Frameworks & Methods II Final Project Report

Akvilė Kovalčikaitė, Lavan Paramesvaran,

Shiyu Hua

Research Question

To what extent can the sentiment analysis of Reddit threads and comments on Ethereum be used to determine its price?

Introduction

In recent years, the popularity of cryptocurrencies such as Ethereum has dramatically increased (Haar, 2022). More so than other financial asset classes, the price of cryptocurrencies are highly reflective of the sentiment around them (Chakraborty & Subramaniam, 2021). In particular, Ethereum is a well-discussed currency on many social media platforms, where investors speculate its value. According to prior research conducted, sentiment analysis of Twitter data was used to successfully predict the performance of Bitcoin, Ethereum, Ripple and Litecoin markets (Valencia et al., 2019). As such, we believe that there is potential arbitrage to be gained through market sentiment analysis and trading Ethereum. However, unlike previous literature, we aim to perform a sentiment analysis of Reddit commenters instead of Twitter users.

We chose to analyze Reddit users for several reasons. First, Reddit has historically been a primary platform for highly engaged crypto discussions revolving around Ethereum (Glenski et al., 2019). Hence, the sentiment data obtained from Reddit discussions are likely to yield more reliable results, since its users are more heavily invested in the market than users from other social media platforms. Next, we believe that the data obtained from the

Reddit platform will be more consistent with the actual market sentiment. The presence of bots on Twitter and Meta applications can sometimes affect the reliability of results. With Reddit having substantially less bots, we can almost eliminate this problem entirely. Finally, Reddit has a clear scoring system for comments and threads, which might allow us to obtain additional information on how much a specific sentiment is shared. These additional data can be used alongside our sentiment analysis to help tune our predictive model.

Data Overview

Using the `RedditExtractoR` package in R, we gathered the threads and comments data from the Reddit website, specifying the subthread to be “Ethereum”, for two periods - the first starting from February 2nd, 2022, ending March 1st, 2022, and the second starting March 18th, 2022, ending April 18th, 2022. Since the RedditExtractoR package, and specifically, the `find_thread_urls()` function does not allow for downloading data for specific dates, solely periods (such as “all”, which is limited to 8 pages worth of URLs if the dataset is large, “month”, “week”, and “day”), we had to come up with an alternative method of gathering data for the missing period (March 2nd, 2022 to March 17th, 2022). This involved using a web scraping function written by @nathancunn on GitHub, which scrapes data from the pushshift.io API, as opposed to Reddit, as the pushshift.io API allows to filter search results by specifying a date range. Unfortunately, using the `getPushshiftData()` function, the number of observations was limited to 100 for the specified period. To circumvent this, we scraped the data for each missing day period separately, and joined the results together, which resulted in a dataframe of 1,600 observations. Unfortunately, after joining the thread and comment data, removing NAs, and removing the comments that had been deleted by their authors, we were left with a dataset of merely 382 observations. However, as there was at least one comment for each of the dates, we nevertheless decided to merge the datasets. For reference purposes, we also collected data from Google Trends using "Ethereum" as the keyword. For the price column, we collected the data from Yahoo! Finance by using the "tidyquant" package in R. In the end, our dataset consists of 23,697 observations and 9 variables. The variables are index, date, comment, commentScore, threadTitle, threadText, threadScore, hits and price:

Data Columns	**Description**
index	unique identifier
date	date of the comment created
comment	text of the comment
threadTitle	title of the thread
threadText	text of the thread
commentScore	number of upvotes minus the number of downvotes for the comment
threadScore	number of upvotes minus the number of downvotes for the thread
hits	the frequency of Google searches for "Ethereum" on a daily basis
price	daily Ethereum adjusted close price

Preparation and Cleaning of the Data

For the two datasets collected using the RedditExtractoR package, we initially separated the eth list of data frames into two data frames, dropped the irrelevant columns (such as author, number of upvotes, number of downvotes, comment ID), and matched the thread title and text to the comments using the URL column.

