A Comparison of Q-Learning and Classification Methodologies to Human Behavior in Stock Trading

Sarvesh Patki & Eugenia Fomitcheva
New York University – Center for Data Science
ssp6603@nyu.edu, edf257@nyu.edu

ABSTRACT

Our goal is to understand how in the stock trading domain humans make buy, sell and hold decisions as compared to two machine learning algorithms – a Q-learning agent and a classification-based agent. Through conducting a behavioral experiment we aim to identify the nature of decisions made by humans and attempt to quantitatively and qualitatively assess the comparability of ML methods and human results to analyze how people versus machines approach the decision-making process.

Keywords: *stock trading, reinforcement learning, classification, Q-learning*

I. INTRODUCTION

Trading and the stock market have long been a topic of interest for many, from retail investors such as ourselves to professional traders and quants. Over 50% of Americans invest in the stock market either directly or indirectly through mutual funds and retirement plans. While most of those who invest in the market do so passively by exercising a long-term buy and hold strategy, others actively manage their investments and rebalance their portfolios to maximize profits.

There has already been a substantial amount of research and work done in the field of reinforcement learning and how it pertains to stock trading. The majority of the works reviewed for the purposes of this paper focus on developing Q-learning agents with the goal of optimizing a portfolio. The reinforcement learning agent performs timed 'buy,' 'sell,' and 'hold' decisions to make a profit. Ultimately, the agent's performance is evaluated against other trading strategies. What we found to be a common missing element in all of these studies was any comparison to human trading behaviors. Thus, we found it would be interesting to bring in a human behavioral study component to better understand whether these models share elements of human cognition and explore the motivations influencing these decisions.

In this project, we aim to further supplement prior research in the field by analyzing and comparing decisions made by human participants in a survey alongside the results of a reinforcement learning agent and a decision tree classifier. We summarize the decisions made by each of these and discuss the comparability of our findings, as well as the observed behaviors that ultimately set human and machine learning trading apart.

II. RELATED WORK

Since the advent of machine learning and data science, fintech has been an active area of interest for many machine learning practitioners and researchers. In this space, predictive pricing in the stock market has attracted the attention of large financial corporations, AI startups, and scholars. While relatively simple techniques such as regression can be used to predict stock prices, recently more advanced methods such as autoregressive integrated moving average (ARIMA), artificial neural network (ANN), and long short-term memory (LSTM) models have entered the spotlight of this domain.

A common advantage of these models is their limited need for underlying assumptions about the data. ANNs and LSTM models especially are adaptive in nature, making them great candidates for approaching non-linear problems[5]. In our literature review, we found that the aforementioned methodologies have primarily been used for explicit price prediction with less consideration given to price trends and patterns which comprise equally as valuable information when it comes to optimizing trading strategy.

This observation prompted us to investigate a slightly different approach – one interested in making decisions and taking corresponding actions with the goal of achieving optimized returns in the market. We do so by looking at i) a reinforcement learning (RL) approach and ii) a combination of RL + classification-based approach. Ultimately, we were interested in understanding whether these approaches could achieve similar decisions to those taken by humans in a stock trading experiment.

Between reinforcement learning and classification we hypothesized RL may more closely represent how humans make trading decisions as most people likely have the experience or at least understanding of trying to gain a reward through investing. Furthermore, trading decisions are not easily labeled, making this a challenging supervised learning problem. Whether one should have bought, sold, or held a stock is understood only in retrospect due to the time-series nature of the data and depends on factors such as investment horizon, and risk appetite among others.

III. METHODS

(i) *Data*

The data used in the behavioral experiment was stock price data for IBM (ticker: IBM) and Microsoft (ticker: MSFT) for the time frame 12/29/2018 – 12/31/2019. We considered 2018 as a ‘training’ year and 2019 as a ‘test’ year.

For the reinforcement learning and classification models, the time frame used was extended to 12/31/2009 – 12/31/2019 with the first 9 years used as training and 2019 again used as test data.

(ii) *Behavioral Experiment*

We asked participants to maximize profit (and incentivized performance with a $15 gift card) by making buy/sell/hold decisions for a series of stock price graphs for IBM and MSFT. In order to prevent any plausible biases, we kept the company names undisclosed. This also ensured that participants made decisions based on the data shown rather than any external information. A reinforcement learning algorithm deployed for the same task would formulate a policy and make decisions (take actions in ML jargon) based on the stock data alone (i.e. the states).

