Predicting Price Spikes in Digital Currencies

Authors:
Augie Doebling and Hans Schumann
Written for:
Cal Poly Senior Project, Winter/Spring 2018

Background of Project

Since its creation in 2009, Bitcoin has revolutionized how the modern world sees currency. As the first all-digital currency, Bitcoin has several aspects to it that made it unlike any other form of currency. First, and most obvious, is the fact that it is entirely digital. Second, that it is decentralized, meaning no governing agency or state has any power over it. Finally, because it is not tied to any physical bills, there is little to no physical influence to stay at a consistent price. These attributes lead to both some challenges and some advantages when compared to traditional currencies.

The fact that Bitcoin is both entirely digital and completely decentralized raises some security concerns. How is it possible to verify someone who says they holds some amount of Bitcoin actually holds that amount? Additionally, once they have spent that amount, how is it ensured they haven't "copied" it? These challenges were met with a technology pioneered specifically to handle this: blockchain. Blockchain is essentially a decentralized ledger of transactions that is cryptographically secure. For Bitcoin there is one blockchain ledger that has a record of every transaction to ever occur.

The non-physical nature of Bitcoin also led to a rather unusual phenomena. Because it was not tied to any physical currency, there is no pressure to stay at a constant price. For example, US currency has several different denominations of the dollar which don't change. While its value has gone down slightly over the last century due to inflation, it has gone down at a relatively constant rate. Barring some economic or political disaster, the dollar would never simply double its value overnight for no reason. Its value is based on the fact that its established, linked to a stable government, and is tied into physical bills. Bitcoin, on the other hand, has none of these ties, and therefore is much less stable.

Because there is no authoritative agency in charge of Bitcoin, its price is entirely based on what the public thinks it should be worth. If enough people think it is going to go up in price, the price really will go up. This is perfectly illustrated by what happened over the course of 2017.

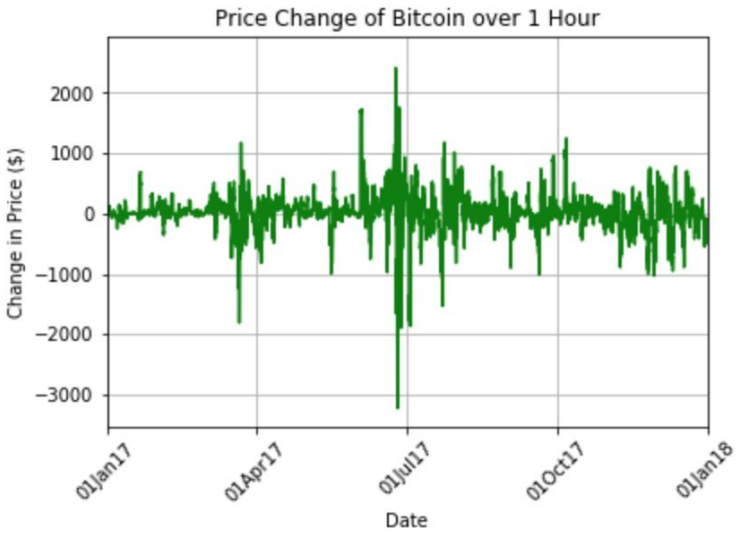

Figure 1: Price change in Bitcoin over one hour in 2017.

As seen in Figure 1, the price change over just a single hour can be huge. The biggest changes happened during July of 2017, with the price changing as much as $2,000 in a single hour. The boom in Bitcoin started small as some banks and governments were deeming Bitcoin transactions to be legal purchases. This started the ball rolling, and people got on board. The end of the year saw Bitcoin valued at 20 times what it was in January.

Initial Vision

Our initial vision for this project was to attempt to harness the wild nature of Bitcoin and see if we could make any statements or predictions about it. We initially thought about attempting to see what Bitcoin would do long term, however we determined that it was unlikely that it was possible to make meaningful long-term predictions about the future of Bitcoin. We instead hoped to make short term predictions about spikes in the value.

Modeling of this kind is actually not anything new. It has been in regular use for stock market prices for several decades. However, modeling a currency like Bitcoin turns out to be quite different. The price of individual stocks are tied to companies in the real world and have predictors such as sales or released financials Wall Street can use to determine a stock's worth.

