CSE 6242 – Data and Visual Analytics

CryptoPunks Final Report

INTRODUCTION

Over the last couple of decades investing has become more accessible and simpler than ever. Multiple platforms such as Robinhood, Webull, and Etoro allow retail investors to directly invest in multiple assets such as common stock and more recently virtual currency, also known as cryptocurrency, without having to go through a traditional broker. Cryptocurrency has been a hot topic for the last decade facilitating over $1.8 trillion in trading volume in 2021. This increased interest in crypto paired with the increased use and influence of social media has peaked attention to the influence of social media sentiment on trading. Research studies show that social media sentiment typically correlates positively with stock and cryptocurrency price action, as witnessed during events such as the price squeeze of common stocks such as GameStop, and cryptocurrencies like Dogecoin over the last 12 months. Financial institutions, such as hedge funds, have been using social media sentiment to their advantage to provide investors with increased capital gains and attract customers. These financial institutions are known for using machine learning algorithms for high frequency trading that use multiple indicators, including social media sentiment, to beat the market. This data, however, is not readily available to retail investors in a way that can be used to improve their portfolio. It is evident that the new wave of young retail investors is persuaded and driven by social media's reaction to trends. These quantifiable trends can be transformed into tangible data, which will add value to strategic analysis carried out by technical investors.

PROBLEM DEFINITION

The average investor does not have the expertise or understanding on how to build a machine learning algorithm for automated (and potentially high frequency) trading. Additionally, social media sentiment has shown to have a correlation with the market price of a stock or cryptocurrency, however obtaining and analyzing the sentiment in this data is not readily available for retail investors. This project aims to achieve three main objectives: 1) analyze and quantify Bitcoin Twitter sentiment 2) combine the quantified Twitter sentiment with other commonly used stock price indicators that can be used to train a machine learning trading algorithm and 3) provide an interactive user interface that will allow the common retail investor to train and test a trading algorithm that will provide them with a better informed recommendation on when to buy, sell, or hold Bitcoin.

SURVEY

One common way to perform Quantitative Market analysis is using historical data to train a model using reinforcement learning in the form of Q-learning. By leveraging indicators based on trend, volume, momentum and volatility, a deep learning model can be used to learn how to trade based on a reward system [5]. A more novel architecture of Q-Learning is found in the following paper that introduces a layer of time-varying linear models [7]. Reinforcement learning, a more robust mathematical form of Q-learning, was explored in the following paper to predict stock prices and had better accuracy than Q-Learning models, but also required significantly more training time [11]. High frequency and automated trading, including but not limited to reinforcement learning models, have been utilized by hedge funds successfully for the last decade [19].

Our project requires us to analyze and quantify social media sentiment. An association was found in a study between tweet sentiment and stock returns [16]. Therefore, we hypothesize that this association applies to crypto stock and supports why sentiment analysis of tweets may be a promising predictor. Several concepts such as NLP, Neural Networks, social media analysis, and text mining can be used with sentiment analysis [12][15][22]. As we are seeking to mine data from Twitter, sentiment analysis techniques such as lexical-based word mappings to sentiment scores and classification-based approaches should be considered, though the latter are usually more accurate [20]. Other useful approaches for Twitter analysis not involving scoring sentiment include relative mention volume to determine the mention rate.

Social media sentiment has been used for analysis for cryptocurrency trading and has a proven correlation to trading price [1][2][14]. Several references provide valuable insight via statistical methods that create a sentiment heuristic [1][2], along with modern cryptocurrency price prediction approaches [16]. Additionally analyzing the type of user making comments, i.e., a follower, influencer, anonymous, or troll has shown to be impactful as well [1][14]. One can also use sentiment analysis specifically to correlate volatility of cryptocurrencies such as Bitcoin to Twitter semantics using a semantic embedding architecture called VADER [1]. [17],[3], and [10] demonstrate the effectiveness of using LDA Topic Modeling, Long Short-Term Memory models, and Gradient Boosting Tree, respectively, to predict future cryptocurrency prices based on sentiment. These can serve as comparisons when measuring the effectiveness of our chosen Q-Learning or Grid Search approaches. Some of the difficulties encountered in sentiment analysis [3] were accurately classifying neutral and negative sentiment and the influence of twitter bot posts which we will need to consider when classifying and handling our sentiment data.