The survey was divided into four sections, two for each stock, with four questions in each section. Participants were first shown a graph for Q1 (first quarter) data and were asked to make a buy/sell/hold decision. For the subsequent questions, participants were shown graphs containing cumulative stock price data (i.e. Q1+Q2 and so on) and asked to answer the same question. Each decision was applied to only one share of stock and participants were only allowed to sell the stock if they had previously purchased it in the same section.

(iii) *Reinforcement Learning Agent*

A reinforcement learning algorithm was used to similarly model buy/sell/hold decisions as in the behavioral experiment. We opted for a Q-learning agent with which we could discretize the space into finite states, a necessary condition for reinforcement learning. The agent was also passed parameters such as number of episodes, number of states, a discount rate (gamma), and a learning rate (alpha).

The agent took actions in order to maximize the portfolio’s profit at the end of the episode. Thus, we computed the profitability as a means to understand the agent's reward over time:

$$Profitability \ = \frac{Total\ capital\ at\ the\ end}{invested\ capital} \ - \ 1$$

Equation 1: Profitability metric

Three major steps involved were as follows:

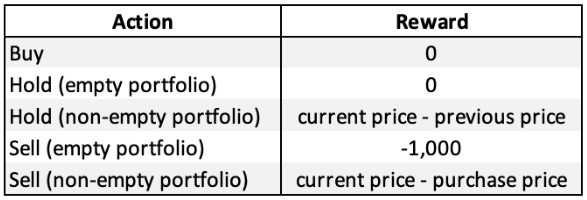
(i) *Fetching and preprocessing data*

Historical financial stock price data for IBM and MSFT was pulled from Yahoo finance for 12/31/2009 – 12/31/2019. We split the data in the ratio 0.9 : 0.1 for train-test without permuting the data, thus keeping the sequential nature intact. This ensured that the test periods for both the humans and the RL agent were the same. Although the training period for humans was just one year as compared to 9 years for the RL agent, we assumed the demographic that was surveyed (ages 21-49) had prior experience in the stock trading domain. To achieve the same level of training as humans, the RL agent was trained on a larger dataset.

We preprocessed the data by normalizing the adjusted close price for each trading day with respect to the first available trading date. Next, we discretized the normalized values to integer states by dividing the ordered values into equally sized chunks and representing each chunk with an integer. It is worth noting that during inference the test data is discretized in the same manner. We decided to use only one predictor, the adjusted close price, in order to as closely match the data provided in the behavior experiment to humans.

(ii) *Setting up the environment, defining actions and states and training the agent*

The Q-learning algorithm requires information about the current state of the agent, the possible actions that can be taken, and the rewards associated with these actions. The Q-table can be interpreted as an '*s* x *a*' matrix where the rows represent states and the columns represent actions. We have defined the number of states to be six per findings in our literature review[1]. The number of actions that the agent can take is three, corresponding to buy, sell and hold. However, the rewards associated with each action can differ depending on whether or not the portfolio is empty, as illustrated in the table below. Since we restricted selling in the behavioral experiment unless a participant previously bought stock, we assigned a high penalty to the agent for selling on an empty portfolio as a means to disincentivize the action.



Action	Reward
Buy	0
Hold (empty portfolio)	0
Hold (non-empty portfolio)	current price - previous price
Sell (empty portfolio)	-1,000
Sell (non-empty portfolio)	current price - purchase price

Table 1: RL agent actions and rewards

The agent acts based on the action policy. The action can be randomly chosen or can be based on the current state in the Q-table. Initially, the agent is allowed to explore more (by setting a high threshold value) but as the agent learns actions that increase its reward during training, the threshold is lowered and the agent is more likely to exploit the already known actions. Based on the actions taken during the training phase, the Q-table is appropriately updated using the Bellman Equation, illustrated below.

$$New\ Q(s,a) \ = Q(s,a) \ + \boldsymbol{\alpha}[R(s,a) \ + \boldsymbol{\gamma} maxQ'(s',a') \ - \ Q(s,a)]$$

