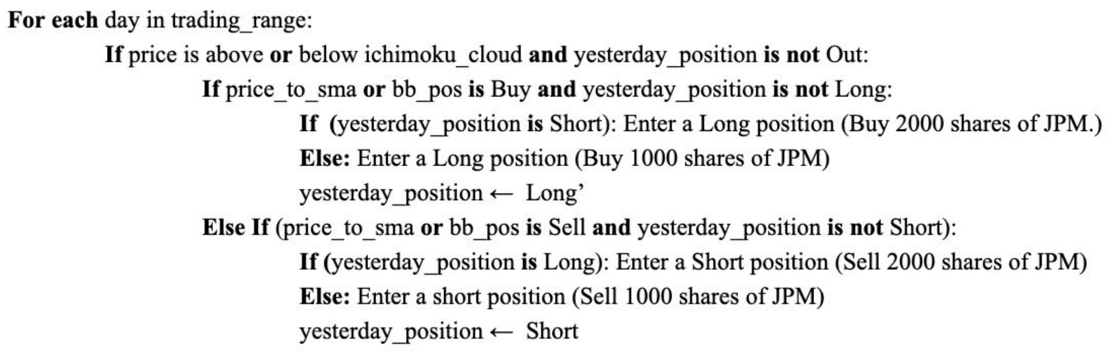
We used the multiplication of the holdings and percent change in share price as our reward because when the holdings is positive (+1000) and the share price is increasing (% change = +), we would get a positive reward. However if our holdings is positive (+1000), but the share price is decreasing (% change = –) then we should receive a negative reward because we bought the stock at a price higher than current. As such this reward system is set up in such a way to follow the logic that we want to “buy low, sell high”, and so the agent is reinforced in the same manner. In addition, the impact is incorporated into our definition of reward for the purpose of showing in experiment 2 its effect on the performance of the Strategy Learner as a whole. However, currently for this experiment, the impact is set equal to 0, so it has no effect on the value placed for reward.

Part II: Experiment 1 - Manual Strategy Trader vs Strategy Learner Trader

The first task that Q-Learning based Strategy Learner had to complete was to measure its performance on the same date range of stock prices of JPM (In-Sample) that the Manual Strategy Trader attempted in the previous assignment. The goal of this experiment was to see how a reinforcement learning based trading strategy would fair against a manual rule-based strategy when given the same set of technical indicators. In addition, both strategies were tested against a benchmark strategy to measure relative performance. This benchmark strategy is outlined as follows: on the first valid trading day within the range Buy 1000 shares of JPMand hold onto it for the duration of the trading range (doing nothing else in between) and then Sell those 1000 shares on the last day. Finally, the conditions for this experiment are (1) commision = 0, (2) impact=0, (3) starting cash =$100,000.00, (4) allowable positions = [Long 1000 shares, Short 1000 shares, 0 shares].

