Real-Time Cryptocurrency Analysis

Fatemeh Renani, Jaskaran Kaur Cheema, & Mohammad Mazraeh

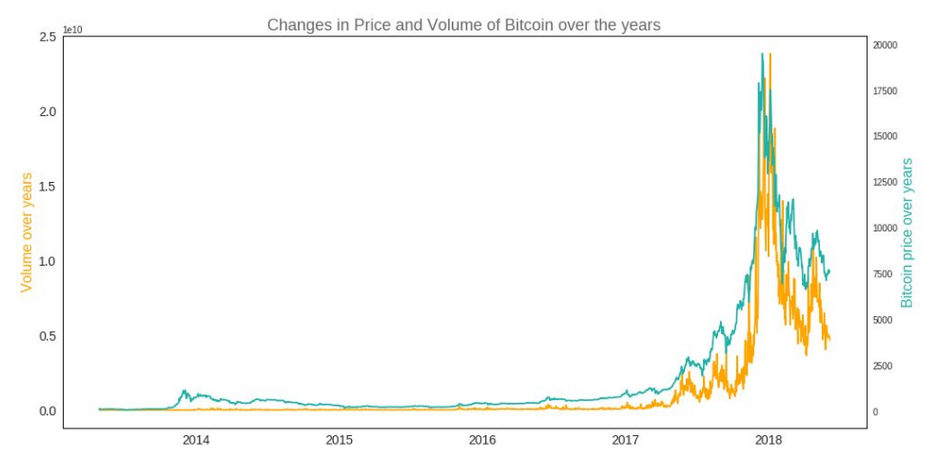
Simon Fraser University, Department of Computer Science

Introduction

Stock price forecasting is a popular and important topic in financial and academic studies and cryptocurrency market is not an exception. In this project we have created a platform for real-time cryptocurrency prediction. To achieve our goal we have combined the conventional time series analysis technique with news.

Why cryptocurrency?

Cryptocurrency market has shown exponential growth over the years especially bitcoin. Bitcoin has constantly been ranked as highest in terms of price and volume, therefore in our project we chose to focus on bitcoin.



Why streaming?

To make a better prediction we can use more data but often it causes long delays in the prediction and is being done in specific time intervals. Processing speed matters and important events around the world can immediately affect the price of cryptocurrencies and we need to be fast! So in this project we introduce a streaming platform in which different kind of data sources can be combined to make a real-time prediction.

Why price and news dataset?

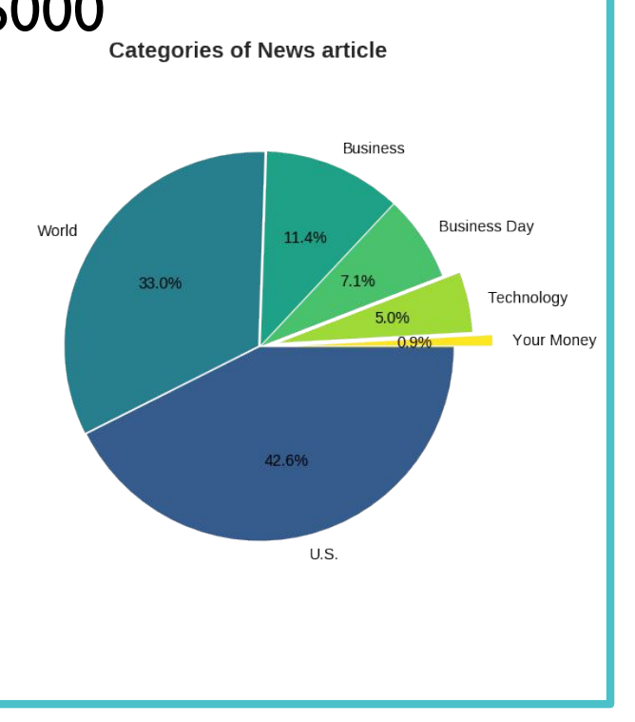
News and online information and social media can have an observable effect on investors' opinions. For instance, in April 2017 Bitcoin value rises over $1 billion as Japan, Russia move to legitimize cryptocurrency.