– New Q(s,a): new q-value for (s,a) pair
– $\boldsymbol{\alpha}$*: learning rate*
– R(s,a): reward for taking action 'a' at state 's'
– $\boldsymbol{\gamma}$*: discount rate*
– maxQ'(s',a'): maximum expected future reward given the new state s' and all possible actions at the new state

Equation 2: Bellman Equation

(iii) *Tuning the hyperparameters to represent the decisions made by the humans*

We trained the algorithm across various values of alpha, gamma, number of states, and number of episodes. As mentioned previously, we set the number of states to six. The remaining parameter values that most closely resembled human decisions from the survey are illustrated in the results section.

(iv) *Classifier built on RL agent labels*

In an effort to approach stock trading from a different perspective we opted to consider it as a classification problem rather than a reinforcement or active learning problem. However, as previously mentioned, we must note that as classification is a supervised learning concept and our data was not explicitly labeled (i.e. we did not have daily buy/sell/hold tags for the stocks), we lacked a crucial element for this approach. Thus, we used the output of the RL agent as labels to the classifier in the interest of understanding whether this approach could learn to differentiate trading actions.

To build this classifier we used the following historical data from yahoo finance for IBM and MSFT: date, open, close, adjusted close, and volume. From these, we engineered a few additional features:

1. Simple moving average (20-day window)
2. Standard deviation (20-day window)
3. A band defined by the simple moving average +/- the standard deviation
4. Up or down movement from prior day close to current day open
5. Distance from mean computed as the adjusted close – simple moving average

We then scaled the data using sklearn's MinMaxScaler in the preprocessing library and fit a decision tree classifier to our training data. The model was evaluated by computing the accuracy and precision of predicted labels versus the RL agent's output.

IV. RESULTS

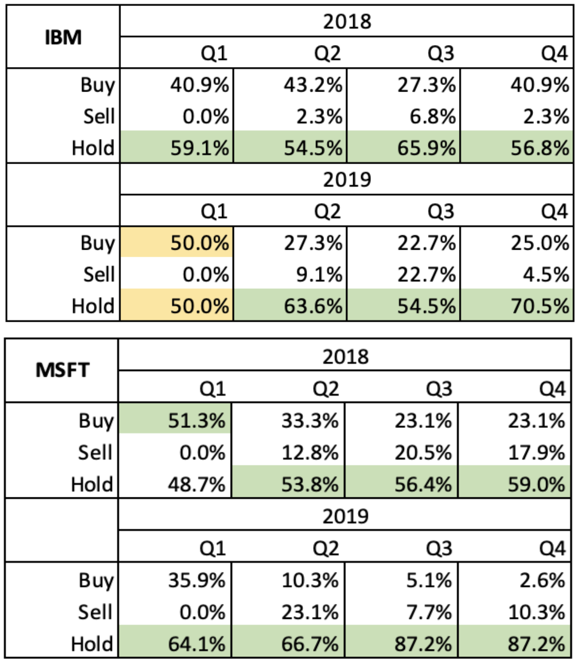
(i) *Behavioral Experiment – Participant Data*

The responses of 44 participants, broken out across buy/sell/hold for each quarter-end are delineated in the tables below. The majority vote for IBM and MSFT in the simulated test year 2019 was hold.

Beyond the overall counts we also identified the series of actions each participant took as one of five patterns:

1. Nothing: only hold actions
2. Buy & Hold: buy and hold for long-term
3. Buy & Buy: buy several times
4. Buy & Sell: buy and sell once
5. Buy, Sell, Buy: repurchase previously bought stock

Participant Responses

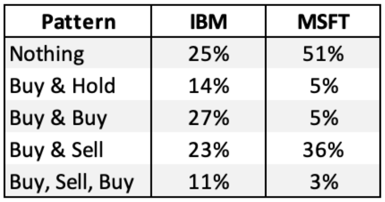


IBM	2018			
	Q1	Q2	Q3	Q4
Buy	40.9%	43.2%	27.3%	40.9%
Sell	0.0%	2.3%	6.8%	2.3%
Hold	59.1%	54.5%	65.9%	56.8%
	2019			
	Q1	Q2	Q3	Q4
Buy	50.0%	27.3%	22.7%	25.0%
Sell	0.0%	9.1%	22.7%	4.5%
Hold	50.0%	63.6%	54.5%	70.5%

