Predicting Cryptocurrency Market

Analyzing cryptocurrency market using Twitter sentiment analysis

Borga Edionse Usifo
Indiana University
Bloomington, Indiana 47408
busifo@iu.edu

Shivam Kapadia
Indiana University
Bloomington, Indiana 47408
skapadi@iu.edu

Sushmita Dash
Indiana University
Bloomington, Indiana 47408
sushdash@iu.edu

ABSTRACT

Cryptocurrencies have a way to change the way Internet-connected global markets interact with each other, clearing away barriers surrounding normative national currencies and exchange rates. this project, we analyze the statistical properties of the most considerable cryptocurrencies of which Bitcoin is a leading example. The bitcoin prices derived from the Coindesk and other exchanges such as GDAX.

We will collect data from form a common microblogging site, Twitter and determine the polarity score of the user's tweets for a given time duration. Sentiment analysis is going to perform on the extracted data. Sentiment analysis classifies the data into either positive, neutral or negative connotation with distant supervision.

From these data, we will try to find out if there is a correlation between the price changes and the polarity scores of the user tweets. We will be using multiple methods to deliver the best result. We will also do a correlation between Bitcoin and other alternative coins.

KEYWORDS

Twitter Sentiment Analysis, Cryptocurrency, Machine Learning

1 INTRODUCTION

Cryptocurrency is a medium of exchange, created and stored electronically in the blockchain, using encryption techniques to control the creation of monetary units and to verify the transfer of funds [1]. Cryptocurrency and more specifically Bitcoin was created in 2009 by Satoshi Nakamoto, a pseudonym developer. Below is a flow diagram from blockgeeks.com that confirms the method of creation of the cryptocurrencies.

Cryptocurrency has a volatile price point. The price can sometimes fluctuate a large percentage in any given month. This is a due to the power of the price that lies with the public (Barker, Jonathan Todd). The perceived value of cryptocurrency changes and affects the market in a major way as holders may adjust their investment accordingly.

The research that has been consists of many different bots that claim to predict the price of the bitcoins using various techniques. One method of analysing the data available in the microblogging site such as Twitter, to attempt to predict the price. In our research, we will glance through various analysis and fitting available to try to find a accurate correlation. A microblogging site as Twitter is a good choice for a project like this due to its user base. The user base is people all over the world who are mainly of the technological age. Thus, since the price of the cryptocurrency is based on the sentiment of the people of the world, there is a strong correlation in the group.

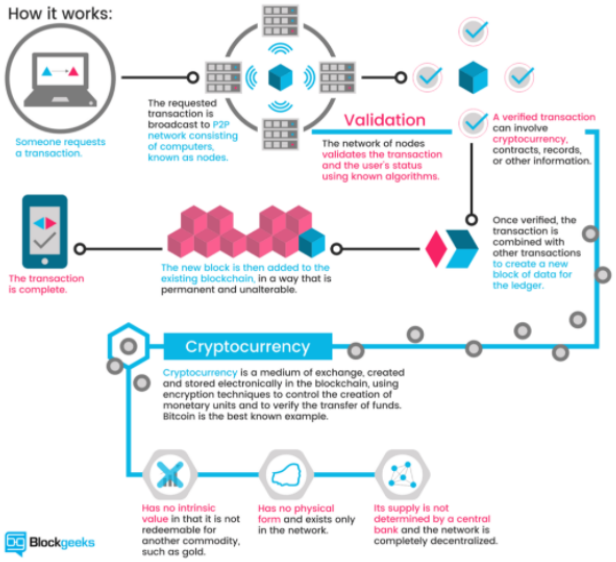


Figure 1: A draft plan for visualization [12].

2 PROBLEM STATEMENT

The objective of this project is to find the correlation between public reaction and the price of Bitcoin. If there exists a strong correlation between the public reaction and the prices, then we will be creating a prediction model based on changes. We will be achieving this using Twitter. We will perform sentiment analysis on data collected twitter and using Machine Learning models to fit the data to see if a plausible model ensues. We will also be seeing if there exists a correlation between Bitcoin and the alternative coins such as Ethereum, Bitcoin cash, and Litecoin. Also, the correlation between each other coins to see if the price of one coin affects others.

3 RELATED WORK

There have been previous research done into the sentiment analysis of Twitter data to predict the prices of cryptocurrency. There has been research done where time data was collected and analyzed and the time chunks were forward by 15 min to 6 hours to get a accurate price point. However, this did not provide real-time feedback. Also we believe that such a model proves to be inconsistent as a lot can change in just a few minutes. For example, a ban in a major trading country could cripple the market while still within the time frame of the research done [20]. We aim to find a solution for real-time data that can predict the price.