PROPOSED METHOD

Intuition: Our work stands out from the state of the art, because we have developed an interactive, customizable approach based on machine learning that helps any type of user trade the cryptocurrency bitcoin. The interactive visualization dynamically trains and tests the Q-learning or Grid Search model. The use of sentiment analysis on Twitter data to create cryptocurrency buy and sell signals is an advanced feature that financial institutions have created and utilize exclusively and can rarely be found open to the public in trading platforms. This project is a solution to investors/users who want access to such technology without the need to know the complexities of the algorithms.

Data Prep/Pipeline: The data preparation was performed using Pyspark/Spark SQL as it allows us to use a transformation framework that can easily be scaled when using larger datasets [6]. The raw Bitcoin Tweets (4 GB of data spanning 5 years from May 2014 to November 2019) was retrieved in CSV format from Kaggle.com, and Bitcoin price data (7 years from May 2014 – February 2022) was retrieved from Yahoo Finance in CSV format. The data was transformed to add the following indicators: Average Twitter Sentiment, Price to Price Simple Moving Average (SMA) Ratio, Bollinger Bands, Bollinger Band Percentage, Moving Average Convergence Divergence (MACD), Stochastic Oscillator, On Balance Volume, Momentum

The pipeline accepts user inputs to calculate the above indicators. For example, the simple moving average of Bitcoin price can be calculated using a 7-day, 14-day, or 21-day lookback window and the Bollinger bands can be calculated using 1, 2, or 3 standard deviations as indicators of the market volatility [13]. Lastly, the NLTK Vader tool for sentiment analysis was wrapped in a Spark UDF and applied on the raw Twitter data [18]. The average sentiment was aggregated to a daily granularity and joined to the daily Bitcoin price. Lastly, the Spark data frame was then converted into a Pandas data frame for consumption by the downstream models.

Twitter Sentiment Analysis: A text tokenization tool known as NLTK's nltk.sentiment.vader module was chosen to classify and quantify sentiment in tweets [9]. Valence Aware Dictionary for Sentiment Reasoning (VADER) is a model used for text sentiment analysis that detects both polarity (positive and negative sentiment) as well as the intensity of emotion [1]. We chose VADER for analysis because it is pre-trained for analyzing sentiment and could easily be applied to Bitcoin Twitter data to quantify sentiment and use this value as a variable in our model for predicting when to buy, sell or hold Bitcoin data. Using 4 GB of Bitcoin-related tweets from May 2014 to November 2019, we calculated an average sentiment score for each day using the compound sentiment scores scaled between -1 and 1. The average sentiment score is taken from the compound sentiment score output by NLTK's sentiment analysis function, which is derived from the positive, negative, and neutral sentiment scores. To classify sentiment based on sentiment scores, we compared sentiment scores to thresholds: Scores greater than or equal to 0.05 are classified as positive sentiment, scores less than or equal to –0.05 are classified as negative sentiment, and scores between 0.05 and –0.05 are considered neutral sentiment.