MSFT	2018			
	Q1	Q2	Q3	Q4
Buy	51.3%	33.3%	23.1%	23.1%
Sell	0.0%	12.8%	20.5%	17.9%
Hold	48.7%	53.8%	56.4%	59.0%
	2019			
	Q1	Q2	Q3	Q4
Buy	35.9%	10.3%	5.1%	2.6%
Sell	0.0%	23.1%	7.7%	10.3%
Hold	64.1%	66.7%	87.2%	87.2%

Tables 2 & 3: All actions taken by participants in the survey for IBM and MSFT stocks

Response Patterns



Pattern	IBM	MSFT
Nothing	25%	51%
Buy & Hold	14%	5%
Buy & Buy	27%	5%
Buy & Sell	23%	36%
Buy, Sell, Buy	11%	3%

Table 4: Breakdown of action patterns identified in the survey

We see that for the instance of IBM the action patterns were rather evenly distributed – for 2019 IBM's stock had a predominantly flat trend. Meanwhile, with MSFT the majority either took no buy/sell actions or bought and sold once – MSFT's stock had an upward trajectory in 2019.

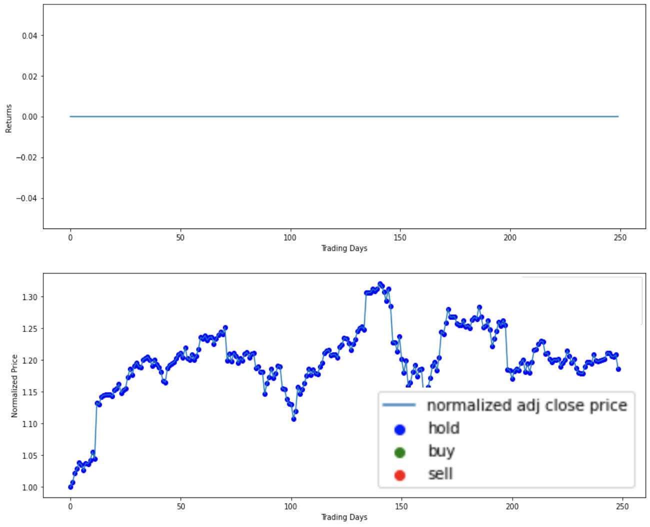
(ii) *Reinforcement Learning Agent*

The hyperparameters which produced model results most similar to human response are included below along with outputs representing the series of actions taken by the RL agent.

Parameter	IBM	MSFT
alpha	0.35	0.35
gamma	0.95	0.95
no. states	6	6
no.episodes	8	8

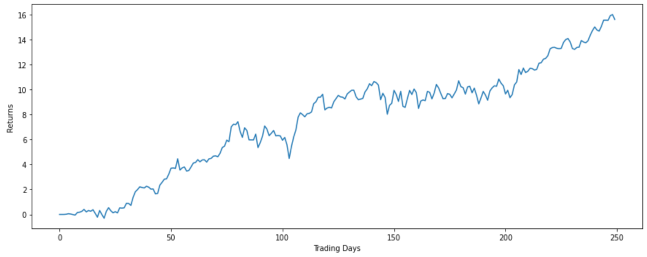
Table 5: Hyperparameters to RL agent

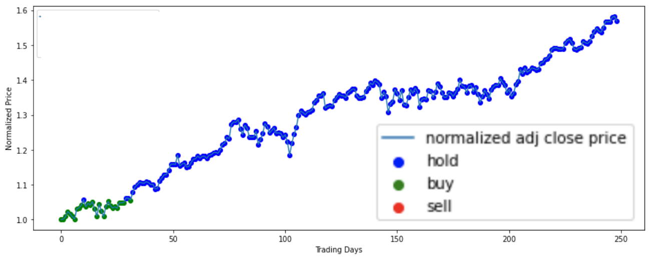
Actions taken by agent on IBM – hold only



Graphs 1 & 2: Reward over time (top) & Actions taken (bottom)

Actions taken by agent on MSFT – buy & hold





Graphs 3 & 4: Reward over time (top) & Actions taken (bottom)

(iii) *Decision Tree Classifier*

Below are the confusion matrices for a decision tree classifier with maximum depth = 5 on IBM and MSFT.