2.1 Manual Trading Strategy

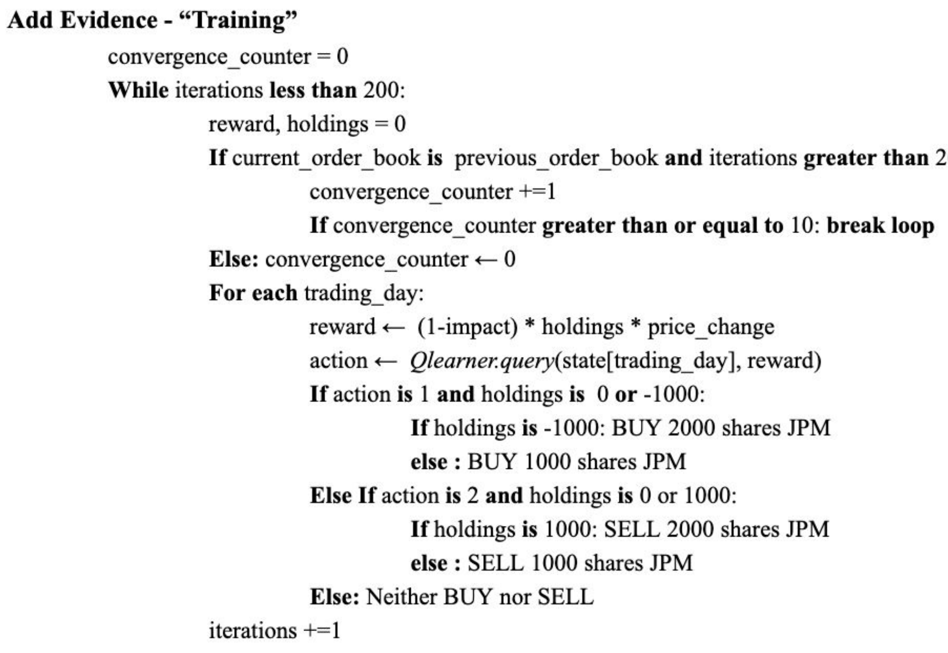
As a recap, the algorithm below represents the trading strategy employed by the Manual Strategy. The general approach was to use the Ichimoku Cloud to determine a good entry point to start trading, then use the Price-to-SMA Ratio and Bollinger Band Position to measure volatility of the stock price, and then finally confirm with the MACD divergence to see if this point is post-reversal. Accordingly, restrictions were set in place so that the trader couldn’t keep selling or buying consecutive days in a row.



```
For each day in trading_range:
    If price is above or below ichimoku_cloud and yesterday_position is not Out:
        If price_to_sma or bb_pos is Buy and yesterday_position is not Long:
            If  (yesterday_position is Short): Enter a Long position (Buy 2000 shares of JPM.)
            Else: Enter a Long position (Buy 1000 shares of JPM)
            yesterday_position ←  Long'
        Else If (price_to_sma or bb_pos is Sell and yesterday_position is not Short):
            If (yesterday_position is Long): Enter a Short position (Sell 2000 shares of JPM)
            Else: Enter a short position (Sell 1000 shares of JPM)
            yesterday_position ←  Short
```

2.2 Reinforcement Learning Trading Strategy

Q-learning is regarded as a model-free instance-based reinforcement learning algorithm. However, in the context of stock trading, having the agent start from zero information and then exploring and exploiting states every time you use it can be very costly (losing money when making bad trades). Thus, it would be useful to tweak the Q-learning agent to be able to utilize a previously developed Q-table on past data so as to make better decisions on current and future data (similar to applying a trained model on a test set). So, for the Strategy Learner trading agent the general approach was to train a Q-Learner by developing an "optimal" Q-table from the in-sample date range, and then use that same Q-table (and indicator bin thresholds) to determine trading decisions for a separate out of sample date range of JPM stock prices. The optimal Q-table was defined as one which resulted in an order book that led to the maximum expected return (optimal policy). The algorithms below represents this process of developing the trained Q-table and applying it. Note: convergence here is defined as when the order book has not changed for ten or more consecutive iterations after the 20th iteration.