For the dataset used to fill in the gaps for March 2nd to March 17th, all dates had to first be converted to epoch format, as that was the sole format the function `getPushshiftData()` took. After scraping the results for each missing day, the dates had to be converted to standard date format. Then, we joined the comment and thread datasets for each missing day together, and merged the comments and threads using parent_id and id. Finally, we removed the comments that had been deleted, and any rows containing NAs.

After matching the date formats of three datasets and making sure that the columns and their order was the same for each, we joined the three datasets, and imputed the missing variables for threadScore and commentScore for the period of March 2nd - March 17th.

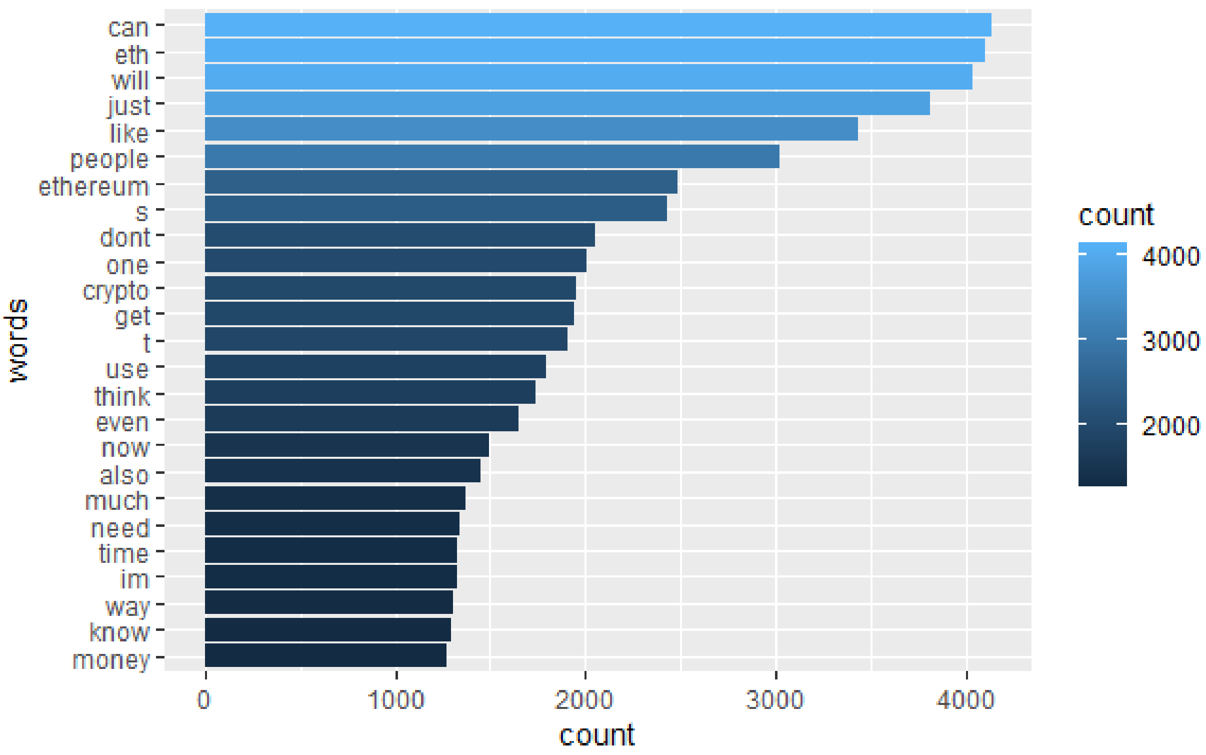
We then populated the ‘price’ and ‘hits’ columns using the date as a reference. For cleaning, we opted to remove any hyperlinks, numbers, punctuation, capitalization, codes for apostrophes (e.g. ‘\030’), and stop words for easier sentiment analysis. Upon visualizing the final results, we are able to anticipate which words will have to be removed for better analysis (e.g. ‘ethereum’, ‘eth’, ‘vitalik’). We then transformed our dataset into a Document Term Matrix which is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In doing so, we choose to remove all terms in the corpus whose sparsity is greater than 95% to reduce the amount of outlier terms in our final analysis. We then combined the Document Term Matrix with price data to produce a dataset for evaluation.