Data

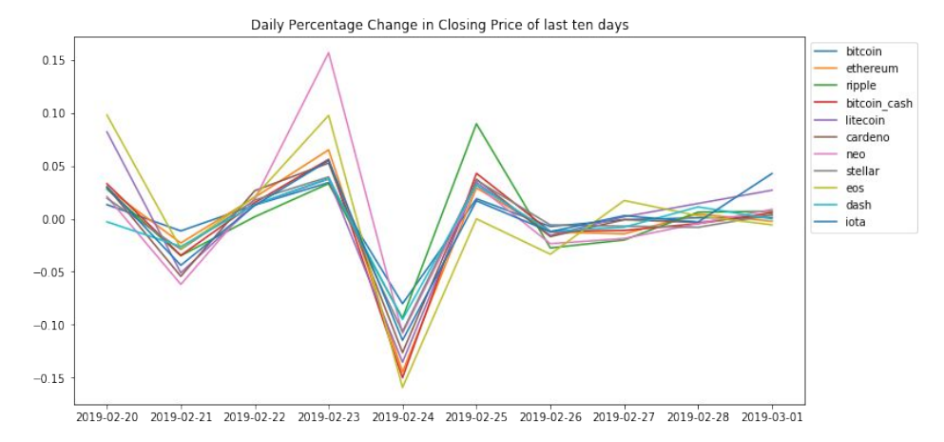
Bitcoin : Initially, we have obtained Bitcoin's daily price history by scrapping the CoinMarketCap website. However, due to our vision of streaming we decided to use the minute-by-minute price history which was available through a Kaggle competition.

News : The New York Times is the source of the news dataset. On an average we scrapped 6000 articles for each month. News articles relevant to the cryptocurrency are later filtered and analysis on the sentiment of news is performed.

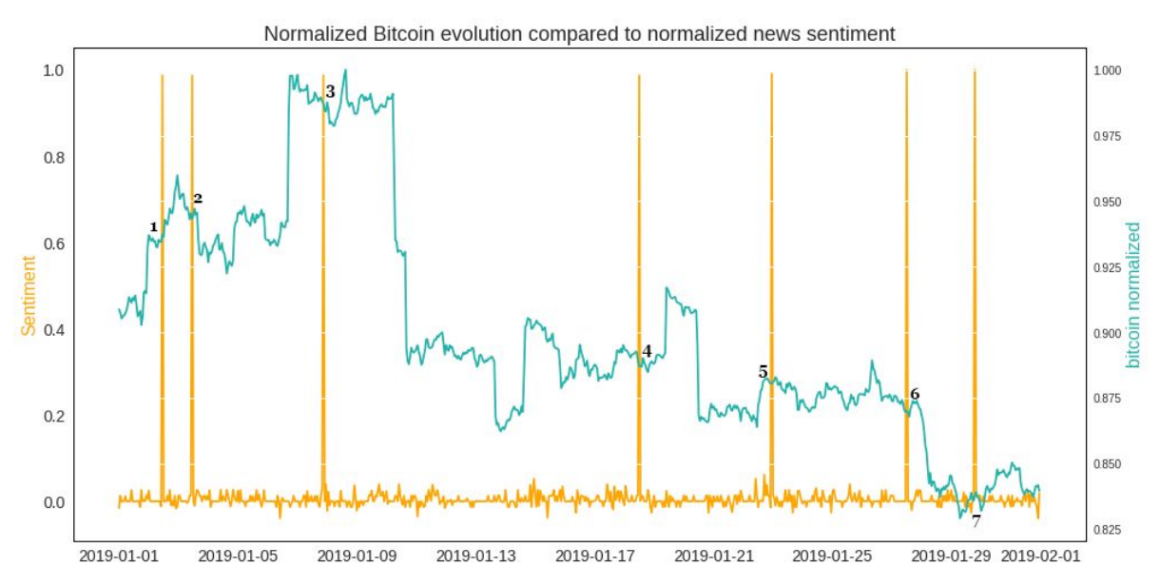
- While analyzing data, time of article publishing has also been considered in order to study its effect on the price of bitcoin.
- Data has been collected and cleaned from July 2018 - March 2019.Though, model has only been trained on Jan-March 2019 minute by minute data set.



