Big Data Science

CSCI-GA.3033-​001

Prof. Anasse Bari

New York University

Final Project

Kevin Choi (kc2296)

**Predicting Ethereum’s Price**

**Based On Its Blockchain Activities and Sentiment Towards Cryptocurrency**

**0. Abstract**

While the market of cryptocurrency has been an unprecedentedly volatile one in the past decade, we explore the case with Ethereum since 2019 such that we gather, preprocess, and process data from three main sources in an attempt to model its price: price data itself, blockchain data, and data from Twitter. Here, blockchain data refer to on-chain data such as average gas price, total hash rate in the blockchain network, number of existing addresses, etc. all of which signify the health of the network as a global data structure. As for Twitter, we note the advent of so-called Crypto Twitter, a Twitter clique of investors, traders, engineers, celebrities, and speculators, and thus use the tweets by top Crypto Twitter influencers in our prediction model. Incorporating these data in addition to Ethereum's price data, we observe that our model is able to generate predictions with smaller error when compared to traditional time series models.

**1. Introduction**

Launched in 2015, Ethereum is currently the second most popular cryptocurrency in the market (by market cap). As it is often compared with its predecessor Bitcoin, one of the key differences is that Ethereum was the first blockchain to enable a wide usage of smart contracts (a concept that dates back to 1994 by Nick Szabo), which have been increasingly gaining popularity in various forms since Ethereum's inception. For instance, some of these smart contracts have taken the form of an auctioning game (e.g. CryptoKitties) while others have demonstrated various use cases such as certificate authentication, copyright protection, insurance, supply chain management, tunable peer-to-peer transactions, decentralized governance, and voting.

As this has been the case, it has been possible for Ethereum to publicly generate a lot of auxiliary data via its network as network participants not only generate transactions among themselves, but also interact with numerous smart contracts that have been deployed in a continual manner. We outline Ethereum's network data in the following.

1. Supply: As an Ethereum block becomes mined by a miner (who by definition acts as a verifier of Ethereum transactions), new Ethereums are generated to compensate those miners. (In addition, there exist compensations related to mining uncle blocks as well.) As a result, the supply of Ethereum can be seen as a function that is always increasing over time.

2. TxGrowth: Transaction growth denotes the number of transactions per time frame.

3. AddressCount: Whenever a network participant generates a new public address, such event makes a contribution of 1 to Ethereum's address count.

4. BlockSize: That an Ethereum block is big means that it is packed with transactions. Bigger block size means either the network is full of transactions, miners are letting each block become big, or both.

5. BlockTime: Historically, the block time of Ethereum has hovered around 10-19 seconds. This is the time interval between when transactions get published and verified.

6. AvgGasPrice: Gas represents Ethereum's network fee. High gas price means the transaction fee is high at the moment due to (perhaps) competition of transactions when there are too many.

7. GasUsed: This denotes the amount of gas that has been used per time frame.

8. NetworkHash: The network hash rate refers to the amount of computing power miners are dedicating to the Ethereum network at the moment in an attempt to mine Ethereum blocks. Higher rate means there exists a higher competition among miners.

9. TransactionFee: This denotes the total amount of network fee in Ethereum (not in gas) per time frame.

10. NetworkUtilization: This denotes the average gas used over the gas limit in percentage. Each (smart contract) transaction can specify the maximum amount of gas that can be used for that transaction.

11. BlockReward: Higher block reward means that miners are compensated more per block mined such that miners are more incentivized to mine.

As such, it is interesting to see that the above parameters would tend to reflect the complex game-theoretic dynamics within the Ethereum network in theory, although things have not necessarily been always predictable (in a theoretical way) in practice. Then the idea is that such dynamics would either reinforce or be reinforced by Ethereum's price such that exploring their relationship is of interest.

Meanwhile, we note the advent of so-called Crypto Twitter, which has sprouted out as a substantial community on Twitter as a clique since the cryptocurrency boom in late-2017. In it, the observation is that thousands of crypto investors, traders, engineers, celebrities, and speculators consume news and sentiment-based opinions on a daily basis such that we deemed it necessary to include such data from Twitter to our exploration.