Evaluation of Data

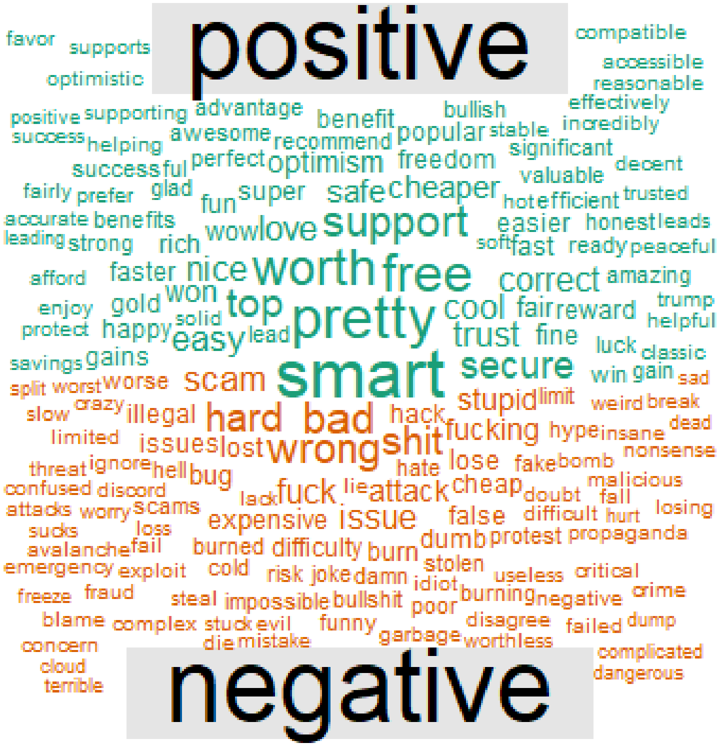
To evaluate the performance of our models we decided to measure the Root Mean Squared Error (RMSE) of predicted prices made by our models on the test data when compared to the actual prices of Ethereum in the test dataset. RMSE is a measure of how spread-out residuals are from the actual prices and is an effective determinant of model accuracy since it is scale dependent. The lower the RMSE score, the more accurate a model is at predicting price.

Visualization of Data

Before conducting our predictive sentiment analysis, we decided to conduct some general sentiment analysis of the data to understand the distribution of words within our dataset. First, we created a word count graph to display the words with the highest frequency in the dataset. This process allowed us to identify additional words such as “eth”, “s” and “t”, which do not possess any meaning without context. Subsequently, we removed these words from the dataset.



We also created a word cloud to visualize the most common positive and negative comments within our dataset. This word cloud gave us insight into what kinds of words were being said, and served as an illustration to what words were going to be scored in our analysis. We learnt that there is a large diversity of terms within our dataset, and consequently had to choose an analysis method that effectively dealt with this diversity.

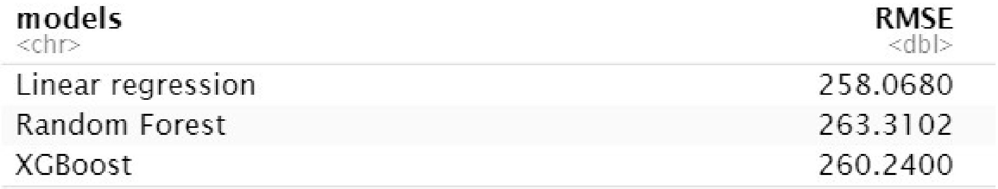


Sentiment Scoring

To conduct our analysis of the data, and eventually develop models, we first went through the process of scoring the data through Latent Semantic Analysis (LSA). We found LSA to be a robust method for scoring comments due to its widespread use and reliability. Additionally, we chose LSA due to its ability to score diverse topics within our dataset effectively since it operates on a global level and looks at trends and patterns from all documents and words to find things that may not be apparent to a more locally based algorithm.