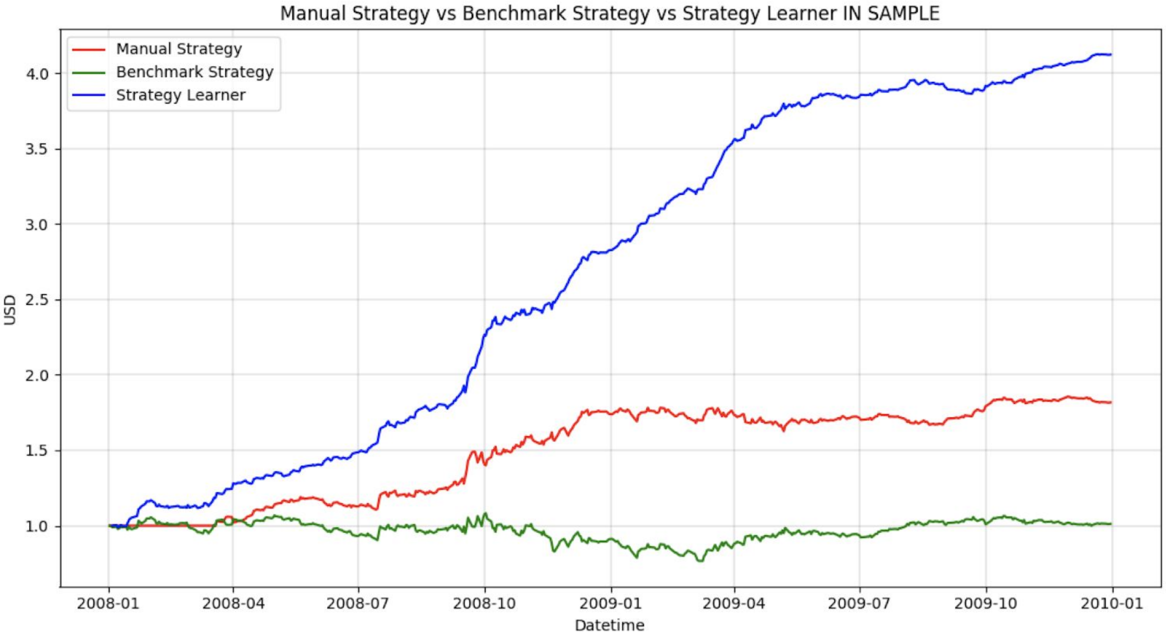


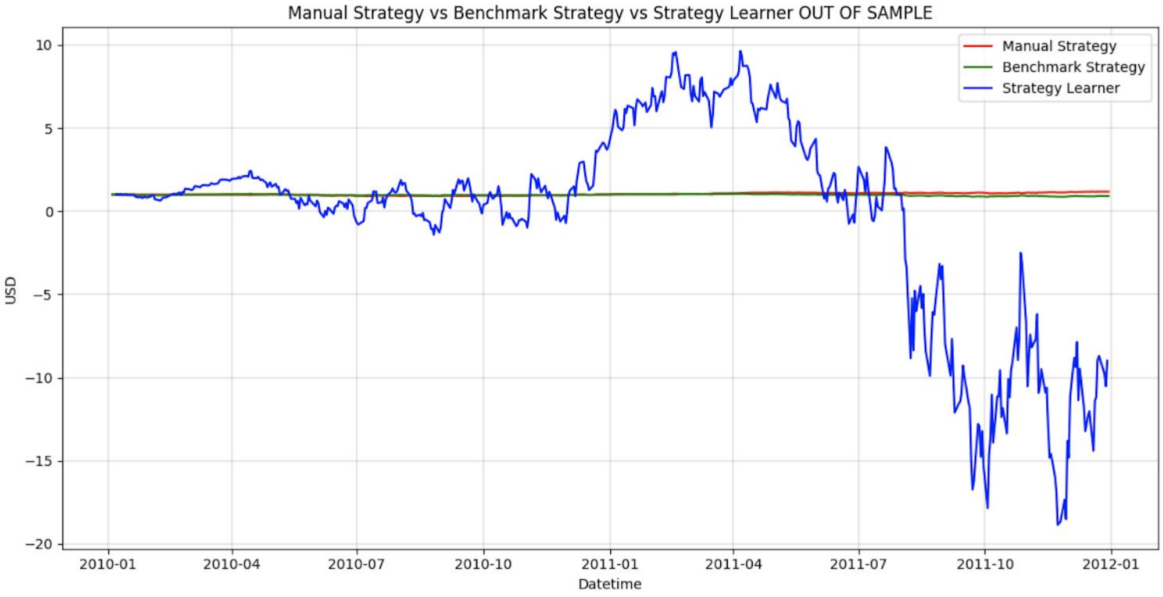
```
Add Evidence - "Training"
    convergence_counter = 0
    While iterations less than 200:
        reward, holdings = 0
        If current_order_book is  previous_order_book and iterations greater than 20:
            convergence_counter +=1
            If convergence_counter greater than or equal to 10: break loop
        Else: convergence_counter ← 0
        For each trading_day:
            reward ←  (1-impact) * holdings * price_change
            action ←  Qlearner.query(state[trading_day], reward)
            If action is 1 and holdings is  0 or -1000:
                If holdings is -1000: BUY 2000 shares JPM
                else : BUY 1000 shares JPM
            Else If action is 2 and holdings is 0 or 1000:
                If holdings is 1000: SELL 2000 shares JPM
                else : SELL 1000 shares JPM
            Else: Neither BUY nor SELL
        iterations +=1
```

```
Test Policy
    holdings = 0
    For each trading_day:
        action ← Qlearner.querysetstate(state[trading_day], reward)
        If action equals 1 and holdings equals 0 or -1000:
            If holdings equals -1000: BUY 2000 shares JPM
            else : BUY 1000 shares JPM
        Else If action equals 2 and holdings equals 0 or 1000:
            If holdings equals 1000: SELL 2000 shares JPM
            else : SELL 1000 shares JPM
        Else: Neither BUY nor SELL
```

