ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Bitcoin pricing prediction and trading simulation through time series and sentiment analysis

Big Data project, spring 2014

Jonathan CHESEAUX (jonathan.cheseaux@epfl.ch)
Fabien SCHMITT (fabien.schmitt@epfl.ch)
Marzell CAMENZIND (marzell.camenzind@epfl.ch)
Igor VOKATCH-BOLDYREV (igor.vokatch-boldyrev@epfl.ch)
Ilia KEBETS (ilia.kebets@epfl.ch)

Supervisor : Aleksandar Vitorovic

Contents

1 Introduction

Bitcoin is a peer-to-peer payment system introduced in 2009 [1]. It is now commonly called a cryptocurrency. It is not controlled by any single entity, so it is completely decentralized.
One way to get Bitcoins is to mine them, but a dedicated mid-range computer could take upwards of a year to decrypt a single block. Bitcoins can be exchanged for other currencies and products. Participants in online exchanges offer bitcoin buy and sell bids. The volume of Bitcoin transactions has significantly increased in the last few months, bringing a lot of interest around this crypto-currency not only in specialized circles but also in the mass media. Currently, the acceptance of bitcoins by merchants is relatively small, but growing. Its usage by speculators, comparatively to the commercial use, is high, and this brought a high volatility to the market. Bitcoin's volatility is unique among commodities, it has been estimated as being 7 to 8 times higher than gold[1]. People tend to agree that it is due to insufficient liquidity and uncertainty of its long-term value.
The main difference between a Bitcoin market and current stock exchanges is, as mentionned before, its high volatility. The other difference is that transactions are not instantaneous in the Bitcoin protocol. The minimum amount of transferred Bitcoins is 0.00000001 BTC[2].

For this project, we have built a framework capable of predicting the evolution of Bitcoin market and simulating different trading strategies. We use sentiment analysis (Section 2) and time-series models (Section 3) for price prediction. A web front end (Section 5) plots in real-time the current value of the cryptocurrency with the time-series models, shows the prediction on the value of a bitcoin and the last tweets on the subject (this permits us to improve the prediction by letting each individual user to classify them).

[1] according to Mark T. Williams of Boston University - "Beware of bitcoin" wbur.org
[2] equivalent to around 0.0000044 USD as of 19^{th} May 2014

2 Sentiment Analysis

The main question we want to answer by using sentiment analysis on price prediction, is if it's a valid method of evaluating/predicting the evolution of bitcoin value. There is no doubt a correlation between the mood of the news and the prices, but we would like to show that it is not only the market driving the news (as an example, a drop in price would generate negative coverage in the medias afterwards).

We used three classes, namely `positive`, `negative` and `neutral` to classify tweets' polarity and average the global mood for each day. The evolution of the mood is then used to influence market transactions.

2.1 Sentiment in media

In order to acquire data we first wanted to crawl news from the GoogleNews app, but since the provided API is deprecated and their Terms of Services (ToS) forbid automatic crawling of search results, we soon decided to use another media.
Another solution that we investigated was to use famous newspaper API (like The Guardian or The New York Times). We have implemented crawlers for these two websites but we couldn't get more than one story about bitcoin per day, which is not good enough for analyzing the market's mood.
This is why we decided to focus on Twitter, which is an online microblogging social media that allows people to publish messages limited to 140 characters.

2.2 Twitter API

Twitter provides several API for different programming languages. For Scala, we used Twitter4J which is really simple of use. After having registered a free developer account on Twitter website, we were provided with OAuth tokens which serves as API keys.
We can then listen for upcoming tweets containing the desired hashtags.[3]

2.3 Naive Bayes Classifier implementation

After having tested and rated several API for sentiment analysis we chose to use the NLTK framework provided for Python. This is a powerful library that allows to use multiples classifiers and models, and was a particularly good fit for our project in term of performance and speed.

The model used is the well-known Naive Bayes Classifier and it assumes strong independence between words of sentences (tweets), which is not true in general but past experiments have shown that this model is a good fit for opinion mining.