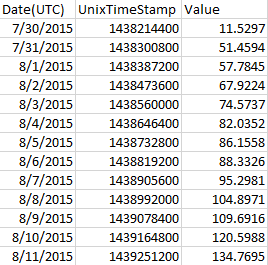
**2. Prior Work**

After an impactful work by Bollen et al. (in which the authors utilized the GPOMS sentiment metric on tweets to achieve an accuracy of 87.6% in predicting the daily up and down changes in the closing values of the DJIA), many more have come into existence when it comes to using social media data to predict the financial market of some sort. Specifically with regards to the cryptocurrency market, there have been works by Stenqvist and Lönnö as well as Colianni et al., in both of which the price fluctuation of Bitcoin has been explored using data from Twitter. In the former, the authors experiment with a substantial number of parameters (e.g. using 7 different time intervals with multiple threshold hyperparameters) while the latter focuses on applying 3 different supervised machine learning algorithms (namely, naive Bayes, logistic regression, and SVM).

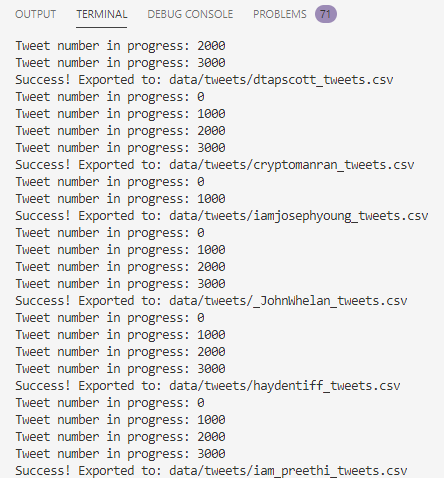
In this paper, we explore the regression problem of predicting prices using techniques such as GBT (gradient boosted trees), SVM, and neural network. In addition, the novelty is derived from the fact that we perform analysis on tweets from influencers as opposed to everyone (in which case noise can be a significant issue).

**3. Data**

Price data and blockchain data have been retrieved from Etherscan. They take the following form when retrieved:

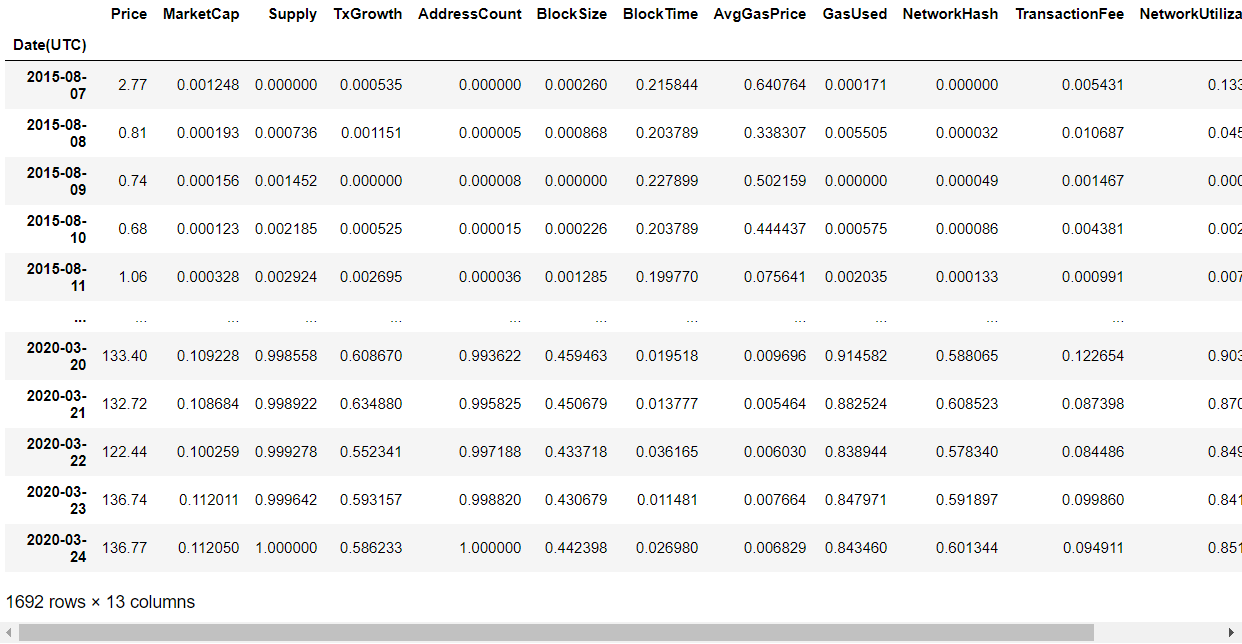


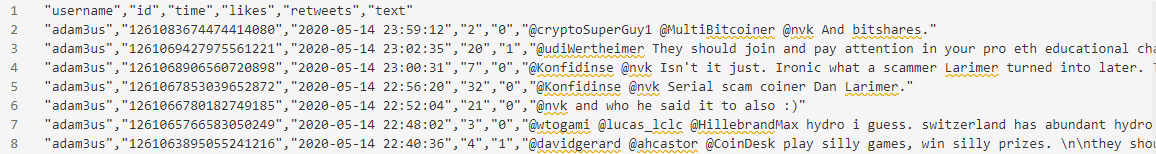
After retrieving 100 Twitter handles of top Crypto Twitter influencers from CryptoWeekly (this process involved scraping the web), we retrieved more than 200,000 tweets from those influencers directly from Twitter using Python's Tweepy. In order to allow future retrieval of data with ease, we even implemented a way to prepend (as opposed to append due to the reverse-chronological nature of our Twitter data) future data to the existing data in a seamless manner.

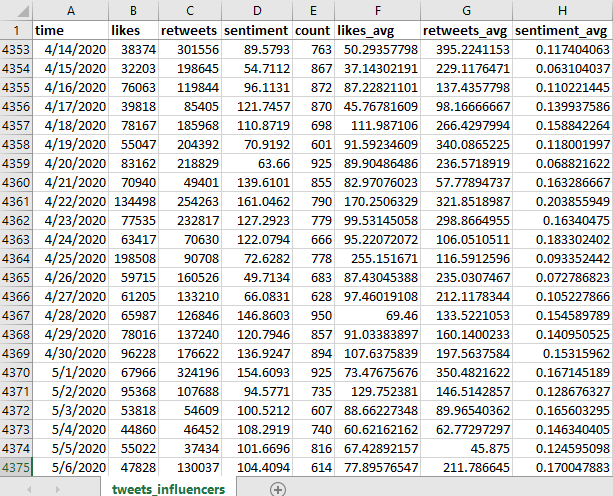


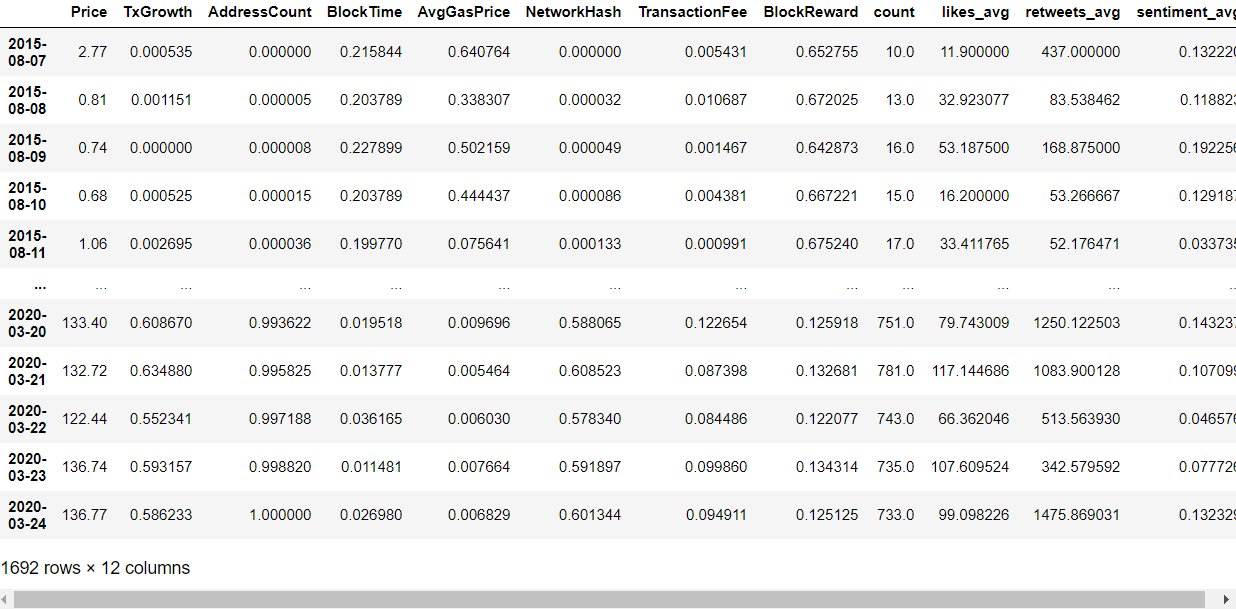
Data preprocessing involved removing and adding columns within a spreadsheet, joining multiple spreadsheets of blockchain data, applying a Pearson correlation threshold of 0.8 to remove some features (some were intentionally not removed), performing min-max normalization, removing handles and links from tweets, performing sentiment analysis and resampling (based on a different time frame) on tweets, and joining blockchain data's spreadsheet with Twitter data's spreadsheet.