2.3 Trading Results

The following graphs and charts show the performance of the Manual Strategy, Benchmark Strategy, and Strategy Learner (Q-Learner) on both in-sample and out-of-sample stock price data for JPM.





	Benchmark In-Sample	Manual In-Sample	Strategy Learner In-Sample	Benchmark Out-Sample	Manual Out-Sample	Strategy Learner Out-Sample
Cumulative Return	$1,230.00	$81,700.00	$321,412.00	-$8,340.00	$17,670.00	-$1,186,620.00
Average Daily Return	$16.81	$124.03	$280.91	-$13.72	$35.10	-$120,900.33
Std Dev of Daily Return	$1,700.44	$1,049.10	$771.91	$848.10	$741.83	$2,217,925.24

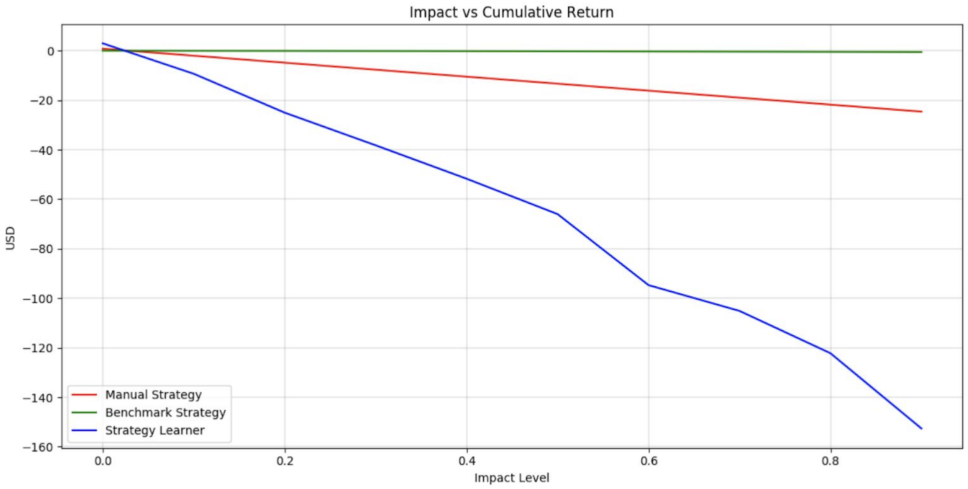
The in-sample the Strategy Learner accomplished a cumulative return of $321,412 compared to the $81,700 of the Manual Strategy (more than 3x the amount). Accordingly, the Strategy learner performs much better than both the benchmark and the manual strategy throughout the entirety of the in-sample date range. In addition, Strategy Learner has the highest average average daily return and lowest standard deviation of daily returns (most consistent) of the three strategies in-sample. However, we must note that this is completely opposite to its performance out of sample (in which it performs orders of magnitudes worse than that of benchmark and manual strategy). This outstanding performance in the in-sample data and terrible performance in the out-of-sample data point to the likely conclusion that the Strategy Learner is overfitting to the in-sample date range (finding more of the nuances of the data partly due to its ability to map many more potential states than manual strategy). Because the indicators used are primarily representations of momentum and volatility, if the prices during the training period are much more volatile with steep increases and decreases but not so during the testing period, then this could explain its incredibly poor performance.