IBM Classification Results – accuracy = 0.24

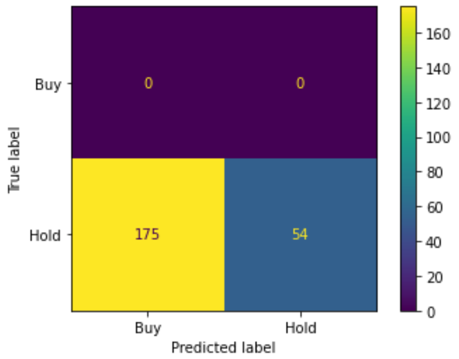


Chart 1: IBM Confusion Matrix for DT Classifier

MSFT Classification Results – accuracy = 0.04

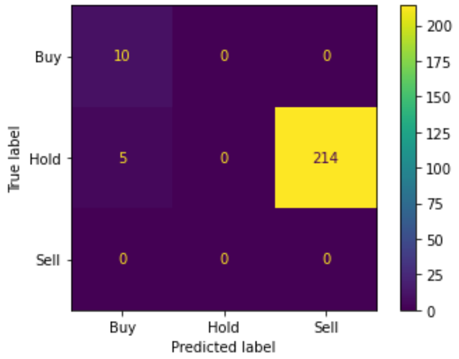


Chart 2: MSFT Confusion Matrix for DT Classifier

V. DISCUSSION

Our findings are interesting when viewing the results of the survey, Q-learning agent, and classifier side-by-side. As we hypothesized, our data illustrate that we can perhaps more closely replicate human decisions in trading through reinforcement learning methods in comparison to classification methods.

In the survey, participants were asked to make trading decisions in order to maximize their returns similar to how one would approach this problem in the real world. While the survey did not explicitly reward or penalize the participant after every quarter-end, the stock price data itself for the following quarter provided some form of feedback. Thus, we can view this experiment as mimicking a reinforcement learning framework with a set goal and policy, changing environment, and a reward system. The most challenging aspect of RL to approximate is certainly the Q-function which, as we saw, required mapping of time series data to discrete state space.

To achieve results similar to those of humans we adjusted a variety of hyperparameters (as illustrated in the methods section). We first increased the number of episodes as this allowed the agent more ‘time’ to learn. Typically the agent learns better with a greater number of episodes, although this increases the computational cost. We then set the alpha value which represents the learning rate/threshold used in RL algorithms to 0.35 to prevent non-convergence. The gamma value, which represents the discount factor for future rewards, was set to 0.95. The closer this value is to one, the more similarly the agent values future and present rewards. If closer to zero, the agent values only immediate rewards. Therefore, a gamma of 0.95 suggests that participants in the survey did significantly weigh the long-term when making trading decisions.

Beyond training and testing the RL agent sequentially for first IBM and then MSFT, we decided to see whether the agent was able to generalize by training on one stock while testing on the other. The result of this cross-testing on the stocks led to some variations in the trading strategy. Namely, for IBM the actions taken by the agent trained on MSFT were the same as those shown in the results while for MSFT the agent trained on IBM opted to buy the stock in the latter half of the year.

We found this behavior to be unsurprising as a strategy that works for one stock may not work for another because of a myriad of non-technical, fundamental, and external factors that might come into play. This suggests that the RL agent learns to optimize its strategy for a particular stock (and each stock has its own environment) and is not readily able to generalize to all. Further investigation as to the accuracy of larger-scale predictive pricing models is a separate topic for discussion which is beyond the scope of this project.

In the behavioral experiment conducted we observed not only the majority action taken in 2019 but also the pattern of decisions taken, as illustrated in Table 4. Notably, for IBM, whose stock maintained a flat trend in 2019, the response patterns were distributed across a variety of approaches. The most common pattern, buy & buy, demonstrates that participants repurchased stock when the price went below a threshold at which they previously bought. This behavior suggests these participants were expecting to see a change in the stock’s trajectory and were reluctant to accept an immediate loss. It also hints toward a long-term investing strategy by the participants, where they were willing to buy the stock when the price dipped. This lowers their average buy-in price and would give a higher return if the stock price went up.