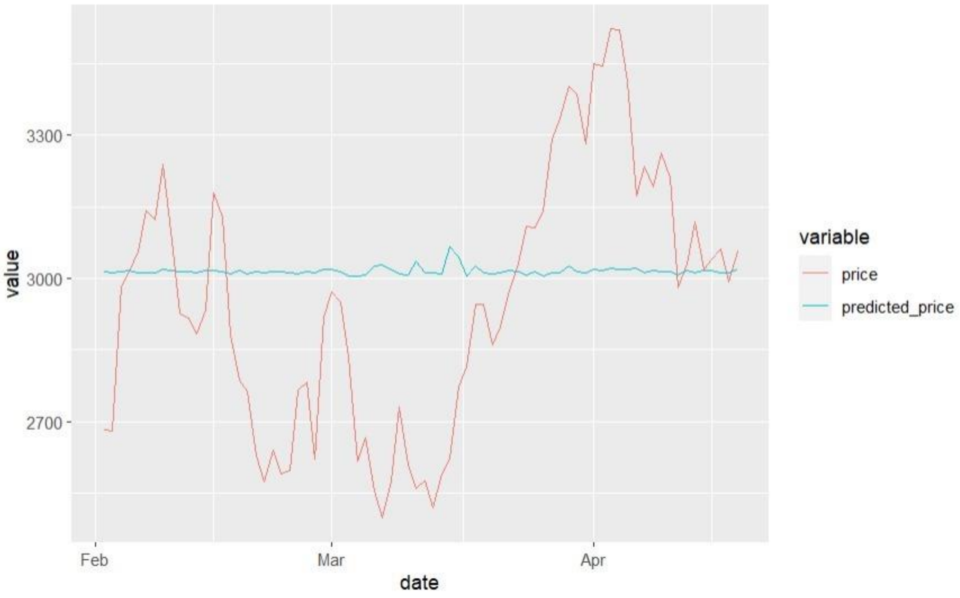
Model Development and Analysis of Data

In our model development process, we attempted to use a variety of different models to score the data and compare RMSE scores. We utilized a Simple Linear Regression Model, Random Forest Model and XGBoost model. The resulting RMSE scores can be seen below.



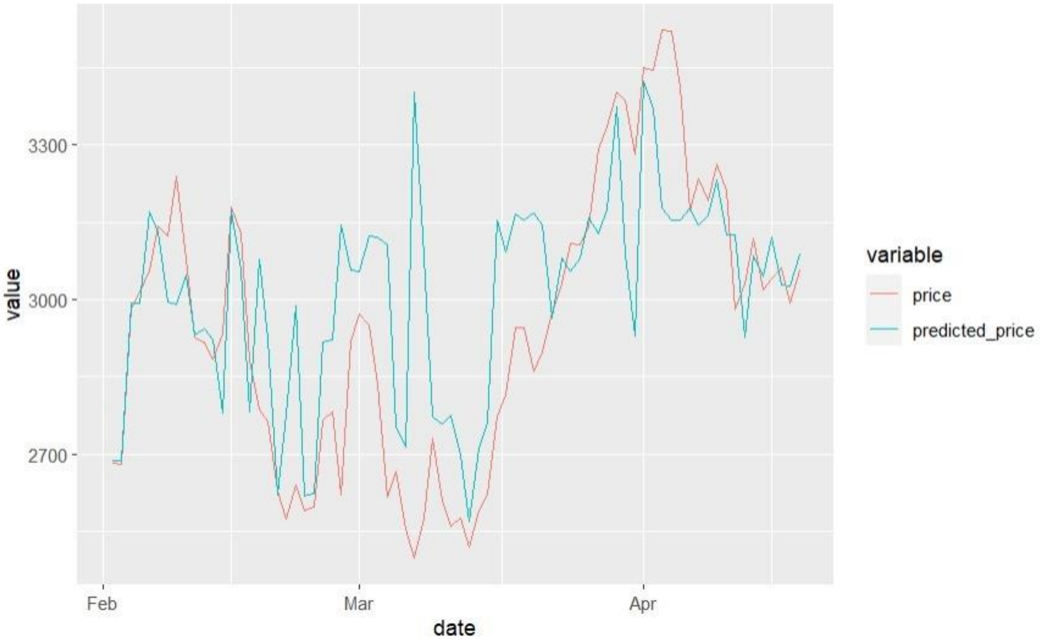
models <chr>	RMSE <dbl>
Linear regression	258.0680
Random Forest	263.3102
XGBoost	260.2400