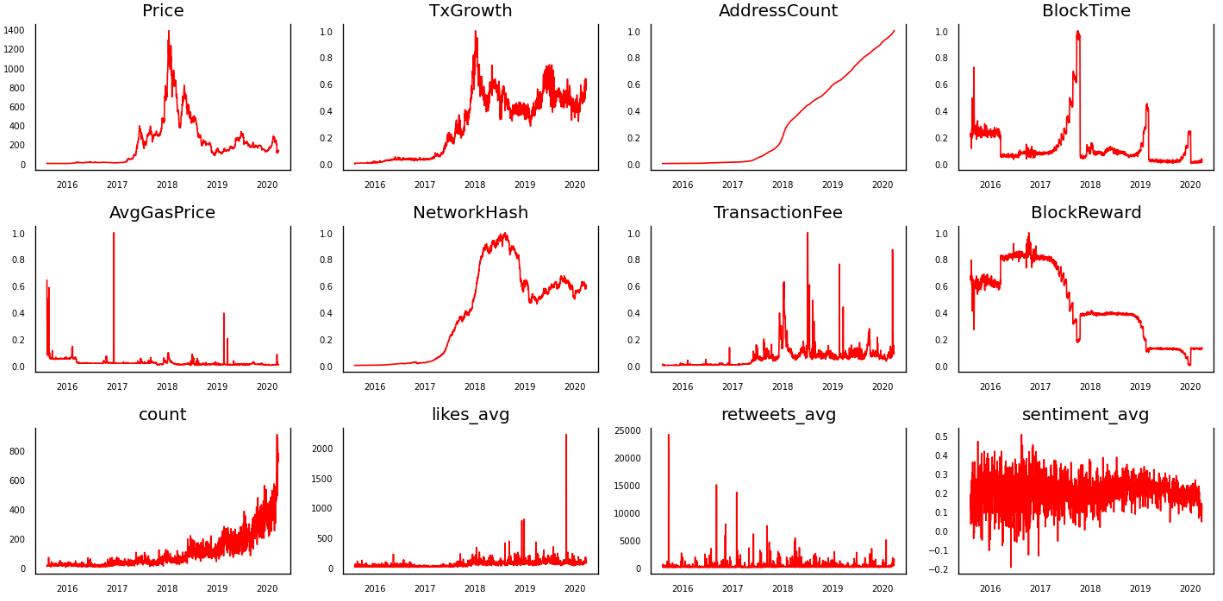
The following 5 figures indicate compiled blockchain data, raw tweets, resampled tweets across all influencers from the list, and final feature matrix in a time series format (and its plot), respectively.











Note that 4 key features were used from Twitter data: number of influencer tweets per day, average number of likes per influencer post, average number of retweets per influencer post, and average sentiment score per post. Also note that both blockchain data and Twitter data were daily-based (e.g. daily prices, daily post counts, etc.).

**4. Methods**

**4.1. Traditional time series modeling**

As the control case, we have performed traditional time series modeling (e.g. moving average, Holt-Winters, and ARIMA) to begin with. Usually, such traditional time series models do tend to work well on traditional times series data. As our Results section would show, however, we verify that our (unorthodox) Ethereum price data certainly pose challenges for these traditional models.

As for moving average, we experimented with 3 different time windows, i.e. 7 days, 30 days, and 100 days. As for Holt-Winters, we show that assuming a 30-day season yields a better result compared to assuming a quarterly (90-day) season, although any assumption of seasonality is in fact not quite foundational (as the ACF plot would reveal) in the context of a cryptocurrency market. As for ARIMA, we use Python's pmdarima to automatically find the optimal parameters p, d, and q for ARIMA(p,d,q) as well as manual testing to make predictions using (p,d,q) = (3,1,1), (7,1,1), (14,1,1), (21,1,1), and (28,1,1). Since p is the parameter that corresponds to the "AR" (autoregressive) part of ARIMA, we deemed it necessary to test p up to 4 weeks, as our basic moving average analysis has supported the case for a longer time frame (e.g. 4 weeks) as opposed to a shorter one (e.g. 3 days).

**4.2. Blockchain data and Twitter data**

While only preprocessing (e.g. normalizing and joining) was sufficient for blockchain data, such was not the case for Twitter data. When it comes to sentiment analysis specifically, we performed VADER (Valence Aware Dictionary and sEntiment Reasoner), which is a combined lexicon and rule-based sentiment analysis framework developed by Hutto and Gilbert. As it is capable of outputting the sentiment intensity from a scale of -1 (negative) to 1 (positive), it was originally developed as a solution to the difficulty in analyzing the language, style, and symbols used in the realm of social media. As it was shown to consistently perform competitively against 11 other semantic analysis tools and techniques, we chose VADER as our sentiment analysis tool.

**4.3. RapidMiner**

Due to the fact that our data concerns time series, we note that the typical cross validation approach to finding the optimal hyperparameters is not ideal. As a result, we experimented with the sliding window validation approach on RapidMiner, using 3 main machine learning algorithms: GBT, SVM, and neural network.

We assumed our features include a lag of n days, for n equals 1, 2, 3, 7, or 10. In other words, blockchain data and Twitter data for up to 3 days were included in training and testing when n equals 3.

**5. Results**

All data (training, validation, and test) concerned dates from January 1, 2019 to March 24, 2020.

1) Moving Average (100 days)

{'rmse': 50.54171659948686,

'mape': 0.27453641877087115,

'mae': 48.59675675499917,

'mpe': -0.04521768454604853,

'me': -8.459796183709718,

'corr': -0.8196626617589882}

2) Holt-Winters (30-day seasonality)

{'rmse': 30.225390595602903,

'mape': 0.14016508672670275,

'mae': 24.116357147381468,

'mpe': -0.06729721535583888,

'me': -9.983193779668694,

'corr': 0.8438655042858493}

3) ARIMA(21,1,1)

{'rmse': 63.43456574895235,

'mape': 0.25273057581705255,

'mae': 52.70160057617348,

'mpe': -0.1825190145722052,

'me': -38.3810569724745,

'corr': -0.2353095103134466}

4) GBT

* Number of trees: 60
* Learning rate: 0.15

root\_mean\_squared\_error: 16.765 +/- 0.000

absolute\_error: 10.619 +/- 12.974

relative\_error: 5.76% +/- 9.06%

correlation: 0.950

5) Neural Network

* Learning rate: 0.003
* Momentum: 0.58

root\_mean\_squared\_error: 14.318 +/- 0.000

absolute\_error: 9.008 +/- 11.129

relative\_error: 5.30% +/- 9.09%

correlation: 0.957

6) SVM

* Kernel type: dot
* C: 100

root\_mean\_squared\_error: 14.726 +/- 0.000

absolute\_error: 9.054 +/- 11.614

relative\_error: 5.22% +/- 9.37%

correlation: 0.958

**References**

Evita Stenqvist and Jacob Lönnö. "Predicting Bitcoin price fluctuation with Twitter sentiment analysis." (2017).

C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth International AAAI Conference on Weblogs and Social Media, 2014.

Stuart Colianni, Stephanie Rosales, and Michael Signorotti. Algorithmic trading of cryptocurrency based on twitter sentiment analysis. 2015. http://cs229.stanford.edu/proj2015/029\_report.pdf

Hong Kee Sul, Alan R Dennis, and Lingyao Ivy Yuan. Trading on twitter: Using social media sentiment to predict stock returns. Decision Sciences, 2016.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis.

Etherscan. https://etherscan.io/charts

CryptoWeekly. The 100 Most Influential People in Crypto 2020 Edition. https://cryptoweekly.co/100