The distribution of patterns we observed for MSFT on the other hand was primarily divided between the nothing and buy & sell patterns. Based on the results, we can interpret that those who took no action saw only a consistent upward trajectory in MSFT’s stock price and felt that they had missed the buying opportunity, expecting a shift to a downward trend eventually. The other group of individuals elected to buy the stock at the beginning and sold after achieving some positive return, with many citing a 10-15% gain threshold in their written responses. This follows a well-known trading pattern known as the *disposition effect* – the tendency to sell stocks that have appreciated and are therefore considered to be 'winners' as compared with those that have declined, or 'losers'[2].

A confine in our chosen domain which we have previously alluded to is the lack of ground truth or labels for the dataset. These greatly vary based on factors such as individual risk appetite, investment horizon, etc. Through various methods, we can attempt to construct these labels but we need to also be cognizant of their extreme variability. We could take the majority equity analyst buy/sell/hold recommendation (i.e. sources such as Benzinga, Refinitiv, etc.) or combine the classifier with another machine learning algorithm as we did so in feeding the RL agent's outputs to the classifier. Nonetheless, stock price data poses many challenges as it is not only a function of time and technical features but also a product of company performance and external economic and geopolitical factors.

Given these constraints we found it challenging to model trading decisions (buy/sell/hold) as a classification problem. This was due to a lack of ground truth labels which are necessary for supervised machine learning algorithms. In experimenting with a decision tree and random forest classifier we obtained consistently poor results from an accuracy perspective. However, we found it notable that the classifier grouped data points with the same true label together but failed to label the entire group correctly. For instance, the decision tree classifier on MSFT identified that 214 points (93% of the data) were of the same 'type' but predicted them all as 'sell' when the true label was 'hold'. In future explorations, it could be worth designing more features or augmenting the data in order to better understand whether the classification approach has validity.

VI. CONCLUSION

Based on the results obtained from our two approaches we can see that given the limitations of the models and behavioral experiment, reinforcement learning did a better job of representing the survey results versus the classifier. Moreover, due to the various hyperparameters in the Q-learning RL agent, we can better understand how manipulating them impacts the model's behavior. Overall, reinforcement learning in this case study provided for better interpretability.

From our analysis of human response in the stock trading experiment we found that while holding the stock is the most commonly exercised option, more holistically, the trading approach taken differs depending on the stock's current trajectory and future expectations.

VII. FUTURE SCOPE

In future work in this field, it would be worthwhile to expand the set of stocks analyzed and the time frame to gain a better understanding of trading behavior in various environments. Stocks from different industries and over different time periods can be chosen.

In the early stages of this project, we considered A/B testing participants in the survey by presenting the visual data in two different formats (line graphs and candlesticks) and comparing decisions made between the two groups. We still view this experiment as one of interest although believe that drawing a comparison between human results and a DQN agent or CNN built on a visual graphical input might be more salable. Furthermore, in this paper, we discussed the results for only one type of reinforcement learning agent. In further exploration, analyzing different agents and their relative performance compared to human behavior could be of interest.

REFERENCES

[1] Chakole, J., Kolhe, M., Mahapurush, G., Yadav, A. and Kurhekar, M., 2021. A Q-learning agent for automated trading in equity stock markets. Expert Systems with Applications, 163, p.113761.

[2] Frydman, C. and Camerer, C., 2016. The Psychology and Neuroscience of Financial Decision Making. Trends in Cognitive Sciences, 20(9), pp.661-675.

[3] Li, Y., Liu, P. and Wang, Z., 2022. Stock Trading Strategies Based on Deep Reinforcement Learning.

[4] Bao, D. and Yang, Z., 2022. Intelligent stock trading system by turning point confirming and probabilistic reasoning.

[5] Ma, Q., 2022. Comparison of ARIMA, ANN, and LSTM for Stock Price Prediction.

[6] Liu, Q., 2022. GitHub - nyxcat/stock-trader. [online] GitHub. Available at: <https://github.com/nyxcat/stock-trader> [Accessed 11 May 2022].