Bitcoin, on the other hand, is tied to nothing. This meant we would have to predict its value based entirely on public opinion data. This gave us a unique challenge to our

project. The idea and the scale at which we wanted to do it at would be in relatively new territory and hopefully would give new and interesting results.

Goals

The main goal of this project is to develop a model that could be used in real time to determine whether or not to purchase Bitcoin. As stated above, we needed some way to capture public opinion to predict price spikes in Bitcoin. The agent for determining the public's view of Bitcoin that we decided to use was Twitter. We looked to use twitter data to quantify a public view of Bitcoin and thus develop a trading algorithm from this data that could profit from trading Bitcoin. If we could find a way to successfully capture changes in the price of Bitcoin, we could make a lot of money, which also motivated the project.

Data Collection

Starting off on this project, we wanted to look at different digital currencies and compare our ability to predict them using multiple predictors, including social media data and news data. First we worked on getting data from Bitcoin. Because of the excitement around this currency, it was easy to find this data on the internet. However, it proved harder to find the data in the precision we wanted it in. Specifically, we wanted the price from Bitcoin's beginning to the present in intervals of 1 minute. This data was found on Kaggle. Thus, we had minute-by-minute price of Bitcoin from December 1, 2014 to January 8, 2018, the day of collection.

The next step was collecting data from Twitter. This is the task that proved to be the most challenging objective for the project. When we started, we assumed that we would just be able to request the data from Twitter and download it. Unfortunately, it was not that easy. To access individual tweets you must go through an API. While the final code for this task ended up being very clean, it was difficult to find an efficient way to process this much data.

The API downloaded Tweets in chunks, which we had in length 100. For each Tweet in that chunk, we had to process that data into the format we wanted, and save it to our database running on Amazon Web Services. Our initial script took six hours to run on our laptops for each day of data collection. Since we wanted data for all of 2017, the run time was unfeasible. So we made some improvements and started running it on a AWS EC-2 server with high network performance. We also cut our collection parameters from collecting any tweet with the phrase "bitcoin" to only those with "#bitcoin". We also filtered out any tweets with hashtags or phrases associated with spam accounts such as

#freebitcion and #giveaway, since these were accounting for a significant portion of the data, and we felt they would have low credibility and would not be reliable.

After running our new program for a few days, we had collected around 6.5 million tweets in our database. Because we had spent a considerable amount of time getting this far, we decided it was more important to move forward with the modeling and live collection than to attempt to collect news articles.

Building the Model

Data Cleaning

After data collection, we had the price of Bitcoin by the minute. Additionally we had every tweet about Bitcoin, with its minute of posting. To combine these data to be used together, we aggregated the tweets by the minute and merge them with the price data, to get a better idea of how the tweets were affecting the price change of Bitcoin. The following steps were taken to achieve this:

1) Get the sentiment of each tweet using the TextBlob package in Python. (This gave a score from -1 to 1 to each tweet)
2) For each minute, count the total number of tweets, sum the total number of retweets and favorites, and take the average sentiment of all tweets.

Using this aggregated dataset, we were able to combine the price data of Bitcoin with the tweets on their commonality, minute. A display of the resulting dataset after aggregation and merging is shown in Table 1 below.

	key	id	datetime	currency	price	logprice	times	count	favorites	retweets	avg_sentiment	day_of_week
100000	100000	1166272	2017-03-30 15:40:00	bitcoin	1030.760728	6.938052	1166253	8	7	7	0.048295	Thursday
100001	100001	1166273	2017-03-30 15:41:00	bitcoin	1031.499326	6.938769	1166254	4	1	1	0.075000	Thursday
100002	100002	1166274	2017-03-30 15:42:00	bitcoin	1031.350672	6.938625	1166255	9	8	3	0.005556	Thursday
100003	100003	1166275	2017-03-30 15:43:00	bitcoin	1031.200647	6.938479	1166256	7	5	4	0.192857	Thursday
100004	100004	1166276	2017-03-30 15:44:00	bitcoin	1030.951612	6.938238	1166257	3	0	0	0.138889	Thursday
100005	100005	1166277	2017-03-30 15:45:00	bitcoin	1030.540000	6.937838	1166258	9	0	0	0.072222	Thursday
100006	100006	1166278	2017-03-30 15:46:00	bitcoin	1030.516410	6.937815	1166259	2	0	0	0.062500	Thursday

Table 1: A portion of the resulting dataset after merging Bitcoin price and aggregated tweet data

Each row in Table 1 represents one minute of the merged datasets. For each minute, from January 1, 2017 to January 8, 2018, we have the "price," along with the twitter variables: "count," "favorites," "retweets," and "average sentiment."