4 DATA COLLECTION

4.1 Time Frame

The data collected for this project between 4/12/18 16:42 to 4/13/18 1:00. A more extensive data set was not accessible to work with due to limitations of time consumption.

4.2 Cryptocurrency Price Data

The price data for Bitcoin, Ethereum, Bitcoin Cash and Litecoin were collected with the GDAX API every 15 minutes. The data was collected and then averaged the open price and the close price to get the average price over that time frame of 15 minutes. This is relevant because we will be using GDAX as the exchange. The price of the coin depends on the exchange. This is an issue due to the price of Bitcoin differs from exchange to exchange. Also for this project, we will be using data from coindesk. The data collected from the Coindesk.com site and it is available in CSV format.

Date	BITSTAMP	COINBASE	ITBIT	KRAKEN
2018-04-03	7335.186319	7314.408376	7348.160038	7343.227556
2018-04-04	7031.668557	7005.337171	7036.746973	7020.550685
2018-04-05	6747.860157	6738.859545	6744.800604	6737.633988
2018-04-06	6639.738385	6641.278312	6660.611024	6627.393905
2018-04-07	6896.514133	6897.519525	6890.480055	6897.212525

Figure 2: A draft plan for visualization.

4.3 Gathering Tweets from Twitter in Real Time

Twitter data was collected using Twitterfis streaming API and a Python module named Tweepy. Tweepy, an open source framework written in Python, facilitates tweet collection from Twitterfis API.

We have successfully implemented the continuous data collection with using the Google Cloud Platform. We created the virtual machine and installed required Python code to run on cloud servers for collecting twitter data continuously.

Tweepy allows for filtering based on hashtags or words, and as such was considered as an efficient way of collecting relevant data. The filter keywords were chosen by selecting the most definitive Bitcoin context words, for example “cryptocurrency“ could include sentiments towards other cryptocurrencies, and so the scope must be tightened further to only include Bitcoin synonyms. These synonyms include: Bitcoin, BTC, Cryptocurrency.

The data was collected in JSON format. The tweet collected included the fiText,fi fiLocation,fi fiCountry Code,fi fiLongitude,fi and fiLatitudefi and when the tweet was created. Then the following libraries are imported JSON for parsing the data, pandas for data manipulation.

5 DATA CLEANING AND PREPARATION

The Twitter data was collected was then tokenized and normalized. All links also removed from the data as well as all images. The tweets were then filtered for only the English language tweets. There were also a few tweets that were collected from the Twitter API that did not parse correctly and were formatted inconsistently. A small number was deleted manually.

Furthermore, for data preparation we created three different dataset one for how many tweets that are being tweeted every hour.

6 DATA ANALYSIS

After data cleaning step is done. We wanted to get some insights from the data and did some statistical analysis to understand the overall premises of the data. As shown in below Table 3 we were collecting an average of 273 tweets per minute. Average polarity and subjectivity score was 0.10 and 0.27 respectively with the standard deviations of 0.13 and 0.15.

	polarity	subjectivity	Close Price	ones
count	487.000000	487.000000	487.000000	492.000000
mean	0.107232	0.273460	7765.055359	272.636179
std	0.132418	0.156638	95.609554	59.442280
min	-0.500000	0.000000	7583.260000	30.000000
25%	0.019185	0.168155	7691.020000	227.750000
50%	0.091582	0.258852	7720.160000	271.000000
75%	0.178733	0.360357	7850.480000	309.000000
max	0.700000	0.833333	7981.900000	563.000000

Figure 3: Overall Statistics of Data Frame

Seventy five percent of polarity and subjectivity fall into 0.17 and 0.36 respectively. We identified that out of 134,195 tweets 40.5% were positive, 50% were neutral, and 9% were negative.

We also wanted see how was the distribution of our data on our variables. As shown in below histogram Figures 4, 5.

