ETL Project:
The Bitcion Market and News Media

Alex Jose, Davis Trinh, and Jing Duan
The Data Analytics Boot Camp at The University of North Carolina at Chapel Hill

(Dated: August 17, 2019)

ABSTRACT

Created in 2009 by Satoshi Nakamoto, Bitcoin is the worlds first cryptocurrency. Bitcoin is a decentralized digital currency in which transactions are verified using cryptography and recorded on a public digital ledger known as a blockchain. In recent years Bitcoin has gained widespread media attention for its exponential growth. It has been acclaimed as the future of currency. It has also been criticized as this centurys greatest bubble. Cryptocurrencys rise in popularity (or notoriety), have also led to an increase in frequency of article publications on the subject. Our question of interest is if these articles affect market participation and price of Bitcoin. To answer this question we obtained the daily price, volume, and google trend for Bitcoin from 2014. We then took relevant articles published by the New York Times and calculated the sentiment using a natural language processing library and compared the sentiment with the quoted data. Our results were then loaded onto MongoDB.

I. DATA EXTRACTION

A. New York Times Articles

We used the New York Times article search API to retrieve articles that mention bitcoin. The search API by default returns all article information for articles that mention bitcoin in the header, abstract or within the article tags. The API response was in the json format. The information we wanted to extract was publication date, headline name and article body. The article body was not included within the response, so we limited ourselves to extracting the article abstract. The API responses were limited to 10 articles per request. The meta information from the initial request indicated that there were a total of 1500 articles matching our query. The first 1000 articles were extracted by looping through an optional page parameter in the API request. The last 500 articles were extracted similarly, but it required formulating a second request loop with a new end date due to a pagination limit on the previous api loop. The resulting object was a list of dictionaries for each article. We decided to extract the article url as well, incase we needed further analysis on the article bodies themselves.

B. Bitcoin Price and Volume

Using Coinpaprikas API we attempted to grab Bitcoins historical prices and volume from 2011. However, due to the infancy of the asset at the time, we were unable to obtain accurately reported data from Coinpaprika. We settled for quotes from early 2013. We encountered an error in our query using the default parameters in which a request for 1000 results only yielded 150 results. This was resolved with relative ease by adjusting the parameters. We adjusted the default parameters of the API query to request for 5000 results (max) of BTCs opening price everyday at 00:00:00Z from the start of 2013. The quotes we received were stored in JSON format.

C. Google Search Trends

For the Google Trends data we originally manually downloaded the .csv from the Google Trends site. Then we found we only can get the weekly data if time range is over 6 months. So we were trying to PyTrends, the unofficial Google Trends API. PyTrends is a Python library that scrapes Google Trends by using custom parameters. PyTrend API has a limit that 1,400 sequential requests of a 4 hours timeframe. After it had been tested, we set up 60 seconds of sleep between requests in order to get the correct amount once you reach the limit. Finally we got 2000 daily bitcoin search popularity from 2014 to 2019.

II. TRANSFORMATION AND MERGING

A. Article Transformations

Since the relevant article fields were selected at the time of extraction, no fields needed to be removed. Although we considered dropping articles that did not contain bitcoin explicitly in the title, we decided this filter was too aggressive, as it dropped many relevant articles. We used the natural language processing library, TextBlob, to apply sentiment analysis to our articles. We applied a model trained by a naive bayes classifier on movie reviews to each headline and appended the sentiment field to each dictionary entry. This gave us a probability that the article headline was a positive headline. These sentiment values do not appear to accurately reflect the sentiment in the headline in most cases. Ideally we would like to use a model trained specifically on article headlines, or train our own network.

B. Bitcoin Price/Volume Transformations

Using Pandas, we loaded the BTC historical quotes as a dataframe. We then replaced any 0 values with NaN. From there we added the closing price of BTC by taking the opening price of BTC for the next calendar day. We then calculated the percent change in price for each day by using the formula:

$$\left(\frac{\text{Close} - \text{Open}}{\text{Open}}\right) * 100\% \tag{1}$$

C. Merging

Merging: For each article we wanted to append the price/volume information for bitcoin at the time of publication. We also wanted to append the search popularity for the word bitcoin at the time of publication. This involved merging on date between two json and a csv. The first step transformation involved loading the information into three separate pandas dataframes. The date and time fields were stored differently between each dataframe. We used a pandas function to_datetime to convert the dates to python datetime objects. This was not enough to begin merging, as some of the datetime fields included timestamps and some didnt. Using the .dt.Date method on the pandas series extracted only the UTC date for each of the fields. A pandas left merge on date was applied to the article dataframe and the bitcoin market information. A second left merge was applied to append the search trends results to each article publication date.

III. LOADING

We opted to use a non-relational database to host our data. Our data had already been merged into a single dataframe/json object, so we did not need to take advantage of the relational aspect of SQL. Furthermore we did not have a natural primary key field for our articles. The date of publication for our article is the cornerstone of each document, but it is not necessarily unique, as multiple articles on bitcoin could have been published on a single date. Since we did not have search data and market data for certain days, some documents did not have supporting details. Rather than dropping these articles entirely, mongodb allows us to still include these documents in these collections, giving us the flexibility to add in this data at a later point. MongoDB also gives us the flexibility to include more fields, which allows us to include other indicators of media participation, such as social media interaction. With mongodb, the loading process was facilitated by the fact that two thirds of our data were initially extracted as json objects.