[3]for this project, we used `#bitcoin`, `#bitcoins`, `#btc` and `#cryptocurrency`

2.3.1 Corpus

In order to train our model we needed a significantly big dataset. Our first idea was to use a movie-review corpus that had the advantage of containing a significant amount of labelled texts. However, while training the model we noticed that the accuracy was really weak. We then tried to find tweet corpus online but there aren't many and none specifically designed for bitcoin, this is why we decided to manually classify a large number of tweets[4]. In parallel, we decided to use an existing corpus[5] of labelled tweets focused on `Apple`, `Microsoft` and `Google` to increase the size of our learning set. We noticed that the accuracy was improved even if the tweets weren't talking about Bitcoin in particular.

2.3.2 Train the model

A lot of preprocessing is needed to `clean` the tweets and build a proper learning set. Punctuation, links, smileys and numbers are removed, as well as short words and stop words[6]. The most frequent words will then be used for the feature selection phase of the learning process. This part is automatically done by the `NLTK` library.

2.3.3 K-Fold Cross Validation

Overfitting is a big concern while dealing with small learning sets, this is why we chose to apply a 5-fold cross validation on the training set. Over the 5 validation computations, the accuracy was on average 70% and uniform.

2.3.4 Results

In order to compute the relevance of sentiment analysis on bitcoin prices we needed to compare the mood with the actual price evolution of bitcoin. To this end, we have downloaded a huge archive containing all the tweets that were written in December 2013, from the website `Archive.org`[7]. We then filtered them to keep only tweets talking about bitcoin and applied our sentiment anaylsis tool on it, summing the scores for each day. In parallel, we gathered bitcoin's price evolution during the same month. Figure 5 shows the correlation between the price evolution and the sentiment of the tweets, and it is clearly noticeable that the mood qualitatively follows the same trend that the bitcoin prices.

[4]We have classified 3000 tweets by hand
[5]https://github.com/zfz/twitter_corpus/blob/master/full-corpus.csv
[6]stopwords are irrelevant words like 'the', 'it', etc.
[7]https://archive.org/details/twitterstream