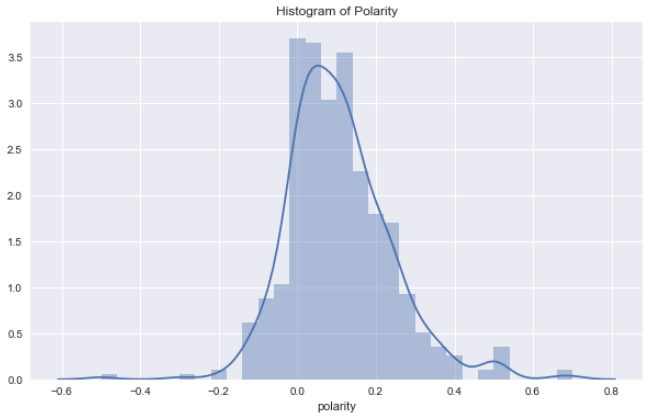


Figure 4: Histogram of Polarity

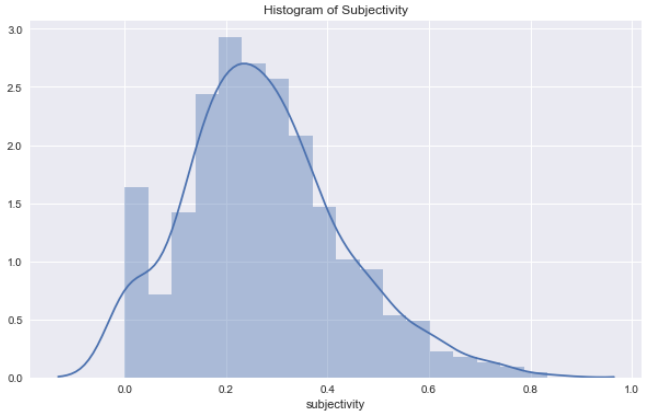
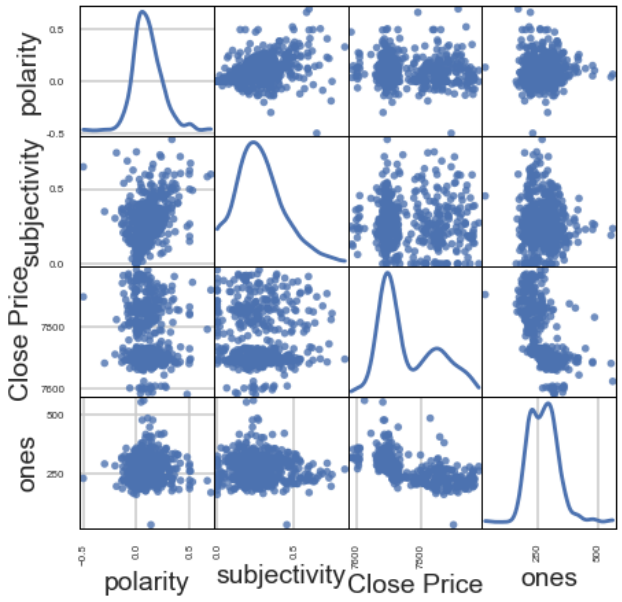


Figure 5: Histogram of Subjectivity

Below Figure 6 and Table 6 shows the correlation matrix between our variables. Note that "ones" is referencing the number of tweets that users tweeted in that minute.



	polarity	subjectivity	Close Price	ones
polarity	1.000000	0.369400	-0.086706	0.021240
subjectivity	0.369400	1.000000	-0.027066	-0.032462
Close Price	-0.086706	-0.027066	1.000000	-0.637943
ones	0.021240	-0.032462	-0.637943	1.000000

Figure 6: Scatter Correlation Matrix

As we can see from Figure 6 that we don't have strong correlation between our variables. The best correlation we identified is between the subjectivity and polarity scores.

6.1 Time Series Analysis

As we stated before we also wanted to get user behavior while using the time series analysis and wanted to see what are the user behaviors especially polarity scores each minute as shown in Figure 7. Time series analysis in bitcoin prices also included in Figure 8.

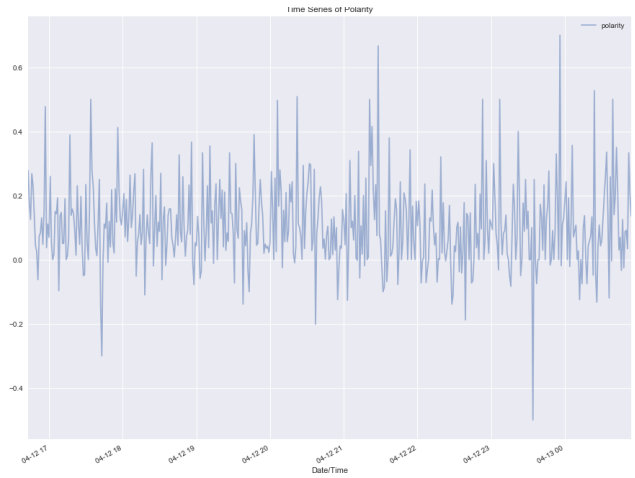


Figure 7: Polarity change in every minute.