Exploratory Data Analysis

Table 2 shows descriptive statistics of the variables in dataset.

Variable	Mean	SD	Range
Price ($)	3562.58	3664.08	19148.71
Price Change ($)	0.037	12.80	3773.13
Count	12.02	10.28	435
Favorites	18.63	135.6	64053
Retweets	18.46	356.5	202337
Average Sentiment	0.075	0.100	2.00

Table 2: Descriptive Statistics of the minute-by-minute Bitcoin price and tweet data

As it can be seen in the table above, the price change from one minute to the next had a standard deviation of $12.80! This is an extremely high value, proving to add to the challenge of our goal in capturing some of the variation in this price change to develop a predictive model and algorithm for trading Bitcoin. The other variables are also shown, with high standard deviations, showing extreme variability in the frequency and characteristics of tweets at every minute.

To get a better idea of the relationship between the price change and tweet frequencies, we graphically displayed their values over the time that our dataset covered. Figures 2 and 3 below show the minute-by-minute price change of Bitcoin and the minute-by-minute number of tweets about Bitcoin from January 1, 2017 to January 8, 2018, respectively.

Figure 2: Price change from one minute to the next for Bitcoin from Jan 1, 2017 to Jan 8, 2018

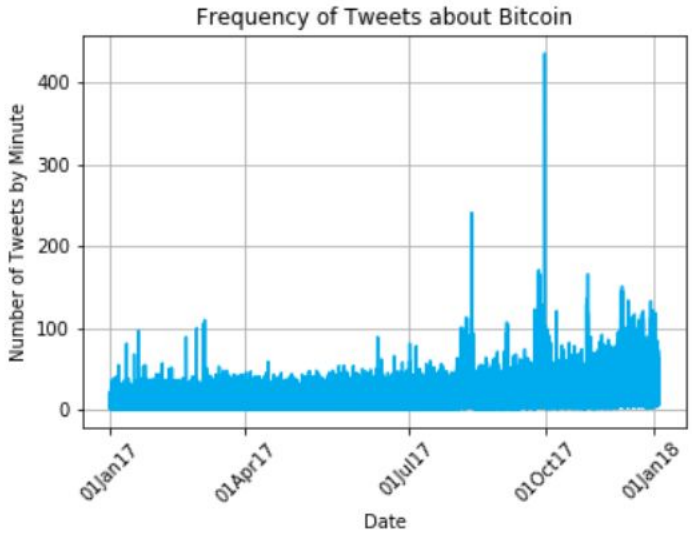

Figure 3: Count of Tweets by minute from Jan 1, 2017 to Jan 8, 2018

The price of Bitcoin shows a dramatic change around the month of November. Before November, the price changes from one minute to the next do not exceed more than $500, and are thus relatively small. Late in the dataset though, we can see that the price of Bitcoin becomes extremely variable. To our desire, a similar trend is present in the count of tweets. The number of tweets is fairly low at each minute before November, but then rises towards the end of the dataset. This apparent association between the number of tweets at each minute and the price changes gave us the idea that twitter could be used to predict price changes and develop a successful trading algorithm.

For the model that we would be attempting to use, the price change in Bitcoin over some time *t* would be the response variable of interest. We are interested in how tweet data is associated with how much Bitcoin changes in price. We chose to use the price change, and not just price because of possible autocorrelation between the prices at given minutes, whereas if the price at this minute is high, we would also expect the price at the next minute to be high. Price *changes* though are not as highly autocorrelated with each other. This can be expressed in Figure 4 below.