Exploratory Data Analysis



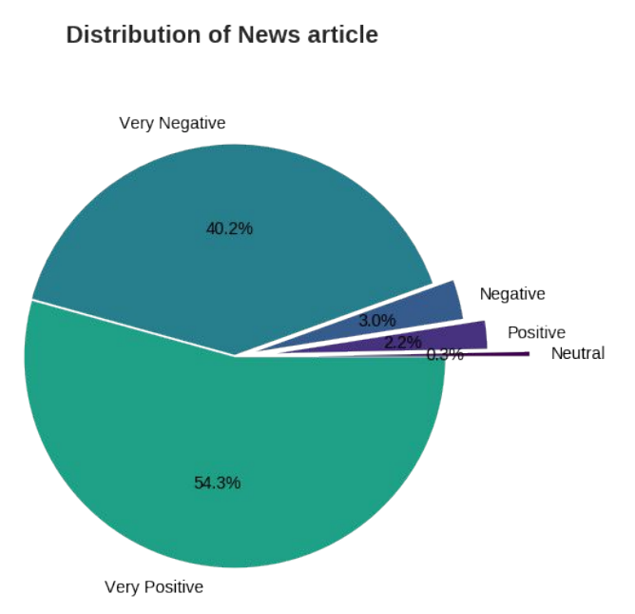
Analysis on price change of top 10 coins by volume shows high correlation.