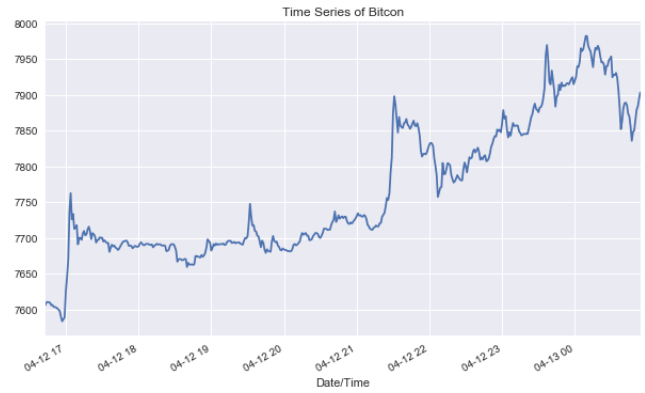


Figure 8: Bitcoin price change in every minute.

We can see that there is some correlation between the spike points of the polarity and the bitcoin price.

7 MACHINE LEARNING ALGORITHMS TO CONSIDER

We have multiple algorithms to consider when we are doing the supervised learning. Each algorithm has its benefits and drawbacks. We will consider several supervised machine learning algorithms for our predictions. The application we will use to implement these algorithms will be Python Scikit-Learn library. We will briefly explain each parameter included in these algorithms in Scikit-Learn.

First we'll look at the Scikit-Learn in Python framework we will go through the advantages in Scikit-Learn how we can implement any machine learning in just couple of simple line of codes in Scikit-Learn.

7.1 Why Scikit-Learn?

Scikit-learn developed by David Cournapeau in 2007. The development came from while he was working on summer code project for Google. After recognized and published by INRIA in 2010 project start the get more attention among worldwide. There are more than 30 active contributors and has secured several sponsorships from big technology companies[8]. "It also has a goal of providing common algorithms to Python users through consistent interface[2]". Scikit-Learn consists of several elements to make analytical predictions. These elements are shown below[11]:

Supervised Learning Algorithms: One of the most fundamental reason that Scikit-Learn's popularity comes from highly available supervised learning algorithms. These algorithms vary from regression models to decision trees and many more[11].

Cross Validation: Scikit-Learn includes various techniques to check the accuracy or any statistical measure between training and unseen testing set[11].

Unsupervised Learning Algorithms: Scikit-Learn had also various algorithms to support many unsupervised algorithms some of these include clustering, factor analysis, and neural network analysis[11].

Various example data-sets: Scikit-Learn comes with different data sets included in its package so users can start learning Scikit-Learn without the need of any data-sets[11].

Feature extraction: It has rich feature for extracting images or text from data-sets[11].

Algorithms that we will investigate shown below; we will go more deep analysis on each of these algorithms.

- K Neirest Neighbors
- Naive Bayes
- Linear Regression
- AdaBoosting

7.2 Gaussian Naive Bayes

Naive Bayes bring many beneficial features; it is widely popular among machine learning applications[21]. The popularity of Naive Bayes comes from being able to handle large projects and data-sets faster than most algorithms[21]. It also can handle complex data-sets with categorical and non-categorical inputs [21]. Naive Bayes based on probabilistic classifier of Bayesian theory. It is also a favorite way of doing text categorization [23].

Term naive comes from it is the method of use probability among categories which assumes of independence among given class of attributes as shown in Figure 9. In other words, if we try to classify individuals from their email communications it will not take the order of words into account. Whereas in the English language we can tell the difference between sentence makes sense or not if we randomly re-order our words in the sentences. So it does not understand the text, it only looks at word frequencies as a way to do the classification. This is why it is called "Naive".