To our surprise, our least sophisticated approach - Simple Linear Regression - had marginally the best RMSE score of 258.07. Despite this, we decided to continue with the XGBoost model since we could potentially tweak its parameters to gain a lower RMSE score. Through a process of trial and error, we were able to optimize the XGBoost model to predict price data on the test dataset. We then merged our predictions with the original dataset to compare our predicted prices with actual price data. This comparison can be seen in the graph below.



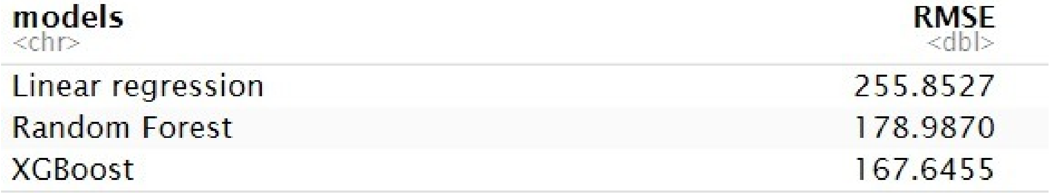
From this graph, we see that our XGBoost model using sentiment data on Reddit comments was not effective in predicting the price fluctuations of Ethereum. This model is relatively ineffective in practice, since the actual fluctuations in price are barely reflected by the model. To improve upon our model, we chose to incorporate some of the additional data that we collected. We decided that since we were using an XGBoost model, we would try to incorporate as much of the relevant available data that we had collected. This data included the comment scores on reddit posts, data on Google hits, and 3 additional sentiment score variables. The additional 3 sentiment score variables were obtained by scoring the original sentiment data using the lexicon packages affin, senti and mcd. We hoped that the wider range of lexicons used would allow us to account for the larger variety of words within our dataset.

The result of this process was that we had a large matrix dataset which we used to run our tuned XGBoost model on. The model had a significantly improved RMSE score of 167.65. Our new comparison can be seen in the graph below.



As seen in the graph, our new predicted model is far better at determining the price of Ethereum. Not only are fluctuations well aligned with actual market fluctuations, but they also respond with a similar change in amplitude. Despite this, we can see that in some scenarios such as in the 2nd week of March (when the dataset was rather sparse), our model greatly exaggerated the price increase in Ethereum. This indicates that our model is either not accounting for some significant variables that determine the price of Ethereum, or that our model is overfitting the data.

In addition to our XGBoost model, we also decided to run our additional data through our Simple Linear Regression and Random Forest models. As expected, the RMSE of these models did not outperform the XGBoost, but it is worth noting that the Random Forest model performed significantly better, and could be a viable approach if we were unable to use a XGBoost model. However, since we are already utilizing a Document Term Matrix, implementing an XGBoost model wouldn't be a significant issue.



models <chr>	RMSE <dbl>
Linear regression	255.8527
Random Forest	178.9870
XGBoost	167.6455

Conclusion

In this paper, we set out to determine to what extent sentiment analysis of Reddit threads and comments on Ethereum can be used to determine its price. We found that sentiment analysis of Reddit comments alone is a relatively weak predictor of the price of Ethereum. However, when combined with a variety of different data points, sentiment analysis of Reddit comments can provide a relatively robust model for predicting the price of Ethereum as seen from the results of our XGBoost model. Part of the success of our predictions can be attributed to the use of an XGBoost model since further testing of the other models on the expanded dataset did not yield an RMSE score that was as favorable as the XGBoost model. However, it should be noted that a majority of the success of our approach comes from combining multiple sources of related sentiment data to more accurately account for the sentiment shared by the market.

Limitations

One of the key limitations of our approach is that our final analysis did not consider sentiment data from Reddit threads. Although we had extracted and cleaned this data for analysis, we were not able to obtain a significant amount of thread sentiment data through our API; there was a substantial number of missing values. As such, we felt that adding this data to our model at this time would not greatly improve its predictive ability.