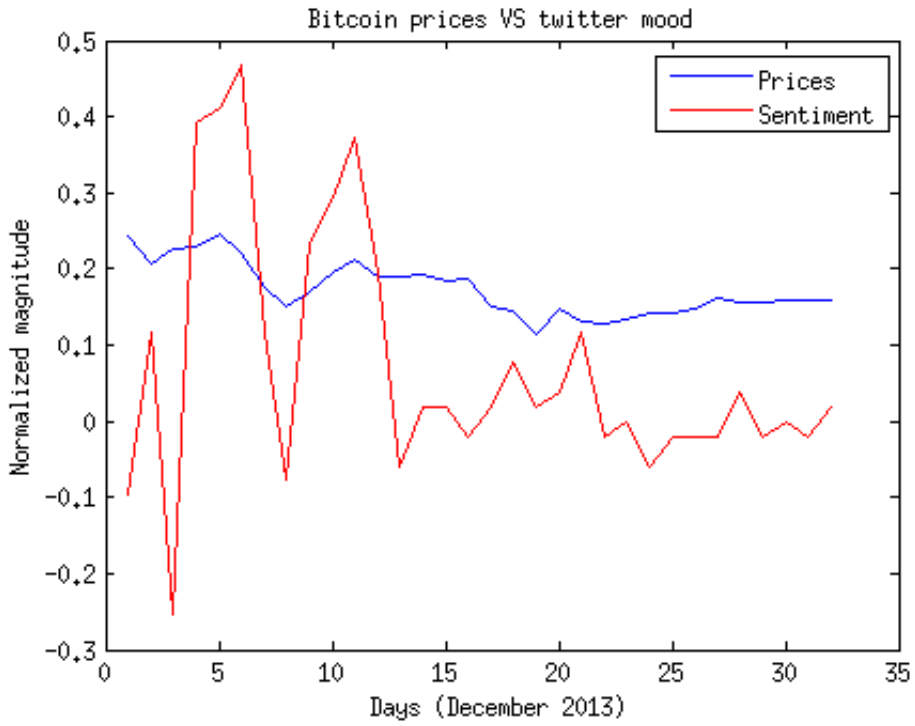

Figure 1: Correlation between Bitcoin price evolution and global market's mood

2.3.5 Crowd-sourcing Supervized Learning

Since manually labelling tweets can be a really hard task, we chose to let people improve the training set by correcting in the web interface (see Section 5.4). When the user thinks a tweet has been missclassified, he can change its label. When an arbitrary number of tweets have been corrected, the sentiment analysis model is trained again using this data. This ensures that the accuracy of sentiment analysis will improve with time.

3 Time-series

3.1 Moving Averages

The moving average (MA) is a tool widely used in financial analysis, its simplicity makes it accessible to almost everyone. Even though there are a lot of different MA types, their definition remain the same : a weighted sum of prices, that compared to the security price gives an indication of the state of the market. The main difference between all of them is the weight attributed to each element of the series, based on their recency [2] [3].

3.1.1 Simple Moving Average

The simple moving average (SMA) also called arithmetic moving average is the sum of all the prices, where each closing price has the same weight. The only parameter is the period over which the mean price is calculated. The advantage of this method, apart from being simple, is that it gives good notions of the general mood and is therefore more resistant to abrupt changes [2]. On the other hand, it will need more time to adapt to the actual fluctuations of the market. It can be computed using the following formula:

$$SMA = \frac{\sum_{k=1}^{n} p_k}{N} = \sum_{k=1}^{N} \frac{p_k}{n}$$

where N is the period and p_k represents the price at time k

3.1.2 Exponentially weighted Moving Average

The exponentially weighted moving average (or exponential moving average) (EMA) is a more complex form of MA. It will give more weight to the most recent price and therefore corresponds more accurately to the actual mood of the market, but will also be more affected by abrupt changes of the price, and therefore may induce wrong sell or buy signals. It can be computed using the following formula :

$$EMA_k = \alpha(p_k - EMA_{k-1}) + EMA_{k-1}$$

Here, we can see that EMA_k represents the EMA at times k. The stopping condition of this recursive computation happens at EMA_1 and can be either the price at time $k = 1$ or a simple moving average based on previous events. Also notice that in this scenario, the period is used to compute α, which is the exponential parameter, as follows : $\alpha = 1/(2N + 1)$ where N is the period [2].

3.2 Trading Strategies

The fact that moving averages are computed over the past prices will induce a lag in their result. Even though, they are still used to find the direction of the trend and create buy or sell signals. Those signals are produced when the moving average suddenly drops or rise. Techniques exist to avoid getting false signals when a moving average is oscillating around a given price. For this project we focused on two of those techniques, the envelope and the double crossover, which are described below. These two methods can be used with both EMA and SMA.

3.2.1 Double Crossover Method

The double crossover method involves two different moving average, one short-term moving average and a long term moving average. A signal will occur when the two MA's cross each other. Typically a sell signal will happen if the shorter MA crosses from above. And, on the hand, a buy signal occurs when the shorter MA crosses from below. This method is really good in case of strong tendencies, but can produce a lot of false detection without them. We also need to note that this method will need more time to find a good signal as it uses two lagging MA [6].

3.2.2 Moving Average Envelope

The main goal of the moving average envelope is to avoid getting trend signals during weak tendencies. This methods uses two envelopes based on the MA and a percentage, one above and one below. They can be computed in the following way :

$$Env_{below,k} = MA_k(1 - \alpha)$$

$$Env_{above,k} = MA_k(1 + \alpha)$$

where MA_k is the value of the MA at time k and α is the percent used. Depending on the market, α may take different values. For this project, we used a value of 2.5%. Using these envelopes, a sell signal will occur if the price goes over the above envelope and a buy signal will occur if the the price goes under the below envelope [5] [4].