As we state above Naive Bayes derives from Bayesian Theory where the dimensionality of inputs is relatively high. Bayesian Theorem is stated below [7].

$$P(C \mid X) = \frac{P(X \mid C) \times P(C)}{P(X)} \quad (1)$$

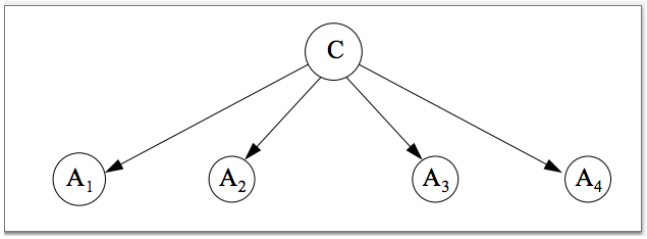


Figure 9: Example of Naive Bayes [24].

Naive Bayes Classifier works as follows [7]:

Advantages of Naive Bayes [7]:

- Faster classification time for training data-set.
- Because of independent classification it improves classification performance.
- Performance is relatively good.

Disadvantages of Naive Bayes[7]:

- Often it requires a large number of data-sets to give adequate results.
- On some occasions which are relative to data-sets, it can give less accuracy.

7.3 K-Nearest Neighbors (KNN)

K Nearest neighbor has been primarily studied, and this popularity comes from it has been applied to many applications some of these applications are "spatial databases, pattern recognition, geographic information, image retrieval, computer game, and many other applications [13]". Due to an increase of mobile devices and people tends to use of applications like navigation K-nearest neighbor found itself another widely used area of location-based services due to an ability to found a target location [13].

Intuition behind the K Nearest Neighbor can be described as follows: " for a set P of n objects and a querying point q, return the k objects in P that are closest to q [13]."

Advantages of K Nearest Neighbors:

- K Nearest Neighbor is a basic and simple approach to implement [17].
- K Nearest Neighbor can perform well and efficiently with the large amount of data [22].
- K nearest Neighbor also does effectively well with noisy data sets ("if the inverse square of weighted distance used as the distance [22]"). In other words, it is flexible to feature and distance choices [17].

Disadvantages of K Nearest Neighbors:

- K Nearest Neighbor typically require large dataset to perform well [17].
- Time complexity could be high due to computing distance of each query to all training data points [22]. This time might be improved with some indexing (K-D Tree) [22].
- Determining the value of K can be time-consuming [22].
- It can be unclear to know which type of distance to use, as well as which variability to use to get the optimal results [22].

- Switching the different K values can result in the predicted class labels [14].

Many of these disadvantages are improving with the help of parallel distributed computing. Recent improvements in MapReduce framework allows users to run KNN algorithms in the cluster which had a significant effect on reducing the computation time [9].

Another area of improvements on KNN, is to implement different mapping functions such as kernel KNN, kernel difference weighted KNN, adaptive quasi-conformal kernel nearest neighbor, angular similarity, local linear discriminant analysis, and Dempster-Shafer [5].

7.4 Decision Trees

Decision Tree is another widely used algorithm model for classification and regression. Decision Trees uses a recursive split model where each recursive split is identified by each data point; this is an example of non-parametric hierarchical model [6].

Representation of decision trees is as follows; we sort the instances from root to leaf nodes, this sorting gives insights about the classification of the instance, every outcome descending from the root node corresponds to possible values for that variable [16]. We can classify an instance by starting from the root node and checking the attributes labeled on that node and moving down from that node based on attribute given attribute values [16] as shown in Figure 10.

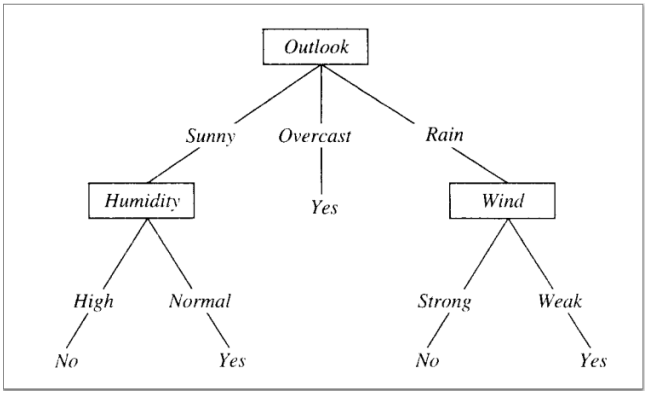


Figure 10: Example of Decision Tree Construction[16].

Advantages of Decision Trees:

- Decision Tree applications are easy to interpret and understand [15]. This ease comes from their schematic representation [15]. Interpretation between alternatives can be expressed with single numerical number which is the expected value (EV) [15].
- Decision Trees can handle noisy or incomplete data-sets [15]. In other words it requires little effort of data preparation because of it is flexibility [3].
- It can handle both nominal and numerical variables [15].
- It can be modified easily whenever the new information is available [15].