Despite having found a relatively strong approach from our results, we believe that the limited timeframe that we have collected data in does not fully represent the range of price fluctuations that Ethereum can go through. As seen from the exaggerated spikes in the price of our predicted model, we aren’t certain that the model will continue to have the same level of predictive capability as seen in our tests. Also, it should be noted that significant price corrections or spikes cannot be accounted for by the model since the model does not consider any external information that might concern the underlying value of Ethereum as an asset. As such, opportunities that may present the greatest arbitrage can never be predicted by our model. Conversely, significant price corrections can hurt individuals who strictly rely on this model. Another limitation is that our model might be overfitting the data presented to it. For instance, there might be periods where market sentiment loses some of its predictive connection to the price of Ethereum. If we were to use our model in such instances, it might not effectively predict the strength/amplitude of fluctuations in the price of Ethereum. Ultimately, the model is only useful in guidance with external information regarding Ethereum and more testing of this approach is required before it can be implemented as a viable trading strategy.

Future Improvements

Since we did not use our thread data for this project, we would aim to find a more reliable way of extracting and cleaning thread data. Additionally, we would like to find a way to contextualize the thread and comment data together, as it might improve the predictive ability of our sentiment analysis.

One immediate improvement that can be made to our approach is to collect data over a longer period such as 6 - 12 months to see how well our predictions hold up when there are major changes in the price of Ethereum. This process would also give us more data to train our model and potentially improve its predictive ability since sentiment analysis can improve significantly with more data available. Additionally, we could consider using sentiment data from other sources. Given that our project was specifically on Reddit data, the opinions of Reddit users might be skewed in a particular direction when compared to users of other communication platforms. As such, incorporating more diverse sources of sentiment data could improve the predictive ability of our sentiment analysis XGBoost model. It is also important to remember that the RMSE and predictive ability of our model was greatly improved with the introduction of other variables. By adding variables that consider the underlying performance of Ethereum, we can likely predict larger fluctuations in its price. One way to accomplish this is by conducting a sentiment analysis of specific Ethereum blogs and analysts to better account for changes in the underlying value of Ethereum as an asset. Finally, if we were to consider building the strongest predictive model, it would involve combining multiple machine learning techniques into a single model or ensemble of models. This might include the use of a time series and risk assessment model. However, this is currently out of scope for this project and would have entirely new considerations if attempted.

References

Chakraborty, M., & Subramaniam, S. (2021, August 4). *Does sentiment impact cryptocurrency?* Taylor & Francis. Retrieved March 2, 2022, from https://www.tandfonline.com/doi/abs/10.1080/15427560.2021.1950723

Fortune Business Insights. (n.d.). *Cryptocurrency market size, Growth & Trends: Forecast [2028]*. Cryptocurrency Market Size, Growth & Trends | Forecast [2028]. Retrieved March 2, 2022, from https://www.fortunebusinessinsights.com/industry-reports/cryptocurrency-market-100149#%3A~%3Atext%3DBased%20on%20our%20analysis%2C%20the%2Cduring%20the%202021%2D2028%20period

Glenski, M., Saldanha, E., & Volkova, S. (2019, May 1). *Characterizing speed and scale of cryptocurrency discussion spread on reddit: The world wide web conference*. ACM Other conferences. Retrieved March 2, 2022, from https://dl.acm.org/doi/10.1145/3308558.3313702

Haar, R. (2022, February 9). *Future of cryptocurrency in 2022 and beyond | nextadvisor with Time*. Time. Retrieved March 2, 2022, from https://time.com/nextadvisor/investing/cryptocurrency/future-of-cryptocurrency/

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019, June 14). *Price movement prediction of cryptocurrencies using sentiment analysis and machine learning*. MDPI. Retrieved March 2, 2022, from https://www.mdpi.com/1099-4300/21/6/589

PushShift.io API:

https://github.com/nathancunn/pushshiftR

https://reddit-api.readthedocs.io/en/latest/