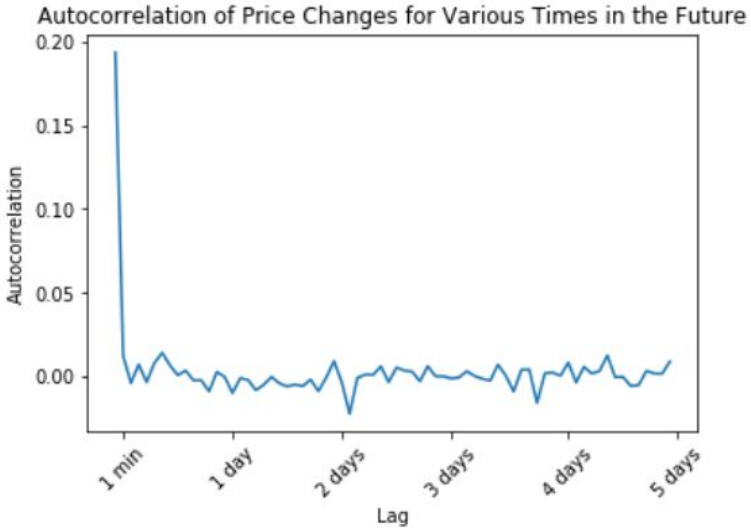

Figure 4: Display of autocorrelation of price changes up to 5 days in the future

The autocorrelation for price changes minute to minute is 0.194. After that though, the autocorrelation in price changes for larger time intervals approach zero. The autocorrelation between price changes one hour in advance is below 0.01, with this trend holding true as we look even farther into the future up to 5 days. This yielded the idea that any price change window past one hour in the future should be safe to use in a regressive analysis because we would have independence of the errors in price changes. Therefore, for all model building we attempted to predict the price change in Bitcoin one day in advance, allowing for us to still satisfy the independence assumption and get a useful model for trading. With an accurate prediction of the price of Bitcoin one day in the future, this would lend us the ability to purchase at one moment and sell 24 hours afterwards.

Feature Engineering

With the data only by the minute, the predictions from the tweet variables were not able to account for the time series nature of the fluctuating Bitcoin prices. This lead us to create variables that accounted for previous aggregated tweet data until a given time,

which could then be used for predicting a certain time *t* in the future. The first set of variables we created were summed aggregations of the four tweet variables from the current time to one day previously, or 1440 minutes into the past. This style of variable was created for one to two days into the past, and then two to three days into the past as well.

For example, the variable, “sum1440_count” is the total number of tweets about Bitcoin from the current minute to exactly one day into the past. “Sum2880_count” though, would be the total number of tweets from one day in the past, where we left off last, to exactly two days into the past, whereas there is no overlap between it and “sum1440_count” to eliminate the correlation between predictor variables. Lastly, “sum4320_count” would deal with the tweet amount two to three days previously. This method of aggregation by summing days into the past was done for not only count, but also the favorites, retweets, and average sentiment. For sake of simplicity, we looked at the day to day cutoffs up to three days into the past. For future use though, different window sizes could be used with an easy adjustment in the code.

Machine Learning Methods

The subsequent step after creating features is to fit models that could potentially be used for a Bitcoin trading algorithm. The y-variable we used was the price change of Bitcoin one day in the future, which would allow for an algorithm to optimize the purchasing of Bitcoin one day and selling it the next. The x-variables we used, which were described briefly above are listed below:

- Count, Favorites, Retweets, Avg_sentiment
- Sum1440_count, Sum1440_favorites, Sum1440_retweets, Sum1440_avg_sentiment'
- Sum2880_count, Sum2880_favorites, Sum2880_retweets, Sum2880_avg_sentiment'
- Sum4320_count, Sum4320_favorites, Sum4320_retweets, Sum4320_avg_sentiment
- Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

Four models were picked for a regressive analysis using the x-variables to predict the price change in Bitcoin for the next day: Linear Regression; K Nearest Neighbors; Random Forest; and Neural Network.

In order to get a good idea of the accuracy of the models we chose, we trained and tested the models on different sets of data. The training set was a random subset of 70% of the data, and the remaining 30% was used for testing.