Disadvantages of Decision Trees:

- Because of it is a use of divide and conquer method they can demonstrate good performance if there are few attributes exists when the attributes level goes into large number decision tree become more complex which will result in poor performance [15].
- Decision Trees are also susceptible to training set which can give a result of over-fitting [15]. In other words, it can believe the training set completely which will give an abysmal performance on testing set.
- ID3 and C4.5 decision tree algorithms require discrete values as input data.

7.5 Ensemble Methods

Ensemble methods goes into classification algorithm category, they are learning algorithms which uses weighted vote for it is prediction methods, in other words, it is learning rules over a small subset of data then we combine these rules which we learn from the small subset of data to make predictions and/or classification on the testing data [4]. The originality of the Ensemble method comes from Bayesian averaging, but with the recent algorithms include "Bagging, error-correcting, and boosting [4]".

Bagging refers to simply the looking at data-sets and dividing the data-set to it is small subsets then learning the rules of that particular small subset. Next step is combining each learned rule from subsets to apply to more significant data set. Combining method mostly done with averaging the learned rules. Bagging also does better on testing set than standard Linear Regression analysis and linear regression does better on training set especially in third order polynomial [4].

Boosting is another method used in Ensemble Methods. The difference from bagging is in boosting we need to pick subsets or examples that we are not good at in other words hardest examples. Then we combine these learned rules with the weighted mean instead mean used in bagging method.

Boosting is little different then bagging.

Advantages of Ensemble Methods:

- Prediction of the ensemble methods is better than most of the algorithms because of the combining methods intuition makes the model less noisy [18].
- They are more stable than other algorithms. [18]

Disadvantages of Ensemble Methods:

- Over-fitting may cause some disadvantages for ensemble learning but bagging operation will reduce this overfitting [18].

8 FITTING DATA INTO MACHINE LEARNING ALGORITHMS

In this section, we will show the techniques we used on the execution of the prepared data into machine learning algorithms. Before fitting the data into the machine learning algorithms, we split the data into two sets. These sets are the training set and the testing set. We do splitting because of gaining an access of the future data will most likely be hard before future occurs, and because of this fact, it is a good idea to test our model with a dataset which our model has not seen it [19]. We also used K-Fold cross validation in order to avoid overfitting. We used scikit-learn for splitting data

into train and test we saved 25% of data for testing purposes. Our data also consist from 4 variables one which is price of the bitcoin will be used as our target variable.

8.0.1 ***Accuracy:*** Accuracy answers the question of how good is the model is. In our case this question will be out of all the price points, how many did the models classify the prices correctly. The mathematical expression of the accuracy is the ratio between the number of correctly classified points and the number of total points. We can think that if we have high accuracy, our model is excellent, but this is only where we have identical false positive and false negative values in our dataset [10].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

Our findings for each of the machine learning model related to their scores are shown in Figure 11.

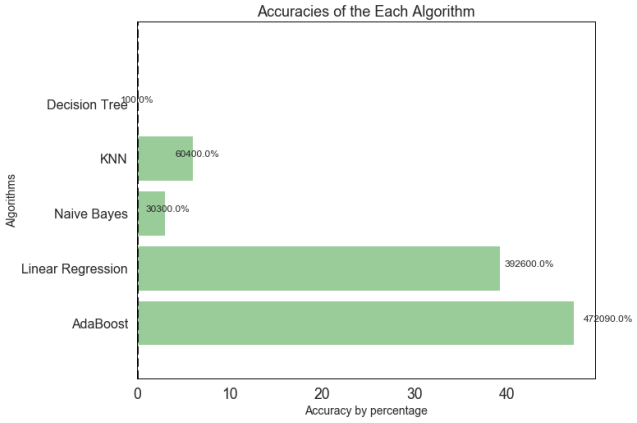


Figure 11: Accuracy of the each model.

As we can clearly see that Ensemble and Linear Regression models gave the best results among the other models.

9 CORRELATION TO ALTERNATIVE COINS

Bitcoin has become an industry staple in the field of Cryptocurrency. However, the market is also populated with a lot of other coins. The primary three other than Bitcoin are Ethereum, Bitcoin Cash, and Litecoin. While they do not have the massive amount of market share as Bitcoin, they do play a significant role in the market. However, what is the factor that Bitcoin plays in the price fluctuation of the alternative coins (Alt Coins)? To look at the possibility, price data was collected using the GDAX API ranging from 2/12/18 to 4/20/18 Figure 12. It is shown below that the Bitcoin trades at a much higher price than the rest of the coins combined. We can also see a correlation between the price of Bitcoin and other data.

