Prediction of Bitcoin Prices

Aashray Yadav | Sai Kannan | Mubarak Abdul | Nicolas Sarquis | Bhavya Vashisht

Advisor: Alexander Fred Ojala, *Visiting Research Scholar, Sutardja Center for Entrepreneurship, UC Berkeley*

Final Project Report
December 8th, 2017

IE290 - Applied Data Science

Table of Contents

1. Project Overview

Bitcoin price was the most talked about item in the world in the last quarter of 2017. Why? It had appreciated 1000% in value from the beginning of the year! And 64% in November alone, with trends suggesting that it will close at 2000% its original value at the end of the year. The reason behind this meteoric rise? Anybody's guess would've been as good as ours. Sentiment seems to the main factor thrusting the price up at breakneck speeds. This project looked at ways to understand this "beast".

2. Problem

The world of cryptocurrencies is a mystery. Unlike fiat currencies and stock prices, they have no real asset backing the price at which they are being traded. Bitcoin in particular is extremely volatile and unpredictable. Our objective is to find a way to make it otherwise.

3. Approach

3.1. Data Analysis

The data of Bitcoin price data for the last four years was downloaded from CoinDesk. Once we had the data, we wanted to understand more about what relationship the prices have with various stock indices and different cryptocurrencies.

We obtained the closing price data for major stock indices such as Nasdaq, S&P 500 and Dow Jones for the last three years. The challenge with this part came in when we identified that the stock market doesn't function during weekends and public holidays while the crypto-market does. After considering two options,

1. Create pseudo data for the stock indices to fill up the missing indices
2. Drop rows from bitcoin prices corresponding to weekends and public holidays

After consulting with our advisor on what approach would give us better results and discussing the ease of implementation, we chose the second option. We adjusted the bitcoin closing prices to match with that of stock indices. This data was then used to find the correlation of each of these indices with Bitcoin. The results are plotted in Fig. 1.

	Bitcoin	Nasdaq	S&P 500	Dow Jones
Bitcoin	1.000000	0.707971	0.688724	0.771410
Nasdaq	0.707971	1.000000	0.989797	0.968166
S&P 500	0.688724	0.989797	1.000000	0.978780
Dow Jones	0.771410	0.968166	0.978780	1.000000

Figure 1 - Correlation Matrix of Bitcoin with US Stock Indices

The insights we gained out of this was that the Dow Jones index had the highest correlation of 77.14%.

We also analyzed the correlations between various alternate coins and Bitcoin and found a stunning correlation between them in almost every case. It was almost as if Bitcoin was the market index and all the alternate coins were merely following suite. A total of 16 alternate coins were studied and the following were the correlation values obtained:

Coin Name	Correlation w.r.t. Bitcoin
Ethereum	0.946
Nem	0.956
Iota	0.846
Litecoin	0.955
Monero	0.955
Neo	0.874
Numeraire	-0.258
Omisego	0.909
Qtum	0.626
Bitcoin Cash	0.592
Bitconnect	0.972

Dash	0.973
RIpple	0.843
Waves	0.928
Startis	0.861

3.2. Basic Prediction Models

As it is rightly said - Simplicity is key - which is why we ran existing prediction models on our data set. This wasn't completely straightforward because we had to extract only a handful of key features and then condense all of these down to one single feature which was then correlated with the price movement of Bitcoin. By running Bayesian and Linear Regression on the data, we got a reasonably lower MSE value of ~0.2 with Linear Regression. Upon applying classification algorithms such as SVM, KNN, LogReg and DTree, we achieved the highest accuracy of 69.7% with Logistic Regression.

3.3. Google Trends and Digital Wallets

Upon exploring the different parameters that could affect bitcoin prices, we came across an article that talked about the relationship between google trends data and the bitcoin price. So we set out to gather data for the number of hits for the search term 'bitcoin' over the past 5 years. Data cleaning was a major part here as we weren't getting the data in consistent frequencies. We used the gtrendsR package in R to retrieve the data from the website and then analysed it in python. We then did time-series analysis by lagging the data by 5 days and found a prediction accuracy of 73%. To add more features to this model we decided to include data about the number of digital wallets and we improve the accuracy to the model to 82%.

3.4. Sentiment Analysis

3.4.1. Investfeed using NLTK

We wanted to explore the influence of what people say on the price of Bitcoin. But as opposed to choosing news articles as our input, we decided to focus on online Bitcoin communities. To begin with, we decided to scrape investfeed.com for our corpus. We used Beautifulsoup to scrape 2500 posts, structured them into a dataframe and created a bag of words model. After converting the sentences into features, we used a Random Forests Classifier with a 80:20 train-test split. A training accuracy of 98% and a test accuracy of 60% only was achieved. We

believe that the model had a lower validation accuracy because of the class imbalance (60:40 split between positive and negative sentences instead of 50:50. Further, the dataset was too small - only 4 months worth of data - 2500 posts because the site was created only in August 2017.

3.4.2. Bitcointalk using NLTK

To improve the model we had with investfeed, we decided to scrape bitcointalk.org which had a lot more data than investfeed. We modified a scraping code to get all the posts and their respective replies for a 1 year period and scraped around 18000 posts. The challenge in this phase was finding a relevant scraping code, understanding how it worked and modifying it to suit our requirements. All this data was fed into a json file. We learned how to properly index values from this file which was essentially a dictionary with nested lists and dictionaries as well. We used NLTK to create a bag of words model. After converting the sentences into features, we used a Random Forests Classifier with a 80:20 train-test split. A training accuracy of 95% and a test accuracy of 75.4% was achieved when lemmatized posts were used. This training set also had a class imbalance, about 68:32. We believe that if this is fixed, it would significantly boost the accuracy of our model.

3.4.3. Vader

In addition to taking the conventional NLP approach, we utilized VADER. VADER comes with its own Lexicon but we still needed to add and remove certain words to make it work more effectively with our data corpus. With VADER, we assigned positive, negative and neutral values to each sentence in the corpus and then computed a single ‘compound’ value which was an indicator of the overall sentiment of that sentence. We think that adding this value to our existing value would surely boost the accuracy and performance of the model. It is something we will be working on in the near future.

3.5. Trading Algorithms

To identify the value of the models previously mentioned we simulate one year of investments according to what our models predict, and compare them with a traditional trading algorithms called Bollinger Bands. This model had the best results with a return rate of 1.4. We only were able to do the backtesting with the classification models because we wasn’t able to build a good strategy using continuous models predictions, and those models were the ones with better accuracy.

4. Future Course

4.1. Micro-level Twitter Analysis

Since there is no inherent legal tender status for cryptos and since they are decentralized, the cryptocurrencies derive their market prices from the perceived value attached to them and the underlying blockchain technology. A logical conclusion hence was to look at Bitcoins from a crowd behaviour perspective. This could most easily be done via web-scraping on a popular social media with public data, e.g. Twitter.

For performing analysis on Tweets, we planned to build a working NLP model on individual tweets along with the corresponding Bitcoin prices available nearest to the Tweet timestamp. The first approach was to use the Twitter API. We made developer account on Twitter and obtained the keys to start interacting with Twitter servers. While the Twitter API was excellent for post tweets or getting tweets/friends or a particular user, we decided we needed to stream the Tweets in order to get access to real-time tweet data. We utilised Tweepy for that.

Once we had access to real-time Tweet feed, we needed historical Tweet data in order to train an NLP model. The Twitter API only grants access to one week's worth of data historically, so we had to turn to Twitter search page. Twitter has indexed all its public Tweets, and so it is possible to search for a particular hashtag like #Bitcoin from as back as 2009. We decided we had to scrape from Twitter's search results in order to get historical data beyond one week.

The first approach was to use Beautifulsoup in combination with the in-built urllib library to open and scrape the web page. However, due to AJAX, we needed a library that supports key inputs for "infinite scrolling". We opted to use Selenium. Using Selenium and Pandas, we have now built a working model for scraping historic tweets ad infinitum, the only constraint being the computer on which the notebook is being run. Which is a very tangible constraint, as the tweets now have a lot of media like hyperlink, images and videos, which once loaded by the browser, continues to occupy memory space. The largest successful run so far is for 2,000 page scrolls yielding 3,486 tweets with corresponding timestamps.

Since we wanted to bypass the physical memory limitation, we attempted to run the same script on Berkeley Datahub, but we were unable to install Chromedriver required for Selenium to run in the PATH folder. We are looking of more ways to obtain historical tweet data.

5. Contributions of team members

Sl.No.	Name	No. of Hours Contributed
1.	Sai Kannan Sampath	110
	● Gathering data of bitcoin prices, stock market indices, alternate coins, and google trends data ● Helped build preliminary models ● Established correlations between stock indices and alt coins ● Explored google trends data and time-series analysis ● Helped with scraping data from the internet for NLP modules	
2.	Mubarak Abdul Kader	115
	● Performed correlation analysis for Bitcoin with top stock indices ● Read and understood research papers that showcased working models for Bitcoin price prediction ● Researched on open source crawling code and developed Python script to scrape both investfeed.com and bitcointalk.org ● Used NLTK to perform sentiment analysis and built a model that could predict Bitcoin price fluctuation with 75.4% accuracy	
3.	Aashray Yadav	95
	● Finding the ‘right’ data set ● Executing Predictive and Classification models and extrapolating insights for utilizing the content moving forward ● Locating rich textual sources for performing NLP such as investfeed, coindesk and bitcointalk ● Extracting posts and news articles from coindesk and Bitcointalk ● Exploring the possibility of computing sentiments of statements using VADER	
4.	Bhavya Vashisht	90
	● Performed stock-wise correlation analysis with BTC for major S&P 500 Information Technology stocks ● Developed the Python script for interacting with Twitter API ● Developed script for real-time Tweet streaming using Tweepy ● Scraped data from the Twitter webpage using Beautifulsoup and in-built URL reading Python library ● Developed script for continuous scraping of Twitter webpage by utilizing the AJAX infinite scrolling on the Twitter web-page with Selenium	
5.	Nicolas Sarquis	95
	● Did Machine Learning for Trading in Udacity and Introduction to Cryptocurrency Trading from Blockchain at Berkeley. ● Find and pre processed data about cryptocurrencies prices and data about	

	the number of digital wallets. ● Build trading algorithms like Bollinger bands. ● Simulate the returns of each one of our models to compare between them. ● Build continuous models using as features: different cryptocurrencies past prices, number of digital wallets and gtrends.

6. References

1) Kim, Y. B., Kim, J. G., Kim, W., Im, J. H., Kim, T. H., Kang, S. J., & Kim, C. H. (2016, 08). Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies. *Plos One,11*(8). doi:10.1371/journal.pone.0161197
2) Kim, Y. B., Lee, J., Park, N., Choo, J., Kim, J., & Kim, C. H. (2017, 05). When Bitcoin encounters information in an online forum: Using text mining to analyse user opinions and predict value fluctuation. *Plos One, 12*(5). doi:10.1371/journal.pone.0177630

7. Appendix

1. https://github.com/bhavyavashisht/Data-X-Project
2. https://github.com/nasarquis/Prediction-Model