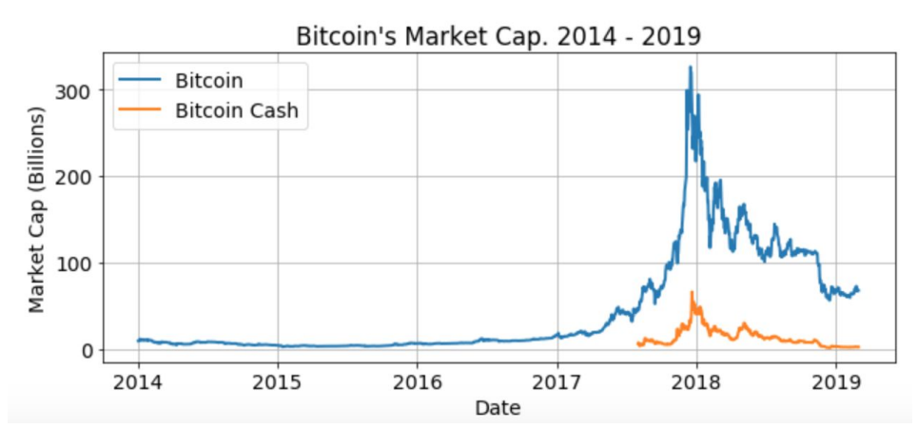


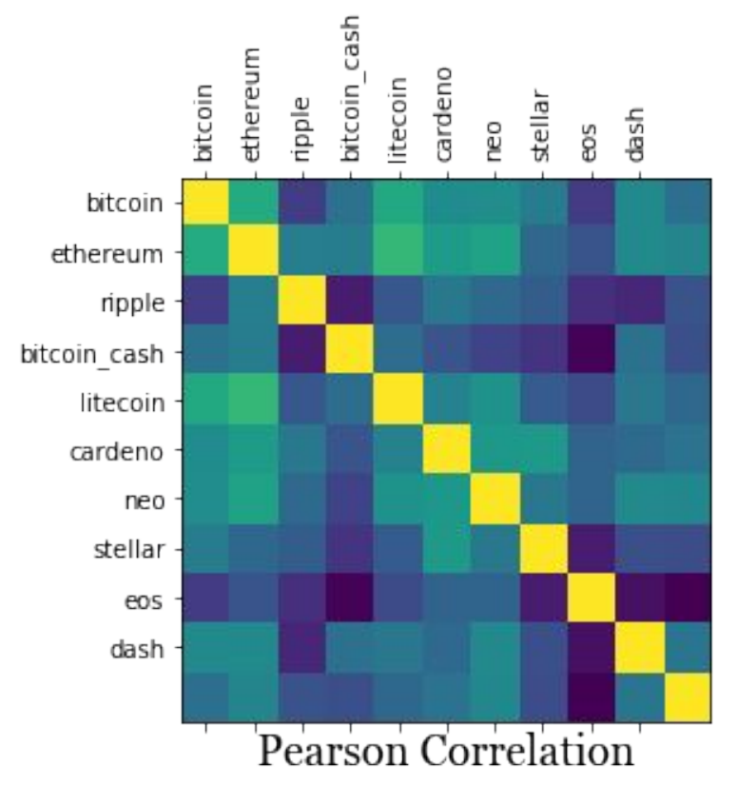
1,2,5,6: Bitcoin Price and sentiment of news are correlated.

3,7: Bitcoin price shows high fluctuation but sentiment of bitcoin as well as overall news is positive. Though at given date sentiment supports rising price

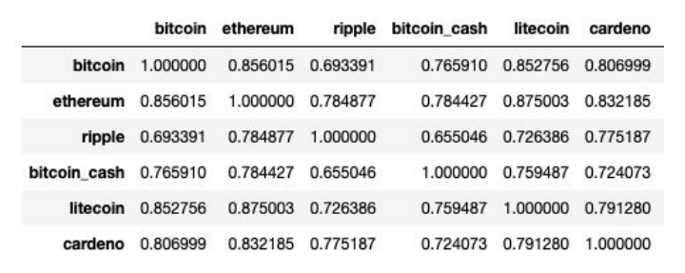
4: Interesting finding, sentiment of bitcoin news is positive but price is dwindling which is supported by overall sentiment of news which is negative.