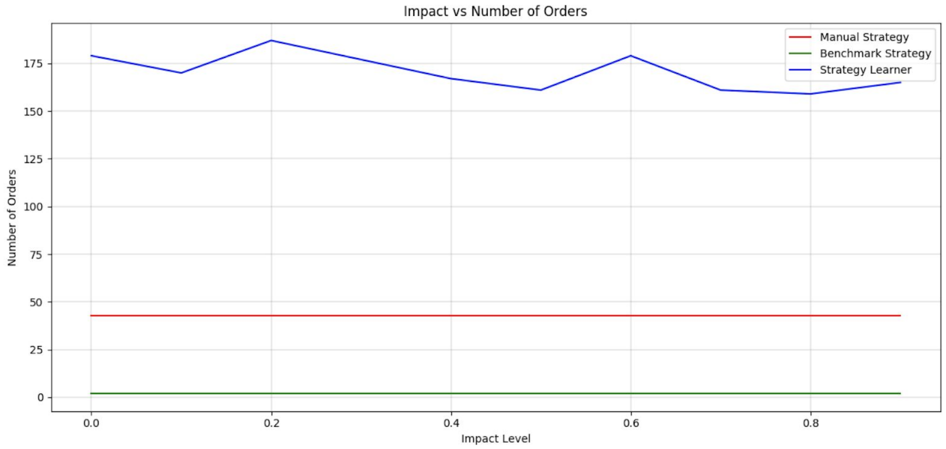
Part III: Experiment 2 - The Effects of Impact

Market Impact can be defined as the extent to which the transaction (buy or sell) moves the price of the stock against the trader who is buying or selling (ranging from 0 to 1). Previously, we traded with impact=0, making the assumption that on each trade the market is not affected by the participant whatsoever. Accordingly, the second experiment the Strategy Learner was expected to complete surrounded the notion of the effect that the Market Impact has on the behavior and performance of the Strategy Learner. In addition, I chose to measure the results in terms of 1) Cumulative Return and 2) The Number of Orders Placed (trade volume). I hypothesize that as the impact increases from 0 to 1, the

Strategy Learner will show an overall decrease in cumulative return as well as decrease in Number of Orders Placed. I believe this will be the case because as the impact value increases, it would 1) lower the reward for our strategy learner (defined above), and 2) increase the cost per transaction (order placed). In turn, I predict that this increase in transaction cost would push the Strategy Learner to trade less often (decreasing the trade volume).

3.1 The Effect of Impact on Cumulative Return & Number of Orders Placed





As you can see in the graphs above, the Strategy learner has a steady decline in cumulative return as impact increases. However, it is interesting to note that Strategy Learner decreases at a steeper slope than that of the manual strategy, which can possibly be explained by its inclusion of impact in not only its calculation of its portfolio values (and thus cumulative return), but also in its reward calculation. Accordingly, in terms of number of orders, it is surprising to see that the number of orders of Strategy Learner does not, in fact, monotonically decrease as impact increases (our hypothesis). However, it is interesting to see that the trade volume for Strategy Learner seems to oscillate as the impact level increases. We would expect that as the market increasingly moves against our agent that it would eventually choose not to trade at all, however, this was not the case. Accordingly, we can also see that the manual strategy trade volume is a consistent straight line because its trading strategy is not based on the impact whatsoever, and thus remains unaffected.