After training and tuning the four models for their best accuracies, we observed the coefficient of determination for each of them when predicting on the testing set. The results, as well as the hyper-parameters, of this testing can be seen in Table 3 just below.

Model	Testing Set R^2
Linear Regression (18 / 24 vars.)	0.1170
K Nearest Neighbors (k = 5)	0.8850
Random Forest (depth = 25)	0.8490
Neural Network (1000,500,250,100,50)	0.7696

Table 3: Comparison of Regressive Models
Linear Regression: picked 18 of the 24 variables in a backwards stepwise regression
K Nearest Neighbors: used k = 5 nearest neighbors
Random Forest: used a maximum depth of 25 for all decision trees
Neural Network: 5 layers of sizes 1000,500,250,100,50

From the analysis of the models, it was found that the K Nearest Neighbors Regressive technique was the best for predicting the price change in Bitcoin one day in the future. The final model built was then saved for future use in the Live Application for trading. The K Nearest Neighbors was able to explain 88.50% of the variation in price change from one day to the next! This was extremely satisfactory to find, and showed that twitter data could be used to accurately model and explain the nature of Bitcoin's price fluctuation. The linear model only achieved a coefficient of determination of around 12%, which means that the KNN must have been capturing some organic, non-linear nature of the data. Figure 5 shown below displays the relationship of our predicted price change and the actual price change. The red line represents a perfect fit of the predicted and actual change in the price of Bitcoin in the next day. Observing the plot, it can be seen that the red line seems to go through the middle of all the blue points, with them hugging tightly around the line, supporting the fact that the KNN regression was a good predictor and could be used in the development of a trading algorithm and application.

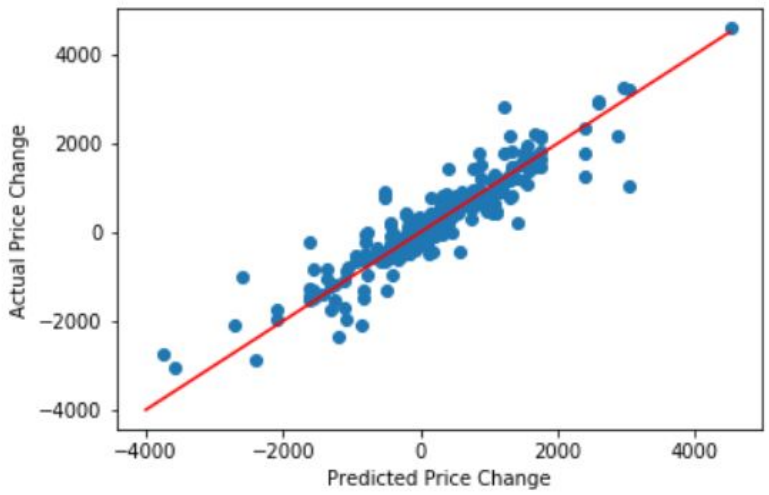

Figure 5: Actual vs. Predicted Price Change of Bitcoin for the KNN model

Model Analysis

One downfall of K Nearest Neighbors is its lack of interpretability. We only know that there is some relationship between days with a similar twitter behavior and the resulting change in Bitcoin price the next day. We obtained a seemingly accurate model, but did not have a direct way to discover which one of the created variables, or features, was the most influential in predicting the actual price change in Bitcoin. We did develop some way to get an idea though, using a form of cross-validation of the variables. With every predictor in the model, we had an R^2, or coefficient of determination, of 0.8850. This meant there was 11.50% of the random variation, or random error in price change, remaining that this model could not capture.

One way of showing how important a variable is in the model is to remove it and look at the percent of the random error remaining had that variable not been included. Essentially the equation: 1 - $R^2_{\text{reduced model}}$. The most influential variables would have the highest remaining random error, meaning that without them, we would be failing to capture more of the variation in price change.

Thus, we systematically removed all the variables one at a time, and determined the random error remaining with them missing. Figure 6 shows the results of this cross-validation like approach to finding the important variables.