When looking at data for Bitcoin, Ethereum, Bitcoin Cash, and Litecoin, there is a very strong correlation in the data as shown in Figure 13. However, even considering the change in price every six hours, it is shown to have a incredibly significant correlation.

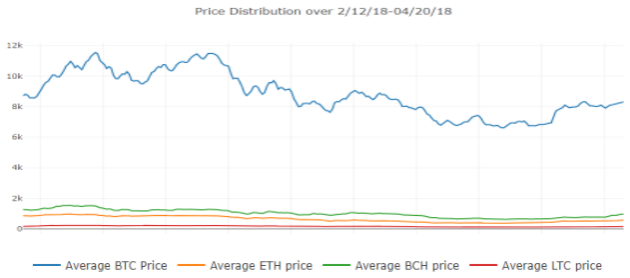


Figure 12: Price Distrubition.

9.0.1 *Heat Map.* Heatmap would be used to show variance in the data. These are most effective in small multiples, based on similar attributes and arranged in a row for easy comparison.

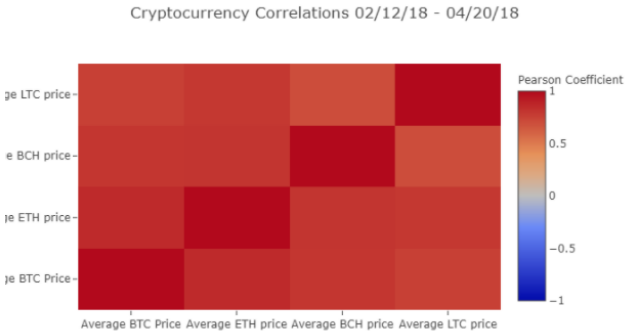


Figure 13: Correlation Heat Map for Bitcoin Prices.

The correlation ranged from .75 to .80 for all the Alt Coins as referenced below. This is indicating of a very strong correlation.

10 CHALLENGES AND OPPORTUNITIES

One of the issues that we faced was data collection. Initially, the project outline was to run the data collection suite for 14 days that collected tweets based on the keywords fiBitcoin, Ethereum, Litecoin, Bitcoin Cash, BTC, ETH, BCH, LTC, cryptos, cryptocurrency.fi However, that proved to be unrealistic due to its sheer size and call volume onto twitter servers. That led to the code quitting and resulting in problems. Therefore, the solution was to filter the keywords down to just fiBitcoin, BTC, and cryptocurrency.fi The issue also arises when data collected within a three day period.

The tweets data file, which is considerably large and comes in at roughly 20GB, ran into many problems when compiling. This is due to a problem we believe to be in the Twitter API which may not have written the file uniformly.

11 CONCLUSION

We didn't get substantial evidence that we can predict the fluctuation in the price of bitcoin based on the change in the polarity of the sentiment analysis of the users. However we found that the best machine learning algorithm to consider was Ensemble methods for giving the best accuracy of 47%.

Furthermore, we found some evidence on between the number of tweets and bitcoin price. Also, correlation between the coins proved to be very correlated so the change in price of one of the coin could be used to predict the change in other coins especially in bitcoin.

12 TEAM MEMBER CONTRIBUTION

12.0.1 *Borga Edionse Usifo.*

- Coding
- Cloud Computing and Data Collection
- Machine Learning Modelling
- LaTex formatting
- Writing
- Visualizations

12.0.2 *Shivam Kapadia.*

- Coding
- Bitcoin and Twitter Data Collection
- Visualizations
- Writing
- Machine Learning Models

12.0.3 *Sushmita Dash.*

- Data Collection
- Coding
- Machine Learning Models
- Writing

ACKNOWLEDGMENTS

The authors would like to thank Dr. Vincent Malic for his support and suggestions to write this paper.