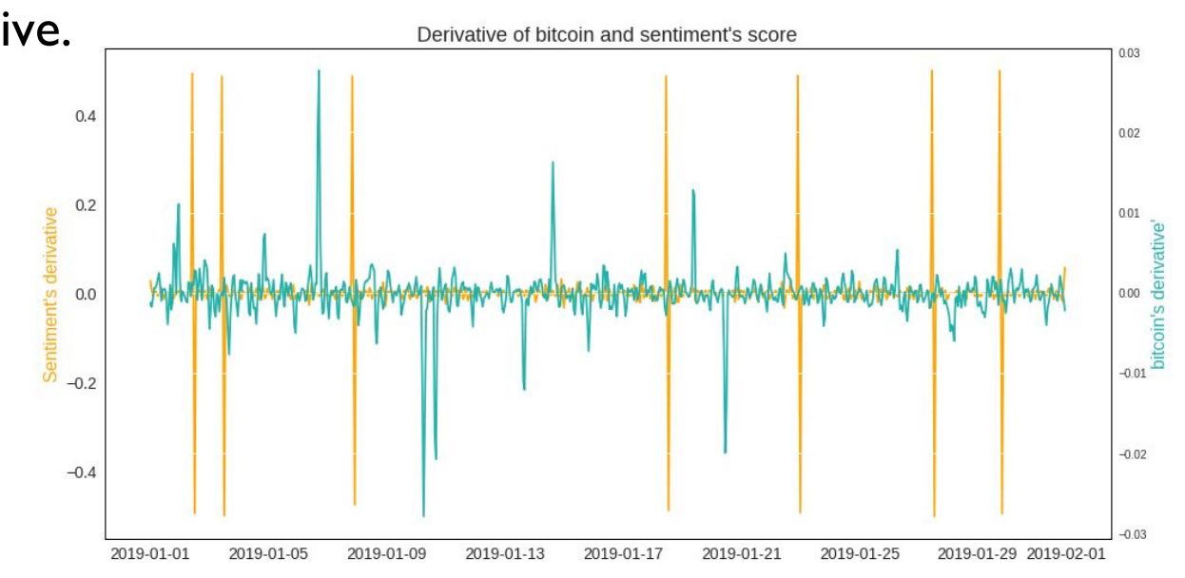




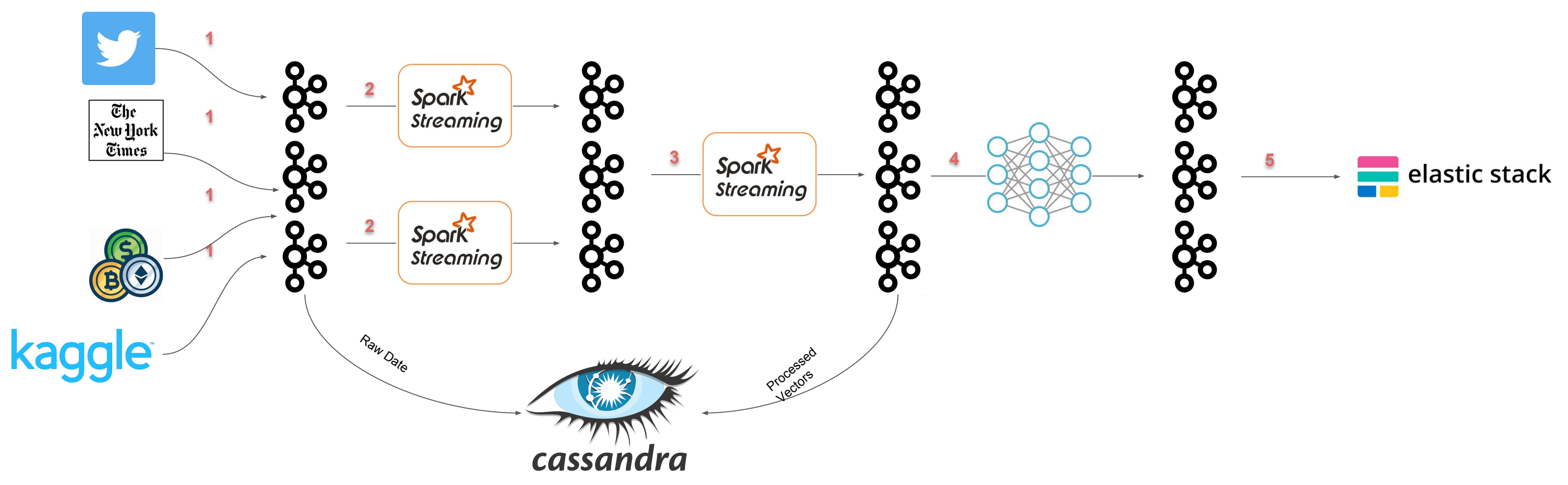
Entries in Correlation matrix had values greater than 0.7, which showed that all cryptocurrencies prices are highly related to each other.



	bitcoin	ethereum	ripple	bitcoin_cash	litecoin	cardeno
bitcoin	1.000000	0.856015	0.693391	0.765910	0.852756	0.806999
ethereum	0.856015	1.000000	0.784877	0.784427	0.875003	0.832185
ripple	0.693391	0.784877	1.000000	0.655046	0.726386	0.775187
bitcoin_cash	0.765910	0.784427	0.655046	1.000000	0.759487	0.724073
litecoin	0.852756	0.875003	0.726386	0.759487	1.000000	0.791280
cardeno	0.806999	0.832185	0.775187	0.724073	0.791280	1.000000



Architecture



1- Data Extraction: Data is collected from multiple sources of news and price data

2- Feature Extraction: Each event type (price/news) has its streaming application to clean it and extract features from it (e.g. sentiment score)

3- Feature Aggregation: Features from different data source types need to be aggregated into single feature vector which can be fed into the model

4- Prediction: Best model would be loaded into a streaming app,which will receive the feature vectors and make prediction.

5- Visualization: Any data visualization stack which can be connected to a queue can be used to visualize raw features/predictions/etc.

Challenges

Aggregation

One important question in real-time prediction is how frequently we want to prediction? Given a time period t (which can be minute, hour or day!), we want to handle data from different sources. We may have different price info from different exchanges and multiple news from different media in that specific time period. So we need an aggregation mechanism for each time period.

News Data

Consider in time period t we have articles $A_t = \{a_0, a_1, \ldots, a_{n-1}\}$. For each article we have a relevance feature which measures how much the article is related to our cryptocurrency and and a sentiment value which indicates how much positive/negative the opinion in the article is.
To aggregate the relevance and sentiment value of articles in a time period, we simply calculate a weighted sum of sentiments:

$$v_{news_t} = \sum_{i \in A_t} rel_i \times sent_i$$

Price Data

For price, we use the same Open, Close, Low, High features but again we need to introduce some aggregation functions to calculate these features for each time step. For Open and Close we simply average the values. For Low and High, we calculate the minimum/maximum price of different exchanges in that time period. For Volume feature we sum up all the Volume.

$$Open_{news_t} = \frac{1}{n} \sum_{i \in A_t} Open_i \qquad Low_{news_t} = min(Low_0,\ Low_1,\ \ldots,\ Low_{n-1})$$

$$Close_{news_t} = \frac{1}{n} \sum_{i \in A_t} Close_i \qquad High_{news_t} = max(High_0,\ High_1,\ \ldots,\ High_{n-1})$$

$$Volume_{news_t} = \sum_{i \in A_t} Volume_i$$

Sentiment Analysis

Gathering Data : Finding a data source which could provide historical news data was challenging. Due to change in structure of website over the course of time period, manual work of checking the structure and data received consumed time. Furthermore, limited access to article at some websites posed a problem while identifying the news data source.

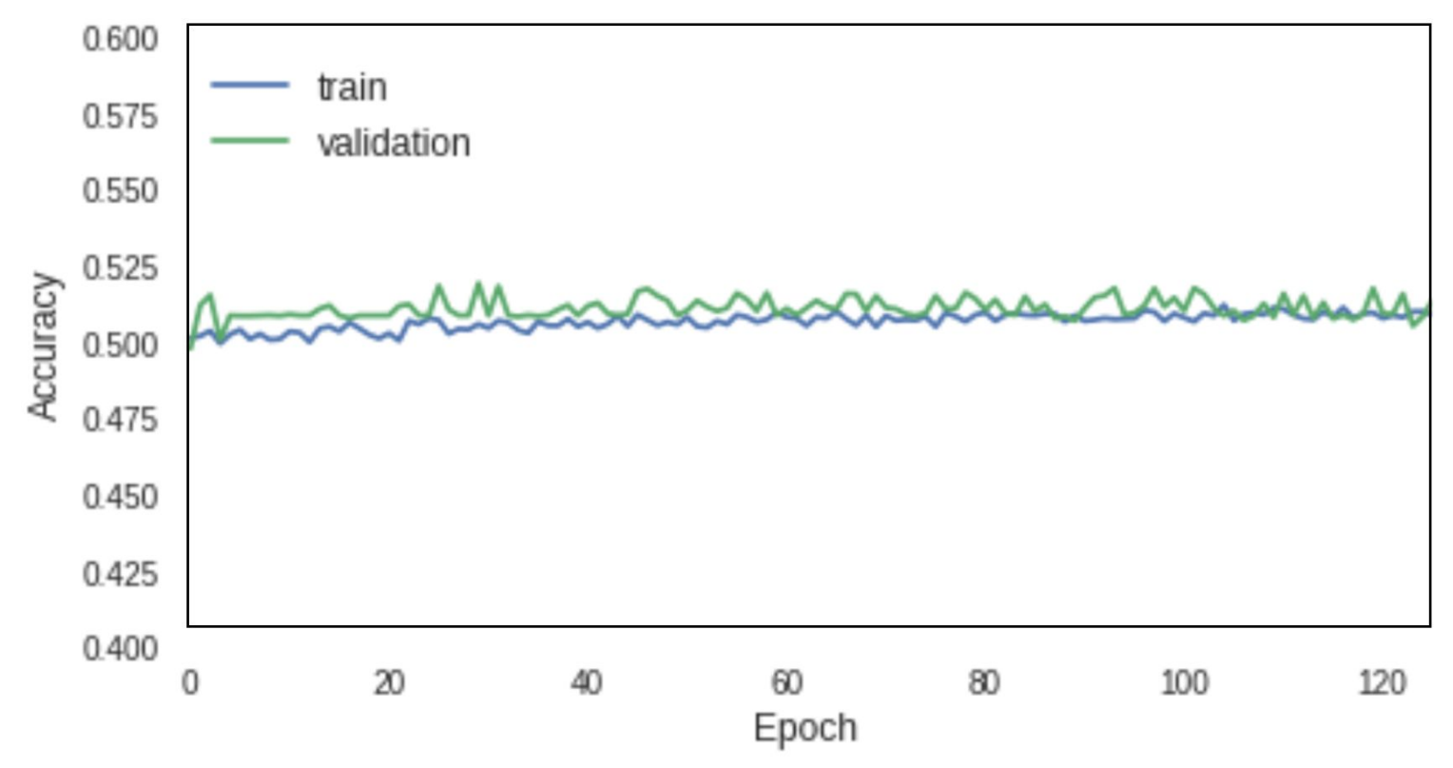
Cleaning Data: Scraped data was messy, contained HTML tags, numeric data and articles not relevant to the model. Such articles were filtered followed by cleaning. While preparing data for Sentiment analysis, extraneous words were identified . After this step, 6000 articles were reduced to around 1800 for each month.

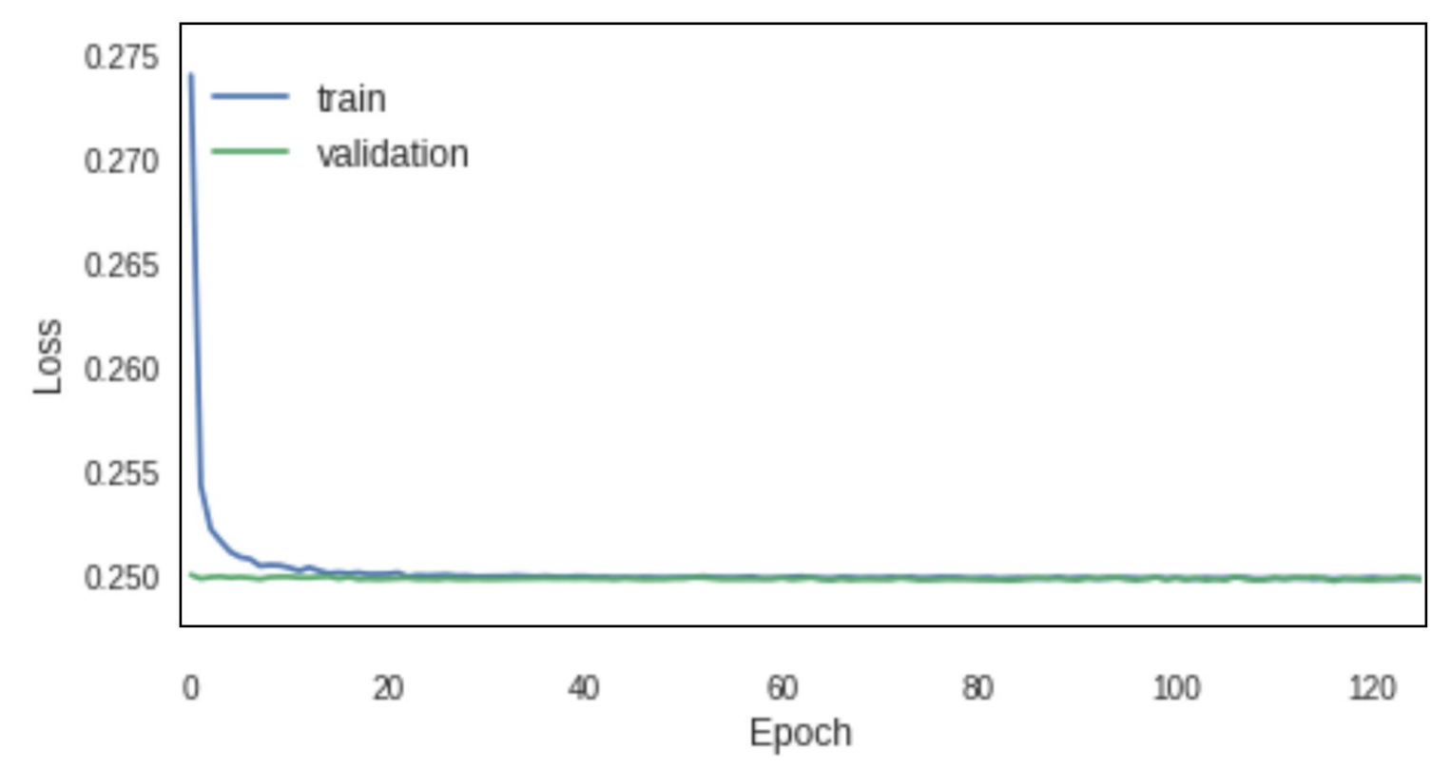
Model Implementation: Use of pre trained VADER model has been made to recognize the sentiments of news articles. Later, results of model are scored,with higher score to articles containing terms such as bitcoin or cryptocurrency.

Model Training

Due to recurrent nature of our data we have employed the recurrent neural network and long short-term memory models. The time-series data are generated by stacking 60 data points of aggregate price and news data. Given 60 minute (an hour) of price history and news for Bitcoin, our models will output a real value between 0 and 1. A value over 0.5 is a prediction that the price will rise, under 0.5, the price will fall.

Features: [Open , High , Low , Close , Volume , Sentiment]
Model: LSTM
Layers: LSTM/Dropout/Dense
Number of parameters: 25761
Size of train/validation/test set: 82533 / 8253 / 10000
Train/Validation accuracy ~ 51%





Discussion and Future Work

Achievements:

- We successfully created a general pipeline for a real-time stock price forecasting.
- New features can be easily calculated be replacing part 2 in the model
- New sources of each data can be easily added to the architecture.
- New data types can be easily integrated into the architecture. You just need to implement feature extractor for it.
- The model used for prediction can be trained offline and replaced with zero down time
- Every step in the architecture is scalable. It can be applied on thousands of events per second

Limitations:

- Not all deep learning models can be integrated with spark easily. We need to do part 5 in the model in a python streaming app
- Sometimes it can be hard to extract features and normalize data on a per event basis
- Our trained LSTM model has low accuracy which requires further parameter tuning

Future work:

- Connecting the pipeline to an exchange platform for real-time data entry
- Adding an scheduled model retraining in the pipeline
- Team-up with another group who have a better training model and improve the accuracy of our prediction
- Add real-time visualization of streaming data such as sentiment analysis and price graph

Resources

1. https://github.com/LinuxIsCool/733Project/tree/master/CryptViz
2. https://www.kaggle.com/mczielinski/bitcoin-historical-data
3. New York Times
4. Rose, B. (2018). Real-Time Crypto - a big data approach to analyzing and automating cryptocurrency trading. Lean Pub.
5. https://www.nltk.org/api/nltk.sentiment.html
6. Machine Learning Algorithms by Giuseppe Bonaccorso

Github : https://github.com/MohammadMazraeh/realtime-crypto-analysis