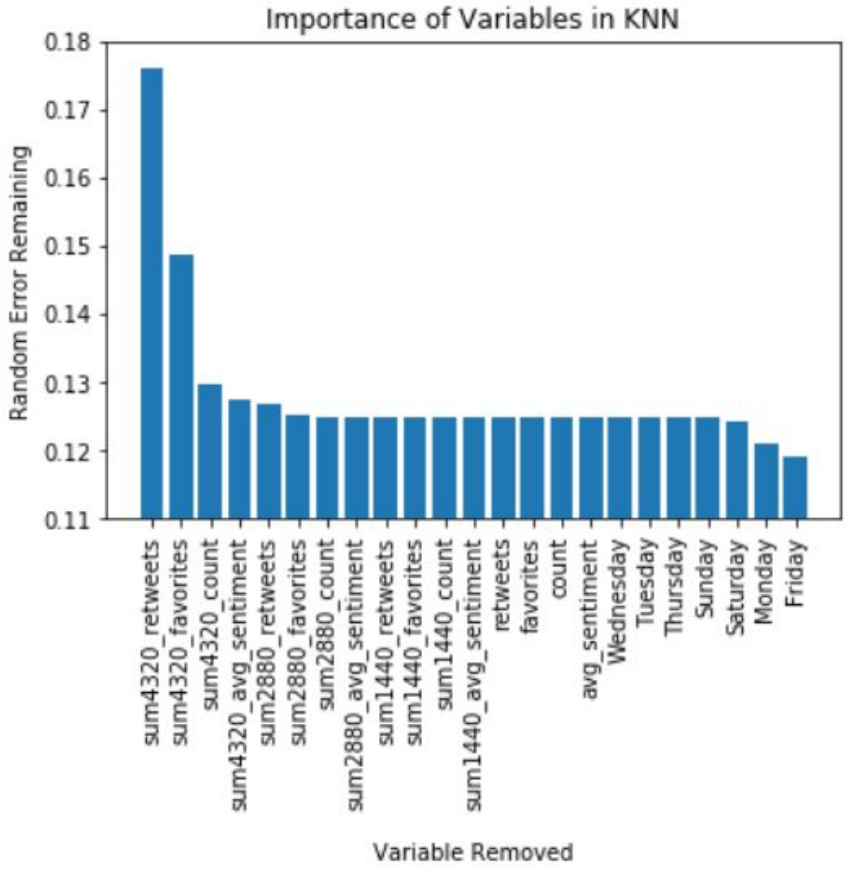

Figure 6: 1-R^2 for each variable removed in cross-validation

Figure 6 shows that the most influential variable in the KNN model was the number of retweets two to three days in the past. A possible explanation for retweets being the most important variable is that the number of retweets represents the validation of ideas and spreading them to new networks. This shared knowledge about Bitcoin, or simply just the validation and increased prevalence, could have an effect two or three days later in the actual market. This time elapse is an interesting note because the next three most important variables were all the aggregated variables from two to three days in the past, with favorites, count, and average sentiment in order. This means that there must be some delay in the twitter data and its association on the price change. We cannot say that there is a cause and effect relationship, but we can say that the variability in the twitter data two to three days in the past is highly associated with the price change in Bitcoin one day in the future.

Figure 7 shows relationship between the two most important variables and the price change. The red line represents the linear regression line between the two variables, while the black dotted line represents the line at a price change of zero, meaning no

relationship between the two variables. The red lines are both seemingly different from the black dotted line, further supporting the claims that the claim of their importance and relationship with the retweets / favorites two to three days in the past and price change.

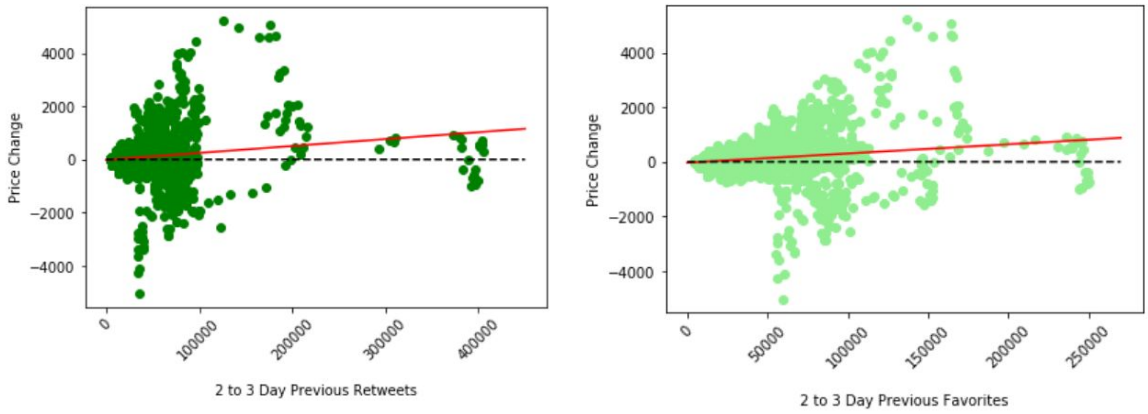

Figure 7: Plots of the Price Change by the two most influential variables

Creating the Live Application

Once we had the data collected and the model created, we needed to write an application that could get real time data and make predictions about what Bitcoin was going to do in the short term.

The application we wrote is hosted on a AWS server. While we were running it, everyday at a specified time (we used 9am), it would collect the tweets about Bitcoin over the last several days. Once it got those tweets into a usable format, it collected the necessary parameters need to pass to the model. Those being favorites, average sentiment, and several others and gave them to the model. The model would take these into account and make an expected change value. If this number was positive it was predicting the price to go up. If the number was negative, it predicted the price to go down.

After the application had an expected value, it would record the price of Bitcoin and whether or not it would buy with minimum price thresholds. The thresholds we used were 0, 100, 200. If the price was above 0, we would make a “purchase” for that threshold, and the same for the other thresholds. We did not actually buy any Bitcoin since we did not want to add purchasing functionality into the application.

The purchases recorded were saved to the database. Any purchases found in the next run of the application were sold and the change in price was recorded.

Results

The Live Application was successfully executed for five days. The results of what the KNN model outputted is compared to the actual price change in Table 4. Unfortunately, in our model testing phase of five days, we never had an expected change in the price change straying more than a couple dollars from -$11. Thus, we never had a day where our model would have purchased any Bitcoin, and instead kept our money, making precisely $0 in profit.

Date	Expected Change ($)	Actual Change ($)
May 31, 2018	-10.19	-111.26
June 1, 2018	-10.87	175.28
June 2, 2018	-11.87	109.14
June 3, 2018	-10.55	-219.12
June 4, 2018	-10.65	-46.63

Table 4: Results of the Live Application

Our Bitcoin trading algorithm at first sight appeared to be worthless, never attempting to purchase, but maybe a solid strategy would have been to never buy Bitcoin over these five days. The prices fluctuate a lot and it is always a risky investment. Thus, we can test whether or not this was a successful trading algorithm by comparing it to one that uses random chance. If we could produce statistically significant results that our algorithm was better than random chance trading, we would have evidence that the algorithm has some knowledge and meaning behind it.

Simulation was used to test the effectiveness of the Live Application's performance. For each iteration of the simulation, we essentially flip a coin (give a 50% chance) to determine whether or not to buy. If it says yes, we buy that day, and then sell the next, and record the money won or lost based on the actual price change. This is done five times, once for each day, and then we will find the net earnings after the fifth day. As with any simulation, we will repeat the process above many, many times and then view the distribution of overall profits. We can then find the proportion of those iterations that made a positive margin, above $0. This is the probability, or p-value, that a random trading algorithm would perform better than our KNN model. The desired p-value for us to state that our model is more successful than a random trading algorithm is below 0.10, meaning that there was a less than 10% chance of getting better results than we

did using random trading. Figure 8 below shows the distribution of all iterations of the simulation with a vertical line at zero representing the desired positive margin or profit.