REFERENCES

[1] T. J. Barker. 2017. Why is Bitcoin's Value So Volatile. Online. (Dec. 2017). https://www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp

[2] L. Ben. 2015. Six Reasons why I recommend scikit-learn. Online. (Oct. 2015). https://www.oreilly.com/ideas/six-reasons-why-i-recommend-scikit-learn

[3] B. Desphpande. 2011. 4 key advanatages of using decision trees for predictive analytics. Online. (July 2011). http://www.simafore.com/blog/bid/62333/4-key-advantages-of-using-decision-trees-for-predictive-analytics

[4] G. T. Dietterich. n.d.. Ensemble Methods in Machine Learning. (n.d.). http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf

[5] O. F. Ertugrul and M. E. Tagluk. 2017. A novel version of k nearest neighbor: Dependent nearest neighbor. *Applied Soft Computing* 55, Supplement C (2017), 480 – 490. https://doi.org/10.1016/j.asoc.2017.02.020

[6] M. A. Hassan, A. Khalil, S. Kaseb, and M. A. Kassem. 2017. Potential of four different machine-learning algorithms in modeling daily global solar radiation. *Renewable Energy* 111, Supplement C (2017), 52 – 62. https://doi.org/10.1016/j.renene.2017.03.083

[7] D. S. Jadhav and H. P. Channe. 2014. Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. *International Journal of Science and Research (IJSR)* 5, 1 (Jan. 2014), 1842–1845. https://www.ijsr.net/archive/v5i1/NOV153131.pdf

[8] B. Jason. 2014. A gentle introduction to Scikit-Learn: Python Machine Learning Library. Online. (April 2014). https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/

[9] J. Jiaqi and Y. Chung. 2017. Research on K nearest neighbor join for big data. In *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, Department of Computer Engineering Wonkwang University Iksan 54538, Korean, 1077–1081. https://doi.org/10.1109/ICInfA.2017.8079062

[10] R. Joshi. 2016. Accuracy, Precision, Recall, and F1 Score: Interpretation of Performance Measures. Online. (Sept. 2016). http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures

[11] J. Kunal. 2015. Scikit-Learn in python - The most important Machine Learnig Tool I learnt last year. Online. (Jan. 2015). https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/

[12] B. Mills. 2018. What is Cryptocurrency: Everythin You Need To Know. Online. (April 2018). https://blockgeeks.com/guides/what-is-cryptocurrency

[13] L. J. Moon. 2017. Fast k-Nearest Neighbor Searching in Static Objects. *Wireless Personal Communications* 93, 1 (01 Mar 2017), 147–160. https://doi.org/10.1007/s11277-016-3524-1

[14] G. Nick. 2014. KNN. Online. (April 2014). http://www.nickgillian.com/wiki/pmwiki.php/GRT/KNN

[15] C. Petri. 2010. Decison Trees. Online. (2010). http://www.cs.ubbcluj.ro/~gabis/DocDiplome/DT/DecisionTrees.pdf

[16] U. Princeton. NA. Decision Tree Learning. Online. (NA). http://www.cs.princeton.edu/courses/archive/spr07/cos424/papers/mitchell-dectrees.pdf

[17] M. Ray. 2012. Nearest Neighbours: Pros and Cons. Online. (April 2012). http://www2.cs.man.ac.uk/~raym8/comp37212/main/node264.html

[18] S. Ray. 2015. 5 Easy Questions on Ensemble Modeling Everyone Should Know. Online. (Jan. 2015). https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

[19] D. Steinberg. 2014. Why Data Scientist Split Data into Train and Test. Online. (March 2014). https://info.salford-systems.com/blog/bid/337783/Why-Data-Scientists-Split-Data-into-Train-and-Test

[20] E. STENQVIST and J. LONNO. 2017. Predicting Bitcoin price fluctuation with Twitter sentiment analysis. Web. (2017). http://www.diva-portal.org/smash/get/diva2:1110776/FULLTEXT01.pdf

[21] K. B. Tapan. 2015. Naive Bayes vs Logistic Regression: Theory, Implementation and Experimental Validation. *Inteligencia Artificial, Vol 18, Iss 56, Pp 14-30 (2015)* 1, 56 (2015), 14. http://proxyiub.uits.iu.edu/login?url=https://search-ebscohost-com.proxyiub.uits.iu.edu/login.aspx?direct=true&db=edsdoj&AN=edsdoj.0e372b34c5d48bcb72cd437eede1fd1&site=eds-live&scope=site

[22] K. Teknomo. 2017. K-Nearest Neighbor Tutorial. Online. (2017). http://people.revoledu.com/kardi/tutorial/KNN/Strength%20and%20Weakness.htm

[23] Wikipedia. 2017. Naive Bayes. Online. (Nov. 2017). https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[24] H. Zhang. 2004. *The Optimality of Naive Bayes.* resreport. University of New Brunswick. http://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf