Cryptocurrency Price Prediction Project

By Will Stenzel, Max Hopley, and Shivam Kollur

Motivation

The overall goal of the project was two-fold, we wanted to understand what factors drove price movements in cryptocurrency prices and use this understanding to be able to develop a data-driven cryptocurrency trading strategy. The financial industry has placed an increasing focus on technical analysis (focusing more on using price patterns to find mispricing in the market), using the increasing capacity of AI and machine learning to identify statistical discrepancies. Given that the three of us are interested in data analytics in general, as well as in finance and blockchain, it seemed intuitive to take an emerging industry trend that we found interesting and to try to apply it to cryptocurrencies which are relatively new. Unlike traditional stocks whose prices depend partially on the companies value, cryptocurrencies have nothing backing them. This means they are more susceptible to changes in sentiment which we were hoping to take advantage of. Our hope for this project was to turn our insites from both the technical and sentiment analysis into a full-fledged trading strategy, similar to those employed by portfolio managers of quantitative analysis funds.

Background

The overall project goal was to be able to use technical indicators and sentiment analysis to predict 10-day price direction and price change (respectively). Although the initial idea was to create an overall model that combined the two analyses (technical and sentiment) to predict price changes, we found that the scaling of each dataset did not allow us to create a single overall time series dataset that included both technical and sentiment-based features. This was because the timeframe of Twitter data that we could access was much smaller than the timeframe of the technicals, and we did not want to compromise the effectiveness of the technical-based model by shrinking the size of its training dataset.

For the technical indicators, the idea was to use the Yahoo Finance API for a historical price dataset of Bitcoin that would be fed into functions from the ta-lib module (a library of functions that generate technical indicators) to create a dataset of technical indicators that would then be used to predict binary labels (1 indicating a positive price direction, and 0 indicating a negative price direction). The indicators used were moving averages, momentum indicators, cyclical indicators, volatility indicators, and Ichimoku trading signals. Moving averages represent a continuous plotting of price over time, momentum indicators illustrate the acceleration of recent price increases and decreases, cyclical indicators try to predict price peaks and price troughs, volatility indicators represent the statistical magnitude of risk of an asset, and the Ichimoku trading strategy is designed to present various buy and sell signals derived from a Japanese trading strategy based on moving average crossovers. After generating features and labels, we sought to import the sklearn module to use RandomForest, k-NearestNeighbor, and Gaussian Naive Bayes classifiers to evaluate the efficacy of technical indicators and comparing the ability of each classifier in predicting price direction.

For sentiment analysis, we aimed to use Twitter data to determine the public sentiment around Bitcoin at any point in time then compare that to the resulting price movement in Bitcoin. We used a sentiment analysis tool to quantify the sentiment of each tweet we had in our dataset. We also used hourly price data, from which we calculated percentage changes so we could easily compare sentiment scores (between -1 and 1) with the resulting percentage change over the next hour or more. Once we had this information we then were able to analyzer the correlation between sentiment and price change.

Experimental

Technical Analysis

For the technical model, we used a "pandas_datareader" and "fix_yahoo_finance" modules from Github to access Bitcoin historical price data from Yahoo Finance. With an array of historical prices, we used methods from the ta-lib module to generate technical indicator values, and appended the resultant arrays of technicals into an overall DataFrame of features using numpy and pandas. We then generated a column of binary labels by iterating through the historical dataset and identifying for each date (represented by a row) whether or not the price was higher in 10 days. This label DataFrame was appended to the feature DataFrame that we generated earlier to create one large overall DataFrame. With this DataFrame, we plotted a couple graphs to try to find relationships between some indicators and our label set to test our hunches on which indicators would be most useful. Below is an illustration of one of the Ichimoku signal graphs.