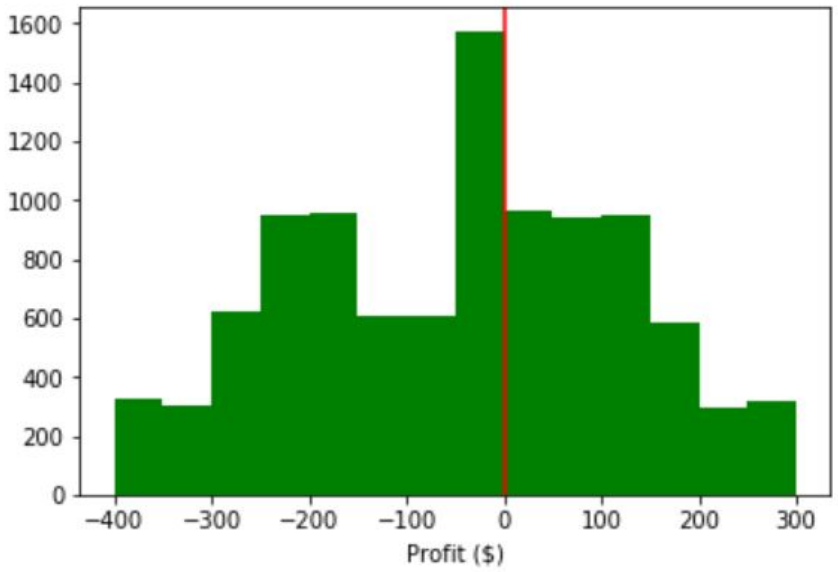

Figure 8: Histogram of the simulation results

The resulting p-value we obtained was only 0.3738. This means that given our five days, there was only a 37.38% chance that we would have done as well as we could have given that we used a random trading algorithm. This is inconclusive evidence, so we cannot say definitively that our model is truly better than flipping a coin on any given day, but there was a very small sample size and possibly with more data, we could increase our power and find true significance that our model is an effective one. The mean of the simulation was a loss of $47.27, so we did outperform the general average of the baseline, random trading algorithm.

Reflection

If we were to do this project again, I would have either scaled down the scope of the project, or done it over a bigger period of time. Overall we were somewhat successful about gauging the amount of time each step in the process would take except for data collection. Data collected took significantly longer than we were anticipating, and we could have spent a good deal more time in getting more and cleaner data.

Additionally, we came into this project not expecting to fully succeed. We knew this was a heavy senior project and that we likely would not be able to walk away with great results. It would have been a good idea to have smaller goals that we could have

leaned toward in the event that we were not as successful in our main objective. For instance, We could have set some side goals such as analyzing the trends in time series data between Bitcoin values and normal stock prices. Then, in the event we were less successful, we could have gotten some good deliverables at the end of the project.

Future Steps

The limits to this project are endless. Bitcoin and other digital currencies are still very difficult to grasp and the need and desire for models to accurately predict them is on the forefront of machine learning. With more time and effort put into this project, one way to really improve the effectiveness of using twitter to develop a trading algorithm is to improve feature engineering. The variables that we created to represent the nature of tweets about Bitcoin were very primitive, with simple aggregations, and a general package for sentiment. There are definitely more ways to account for the changes of tweet data, specifically, over time which could give more insight into the changing price of Bitcoin.

Along a similar path of accounting for the time aspect of this project. We eliminated much of the time series nature of the data by accounting for it in the features built. Another approach could be to use machine learning models that are more tailored towards time series data, like a forecasting package (arima in R), convolutional / recurrent neural network, or dynamic time warping KNN. These models are extremely labor intensive, but could give new outcomes and an even stronger model.

Additionally, much of trading is based on whether or not the price of Bitcoin is going to increase. This could be a new way to model the data where we can look at a binary response of whether or not Bitcoin will increase price, tailoring it toward the Live Application even more.

In conclusion, this project is merely just beginning. With more time, we could develop better algorithms using the existing and the forever compounding amounts of data. We could test the success of our current model and improve its functionality. Bitcoin is highly volatile as are the directions of work that could be done in the future on this project.

Links & Acknowledgements

Github Code Repository
https://github.com/AugieDoebling/DigitalCurrencies-SeniorProject

Collected Twitter Data
https://www.kaggle.com/augiedoebling/bitcoin-tweets

Thank you to our advisors, Dr. John Clements from the Cal Poly Computer Science department and Dr. Samuel Frame from the Cal Poly Statistics department. We are grateful to both of you for allowing us to run with our vision for this project. This project would not have been possible without your help and advice.

Sources
Bitcoin Icon designed by Freepik from Flaticon.