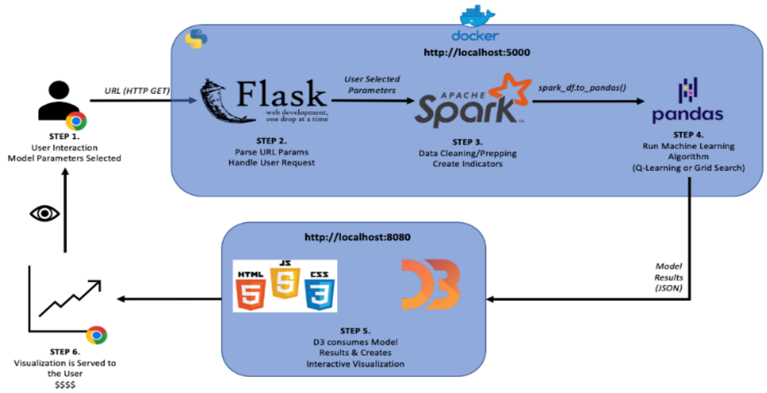
Q-Learner Prediction: Q-learner is a reinforcement learning method where an agent takes actions in an environment to maximize the reward. The Q-Learning technique is used to predict the stocks over a Partially Observable Markov at each of the three possible actions: Hold, Buy, or Sell to maximize the reward. We decided to use Q-learner because it does not require labeled data, and agents can learn and improve in real time. Thus, Q-learner can work well with less historical data [5]. After collecting stock data and sentiment data, we can frame our stock prediction as a Markov Decision Process (MDP) with states, actions, and rewards [7]. The Q-learner agents will be trained on the stock portfolio value. After Q-agent is trained, we can use the Q-learner table by sending stock data and sentiment value to the Q-agent, and Q-agent will pick action with highest Q value. All the training process, selected indicators by users, and portfolio results will be displayed and make it easy for user to select and visualize.

Grid Search Indicator Analysis Method: Grid Search Technical Indicator Analysis is a type of Optimization-based learner method that often combines with machine learning methods to predict future stock market prices. Those technical indicators reflect the market's behavior. Thus, by studying those indicators, we can develop a scan-based strategy using a set of rules and thresholds to decide when to Buy or Sell. The main goal is to determine the set of thresholds that optimized the portfolio return.

Visualization**:** Using the Data-Driven Documents (D3) JavaScript Library, along with the platform Observable, we have created a custom dynamic and interactive visualization that allows the user to explore the available data freely. The user is provided with the indicators listed in the "Data Prep/Pipeline" section above to select and include in the model. Additionally, the following hyperparameters are selected by the user as well: Alpha, Gamma, Random Action Rate (RAR), Random Action Decay Rate (RADR), Simple Moving Average (SMA) Threshold, Stochastic Oscillator Upper and Lower Limit, Momentum Threshold, Sentiment Threshold

Provided with the above set of indicators, the user can select a subset (or all) of the indicators desired to retrain the algorithm that will provide buy/sell signals as a result. The result will be a graph that compares the accumulated capital gains of the algorithm to a benchmark scenario. The benchmark scenario would be the gains accumulated from buying and holding over that same time period. The user has the freedom of tuning the algorithm with the desired features and additional hyperparameters such as the window lengths for individual indicators and upper/lower limits for those indicators that apply. Additionally, parameters such as the learning rate and decay rate for the Q-learner algorithm will be able to be adjusted to improve performance. To provide even another additional layer of exploration, the user is also able to define the training and testing periods. [4][8]21]

Application Flow



<u>Figure 1.</u> Illustration of the interactivity in application

<u>EXPERIMENTS:</u>

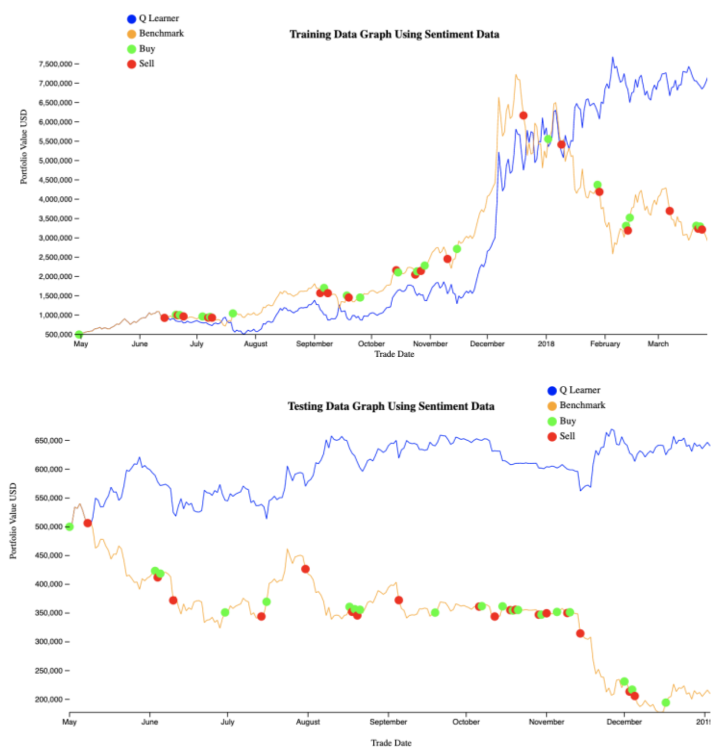
The visualization seeks to provide users with the ability to train the model with their selected indicators and hyperparameters to create a model they can use to determine when to buy, sell or hold Bitcoin. The visualization also provides users with the ability to compare how much a portfolio would be valued in comparison to the benchmark scenario of holding the Bitcoin without buying or selling it. For example, if someone invested $500,000 in Bitcoin at the start of the testing period (or training period if you choose to re-run the model on the data used to train it) and then held it for the duration of that period, how would the financial return (or loss) from that benchmark scenario compare to the amount of money made (or lost) if the same investor were to instead follow the buy, sell, or hold recommendations of the model? Lastly, the visualization also answers how the indicators selected by the user vary with time. Once the data preparation pipeline has been run to obtain the price and sentiment indicators the user has selected, user can select which of the two models they want to subsequently execute.

For the Q-learner method, the Q table is a table of states x actions and is initialized as a matrix of zeros. There are 3 actions: Buy, Hold, and Sell. The number of states depends on the number of user's input indicators. Each indicator is discretized into bins: buy, sell or do nothing. States are all combination of the indicator's bins. At each state, the agent selects an action that with the highest reward in previous experiences. The Q model is trained using the reward in form of the stock's profit and loss. For each action at each state, the Buy/Sell signal can add or subtract money from the portfolio. The input alpha value is the learning rate of the Q-agent. The Q table gets updated after every action and eventually converges to a best action path. Once the model completes the training, we can evaluate the Q-agent's performance by comparing the Q-agent portfolio vs. a benchmark position of buying X shares and holding.

Both Q Learner and GridSearch have two main functions: addEvidence and testPolicy. AddEvidence takes in the indicators and prices in the training dataset to learn the optimal parameters for trading. GridSearch uses a grid search optimization tool runs all the possible permutations of the thresholds and returns a combination that yields the highest portfolio return. We use the Scipy tool called scipy.optimize.brute. TestPolicy function takes indicator inputs, thresholds learned from addEvidence to return the list of Buy and Sell orders. To analyze the performance of the learner, we plot benchmark portfolio return against the model's portfolio return. Our benchmark is the case where the portfolio uses the user input's start value of cash, investing max number of shares, and sell it at the end period date.

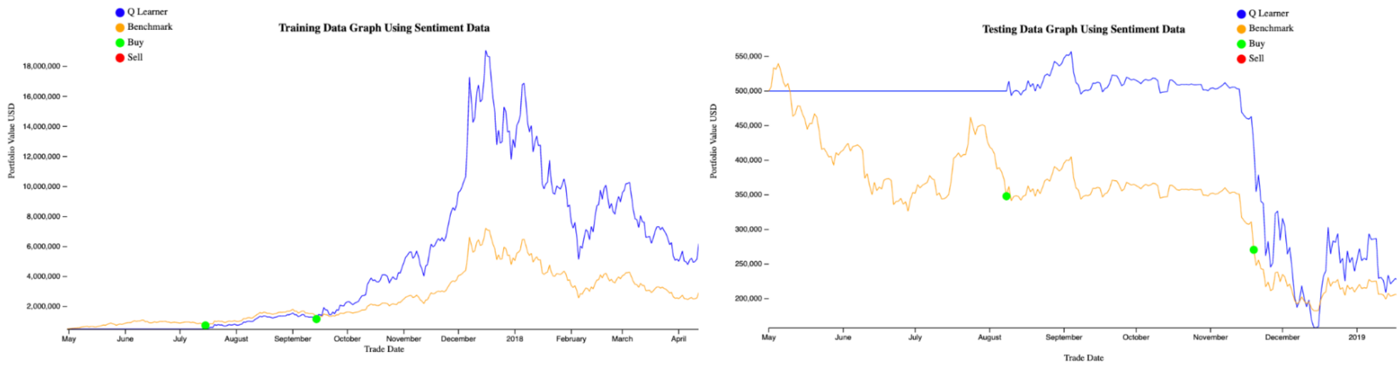
EVALUATION:

Figure 2 below shows Benchmark vs Q-Learner portfolio performance during training and testing period. Red and Green dots present the Buy and Sell signal that Q-Learner takes. Overall, Q-Learner yields higher return in both training and testing periods. To improve the accuracy and consistency of price prediction, Q-learner needs to be trained for longer periods of time. The hyperparameters used to train the Q-Learner are vital to the convergence of the trading strategy. The Q-Learner takes in multiple parameters like the learning rate (alpha), reward decay rate (gamma), random action rate (rar) and random action decay rate (radr). The balance between exploration and exploitation is the main decision being taken in this approach. Because it takes the Q-Learner some time to learn how to trade the market you need a lot of data. The Q-Learner iterates through the training range multiple times to find the best path of action. On the other hand, because the Q-Learner has a random aspect to it, it may not always converge to the same path. This might mean that you want to decrease your random action rate or decrease its decay. However, you want to allow the algorithm to explore in case there is a better decision sequence. Our visualization allows the user to explore these parameters in combination with the multiple provided indicators. Figure 2 below includes sentiment data for the given time range. It is difficult to tell what impact sentiment has on performance without executing more experiments. However, within our set of assessments sentiment seems to add values to the analysis when chosen as an indicator. The direct effect of sentiment was not quantified within the scope of this project.



<u>Figure 2:</u> BTC Benchmark vs Q-Learner Strategy Portfolio Performance for training (Top) and testing (Bottom) using all available pricing, Twitter sentiment, and other indicators.

Results in Figure 3 below show that Grid Search Learner algorithm is successfully trained to determine good times to buy that result a higher portfolio profit compared to the benchmark. For both the training and testing period, Grid Search produces a return that is 70% higher than the benchmark. One concern is threshold permutation runtime limitation. To keep the runtime less than 10 seconds, we can only train on 3 indicators at a time.



<u>Figure 3:</u> BTC Benchmark vs Grid Search Strategy Portfolio Performance for training (Left) and testing (Right) using 3 indicators: Bollinger Band, Simple Moving Average (SMA), and Sentiment

As a comparison, Q learner may not perform well in the beginning as it hasn't seen all the possible states and is exploring the experiment space. However, as the Q-learning performance gradually improves over time as it gets more mature. Grid Search performance is more consistent throughout training and testing. However, it takes longer to train and won't perform well in new scenarios where indicators reach values it hasn't seen before. If the market is stable, the Grid Search method would perform better. If the market fluctuated too much, the Q-learner would perform better as it can learn and improve the Q-table.

Visualization and user interactive capability enable users to learn more about how indicators reflect market behavior. Users will be able to explore possible indicators options and two machine learning methods to find a personalized market analysis technique.

<u>**CONCLUSIONS AND DISCUSSION**</u>

In conclusion, our team built an end-to-end pipeline displaying statistical indicators for Bitcoin Price, as well as creating a proprietary Q-Learning Algorithm and Grid Search to display Buy/Sell Signals. Our front-end visualization user-interface also allows the end-user to customize the statistical indicators themselves and input their own Machine Learning Hyperparameters to test their own perspectives on the market for historical data. Our team spent a significant amount of time iterating on Q-Learning and Grid Search Algorithms to optimize the buy/sell signals against the benchmark, as well as finding a way to display these results in an intuitive manner. Q-Learning and Grid Search are two significant algorithms used in academia to study markets that our user-interface allows end-users to easily manipulate and intuitively visualize. The team also iterated on automating the preprocessing of raw data using Spark to feed into the algorithms

In the future, our team would like to update our application to use real-time pricing and sentiment data. Our team was unable to do this, as there are strict limitations on Twitter API Requests, as well as a lack of free real-time Pricing APIs. The inclusion of this real-time data feature would allow users to make money trading using the buy/sell signals from cutting-edge Q-Learning/ GridSearch Algorithms in real-time.

All team members have contributed a similar amount of effort and continued to do so throughout the project. Pairs of team members were created to focus on specific verticals of the project.

REFERENCES

[1] Abraham, J., Higdon, D., Nelson, J., Ibarra, J. (2018). Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis.

[2] Akbiyik, M., Erkul, M., Kaempf, K., Vasiliauskaite, V., Antulov-Fantulin, N. (2021). Ask "Who", Not "What": Bitcoin Volatility Forecasting with Twitter Data.

[3] Akhilesh, P., Akarsh, T. S., Parkavi, A. (2018). A Study of Opinion Mining and Data Mining Techniques to Analyse the Cryptocurrency Market.

[4] Amr, T. & Stamboliyska, R. (2016). Practical D3.js. 10.1007/978-1-4842-1928-7.

[5] Asgari, M., Khasteh, S. (2022). Profitable Strategy Design by Using Deep Reinforcement Learning for Trades on Cryptocurrency Markets.

[6] Damji, J. S., Wenig, B., Das, T., & Lee, D. G.-Y. (2020). *Learning spark: Lightning-fast big data analytics*. Sebastopol, CA: O'Reilly.

[7] Gu, S., Lillicrap, T., Sutskever, I., Sergey, L. (2016). Continuous Deep Q-Learning with Model-based Acceleration.

[8] Hanson, B., Seeger, C. (2016) D3.js: Introduction to Mapping

[9] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[10] Li, T., Chamrajnagar, A., Fong, X., Rizik, N., Fu, F. (2019). Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model.

[11] Li, Y., Ni, L., Chang, V, (2017). Application of Deep Reinforcement Learning in Stock Trading Strategies and Stock Forecasting.

[12] Liu, B. (2012). Sentiment analysis and opinion mining. Morgan & Claypool.

[13] Media. (2021). Summary of John Bollinger's Bollinger on Bollinger Bands. IRB.

[14] Namugera, F., Wesonga, R., Jehopio, P. (2019), Text mining and determinants of sentiments: Twitter social media usage by traditional media houses in Uganda.

[15] Narman, H. S. & Uulu, A. D. (2020). Impacts of Positive and Negative Comments of Social Media Users to Cryptocurrency, 2020 International Conference on Computing,

Networking and Communications (ICNC).

[16] Pathak, S., Kakkar, A. (2020). Cryptocurrency Price Prediction Based on Historical Data and Social Media Sentiment Analysis. In: Saini H., Sayal R., Buyya R., Aliseri G. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 103. Springer, Singapore.

[17] Poongodi, M., Nguyen, T., Hamdi, M., Cengiz, K. (2021). Global Cryptocurrency trend Prediction Using Social Media.

[18] Real Python. (2021, September 24). Sentiment analysis: First steps with Python's NLTK library. Retrieved April 2, 2022, from https://realpython.com/python-nltk-sentiment-analysis/

[19] Romero, P., Balch, T. (2014). What Hedge Funds Really Do: An Introduction to Portfolio Management.

[20] Sprenger, Tumasjan, A., Sandner, P. G., & Welpe, I. M. (2014). Tweets and Trades: The Information Content of Stock Microblogs. European Financial Management: The Journal of the European Financial Management Association, 20(5), 926–957. https://doi.org/10.1111/j.1468-036X.2013.12007.x

[21] Wattenberger, A. (2019) Fullstack D3 and Data Visualization: Build Beautiful Data Visualizations and Dashboards with D3. San Francisco, California, Fullstack.Io.

[22] Zainuddin, N., Selamat, A., & Ibrahim, R. (2017). Hybrid sentiment classification on Twitter aspect-based sentiment analysis. Applied Intelligence. https://doi.org/10.1007/s10489- 017-1098-6