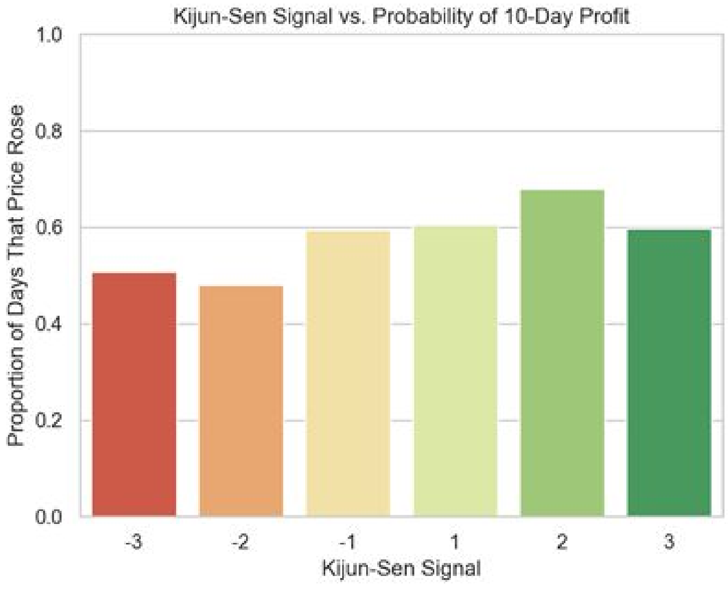


Figure 1. Horizontal axis represents Kijun-Sen Signal with -3 being a strong bear signal, -2 being an ordinary bear signal, -1 being a weak bear signal, 1 being a weak bull signal, 2 being an ordinary bull signal, and 3 being a strong bull signal. Buy signals are positive and indicated in green and sell signals are negative and indicated in red with magnitude and shading representing strength of the signal. Vertical axis represents the proportion of days with each signal that the value of Bitcoin was higher in 10 days.

We then used the sklearn module to access the classifiers and begin training, testing, and evaluating each classifier. Prior to training our data, we split our overall DataFrame into the feature and label sets (X and Y, respectively) and applied the preprocessing.scale() method to the feature DataFrame. We then used the ShuffleSplit method from sklearn.model_selection with 5 splits to cross validate our results and repeated this process 5 times, averaging results. After deploying each classifier, we found that the RandomForest Classifier outperformed the k-NN and Gaussian NB classifiers with an average accuracy of 0.85 and an average AUC of 0.85. These results seem to indicate that technical indicators are useful in predicting price directions, and that the RandomForest Classifier is best able to use the technicals to accurately predict price trends. Complete score comparisons are plotted below:

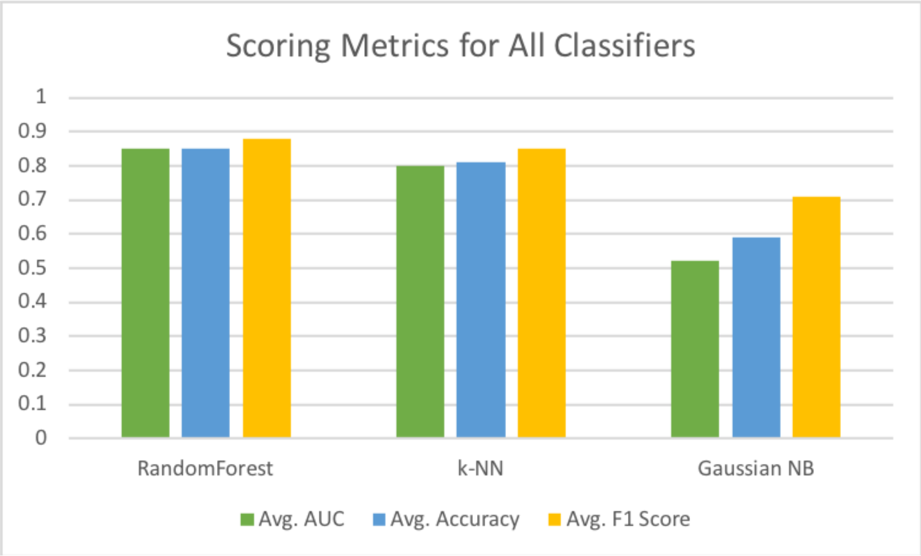


Figure 2. Horizontal axis contains three categories that indicate classifier type, with three bars that represent different metrics (AUC, Accuracy, and F1 Score from left to right). Vertical axis represents concrete value on a 0 to 1 scale.

Sentiment Analysis

Collecting applicable Twitter data was difficult for sentiment analysis. Twitter sentiment data for a single topic can be hard to find. Our original tweet dataset only spanned over a single day but after some looking we managed to find a dataset that had Bitcoin-related tweets that spanned over a year and three months. We then used the Vader sentiment library to analyze the sentiment score of each tweet. In order to match the time frequency of price data which we were going to compare it too we computed the mean sentiment score over each our and normalized our results. Next we put out data into a KNN model to get a baseline for our results. The results of this showed a very high mean squared error so we decided to go back and see if our sentiment score had any correlation to the price by checking the correlation. Given that there tends to be a delay between the change in the sentiment and corresponding the fluctuation of the price, we decided to plot the correlation over different hour offsets to see which amount of delay that would cause the highest correlation. The results from the sentiment analysis was not quite what we hoped for. With a max correlation score of 0.07 at about the 20 hour mark, we could conclude there was essentially no correlation between sentiment around Bitcoin and the change in its price (Figure 3). Along with this line graph, we also created a scatter plot of the percent change for the hour versus the sentiment score for that hour (Figure 4).

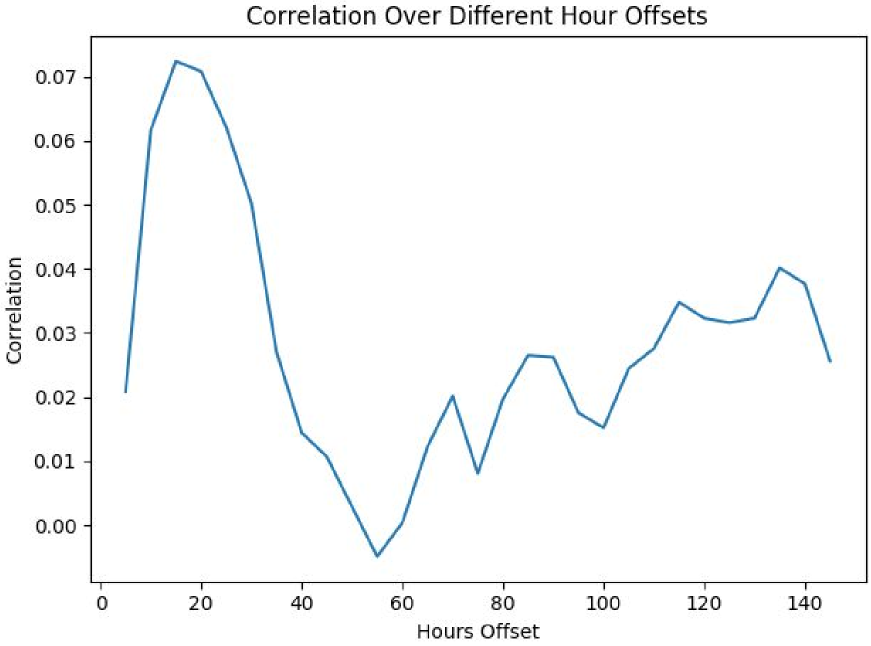


Figure 3. We examined how the correlation changed as we compared sentiment to various time periods of price changes. Though the correlation never really became significant, it clearly peaked at around 20 hours before plummeting. It rose again gradually once we began looking at price changes over the next 50+ hours.

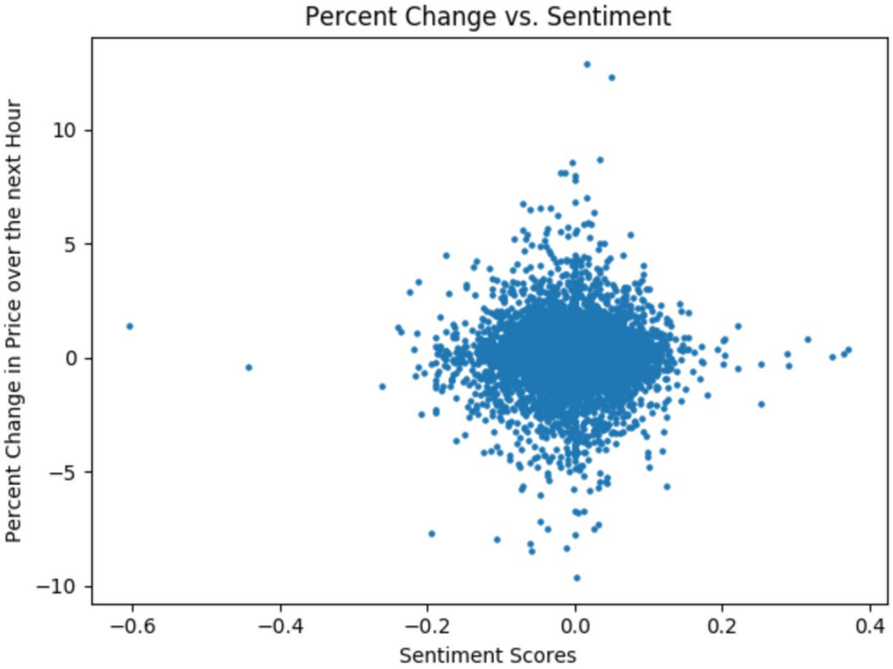


Figure 4. Each point on the scatterplot represents one hour of tweets from our dataset and every hour is included. The x axis indicates the average sentiment score of the tweets, and the y axis shows the percentage change in the price over the next hour.

We narrowed the potential cause(s) of the lack of correlation down to a few problems. The first potential issue was that our sentiment scoring tool (vaderSentiment) was designed to gauge general sentiment so it would miss some words that were clearly positive or negative in a financial setting (eg "buy" and "sell" and "bullish"). Another possible issue was that our dataset of tweets included some tweets that should not have been included or were potentially irrelevant. It appeared that some tweets didn't actually have to do with Bitcoin, and many certainly weren't forward-looking recommendations as we would've liked. If the dataset only included tweets that suggest whether or not Bitcoin should be bought at that point in time it would be much more useful for our purposes. The dataset that we looked hard to find unfortunately had many "noisy" tweets unrelated to Bitcoin or about Bitcoin's past performance with no regard to the future, which would not influence anyone to buy Bitcoin and therefore drive up the price. It would also have been better if we only used tweets from influential accounts, or at least weight each account according to its influence by using its follower count to estimate how influential the account is.

Conclusions

In all, we found that the technical indicator model had strong potential as a means of creating a system of buy/sell triggers, with the RandomForest Classifier being the most effective of the three tested classifiers. This knowledge, armed with the ability of the RandomForest Classifier to predict the probability of different classes (in our case an upward price movement or a downward price movement) would make it very simple to develop an executable strategy that buys a cryptocurrency if the probability of price increase exceeds a certain threshold and sells if the probability of decrease exceeds a certain threshold.

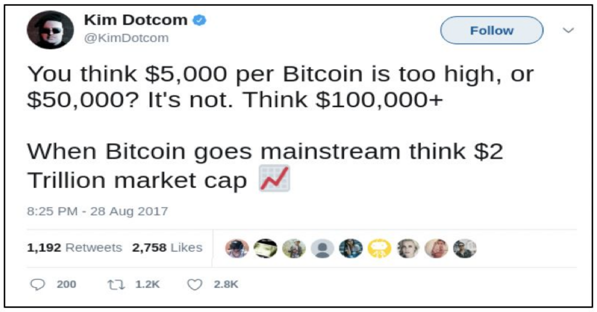
Although the sentiment analysis portion of our project was not successful in predicting price changes in the end, we learned a huge amount and have discussed some changes that we could implement that may potentially lead to a successful Bitcoin prediction algorithm based on Twitter sentiment. In the future, regarding sentiment analysis, we'd like to change our process of sentiment analysis. Studying what tweets and types of tweets resulted in the biggest price changes should be our starting point for this, as opposed to making assumptions about what we think should be classified as positive or negative sentiment. To do this we would use specific words as features, train a model and then use feature selection to see which words have the most impact on price. From there we could reverse engineer our own sentiment analysis algorithm that we could use to analyze the most current tweets. It would also improve our project if we could narrow down our tweet dataset to just the relevant tweets and then weight those tweets based on how many followers they have, how many likes/retweets the tweet gets, or how many people view the tweet, or a combination of all three of those options. If we implement those changes, we still have confidence that we could create a successful price prediction model based on Twitter sentiment.

References

- https://www.ichimokutrader.com/signals.html
- https://finance.yahoo.com/quote/BTC-USD/
- https://mrjbq7.github.io/ta-lib/funcs.html
- https://medium.com/mlreview/a-simple-deep-learning-model-for-stock-price-prediction-using-tensorflow-30505541d877
- https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
- http://cs229.stanford.edu/proj2017/final-reports/5212256.pdf
- https://www.cryptodatadownload.com/index.html
- https://www.kaggle.com/augiedoebling/bitcoin-tweets
- https://github.com/cjhutto/vaderSentiment

Appendix

Visualizations of sentiment analysis process:



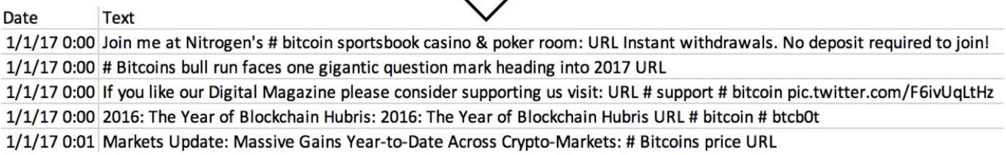
Kim Dotcom
@KimDotcom

You think $5,000 per Bitcoin is too high, or $50,000? It's not. Think $100,000+

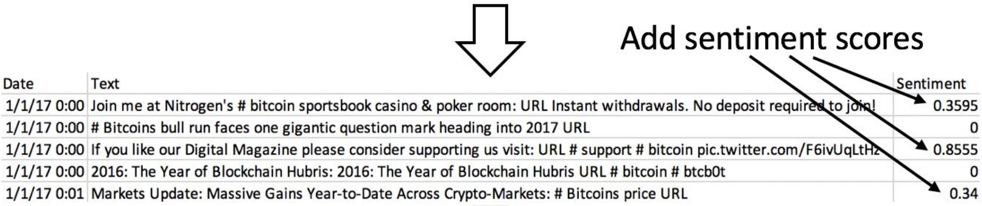
When Bitcoin goes mainstream think $2 Trillion market cap 📈

8:25 PM - 28 Aug 2017

1,192 Retweets 2,758 Likes

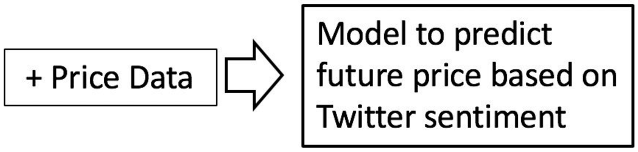


Date	Text
1/1/17 0:00	Join me at Nitrogen's # bitcoin sportsbook casino & poker room: URL Instant withdrawals. No deposit required to join!
1/1/17 0:00	# Bitcoins bull run faces one gigantic question mark heading into 2017 URL
1/1/17 0:00	If you like our Digital Magazine please consider supporting us visit: URL # support # bitcoin pic.twitter.com/F6ivUqLtHz
1/1/17 0:00	2016: The Year of Blockchain Hubris: 2016: The Year of Blockchain Hubris URL # bitcoin # btcb0t
1/1/17 0:01	Markets Update: Massive Gains Year-to-Date Across Crypto-Markets: # Bitcoins price URL



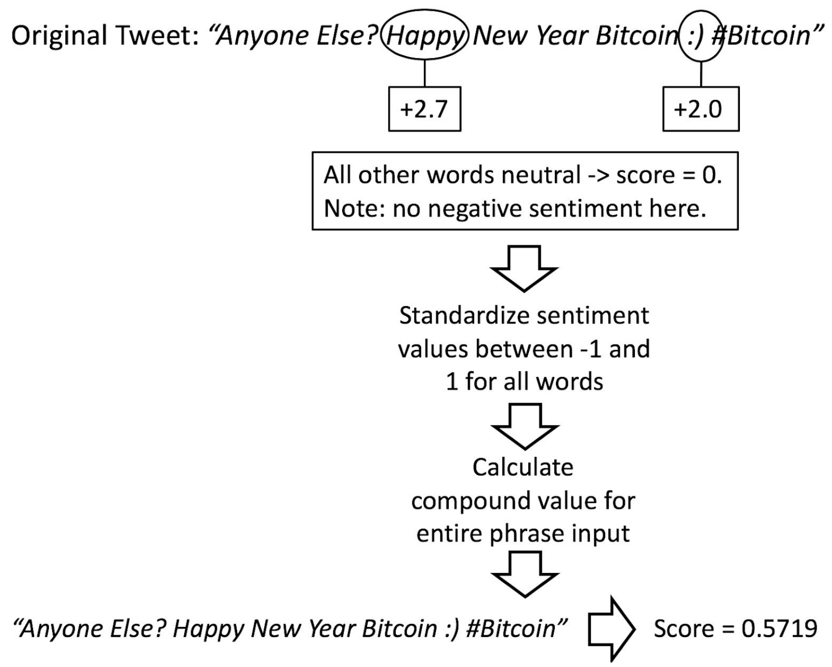
Add sentiment scores

Date	Text	Sentiment
1/1/17 0:00	Join me at Nitrogen's # bitcoin sportsbook casino & poker room: URL Instant withdrawals. No deposit required to join!	0.3595
1/1/17 0:00	# Bitcoins bull run faces one gigantic question mark heading into 2017 URL	0
1/1/17 0:00	If you like our Digital Magazine please consider supporting us visit: URL # support # bitcoin pic.twitter.com/F6ivUqLtHz	0.8555
1/1/17 0:00	2016: The Year of Blockchain Hubris: 2016: The Year of Blockchain Hubris URL # bitcoin # btcb0t	0
1/1/17 0:01	Markets Update: Massive Gains Year-to-Date Across Crypto-Markets: # Bitcoins price URL	0.34



+ Price Data

Model to predict future price based on Twitter sentiment


Original Tweet: *“Anyone Else? Happy New Year Bitcoin :) #Bitcoin”*
+2.7
+2.0
All other words neutral -> score = 0.
Note: no negative sentiment here.
Standardize sentiment values between -1 and 1 for all words
Calculate compound value for entire phrase input
“Anyone Else? Happy New Year Bitcoin :) #Bitcoin”
Score = 0.5719