3.3 Computation of the Gain

For the computation of the gain, we had to set up some parameters for our wallet. As this project is experimental, we decided that we have already in our possession 100 bitcoins, and 100'000 USD. We also set up the maximum amount of bitcoin traded per signal to 0.2.
Our gain computation has to take into account both the sentiment analysis and the signals sent by the time series analysis. Every time a new moving average is computed, the program will check if there is a buy or sell signal. If there isn't it will compute the gain based on the money we have made or lost and the bitcoins that we have bought or sold [8]. On the other side if there is a signal, the program will check the sentiment analysis. If the sentiment analysis and the time series have the same view of the market tendency, the investment is computed as a function of the moving average. If they are not, the investment will still be made, but will be divided by a factor of 4. The investment is computed in the following way :

$$Inv_{btc} = (1 - \frac{min_{MA}}{100 max_{MA}}) max_{btc}$$

[8] This computation is based on the actual price of the bitcoin

4 Architecture

4.1 Bitcoin Crawlers

The Back-end runs a crawler for the BTC-e[9] Bitcoin exchange platform. It fetches the last 2000 transactions when started to populate the price graph and compute the moving average curves.

The internet's bitcoin platforms provide publicly accessible APIs allowing the user to retrieve the latest executed transactions, the market depth, informations about the platform and the current fees. There are also private APIs that are bound to a user, these can buy, sell and place options. RESTful APIs make around 80% of the market share. The data formats are 80% Json, 15% XML and 5% CSV.

Our implementation currently supports three platforms, namely BTC-e, Bitstamp and Bitfinex, each providing exchange data for several cryptocurrency pairs. By default, we only display BTC-e on the GUI.

The transaction crawlers are managed by a centralized object, called DataSource, which starts, stops them and manages a local in-memory cache. DataSource is also the drop-in center for external actors requesting data. It provides data in form of transactions sorted by exchange and currency, precomputed OHLC (Open-High-Low-Close) values for configurable periods and counts.

When the crawler receives new data from the exchanges, DataSource updates its cache and pushes the updates to its observers, such as trading algorithms and user interface.

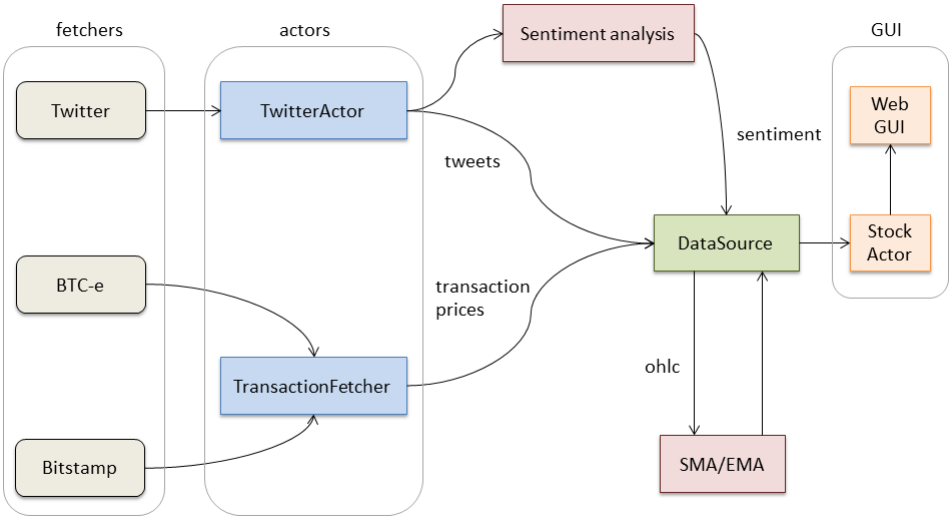

Figure 2: System Architecture

[9]https://btc-e.com/

4.2 Communication

The front- and back-ends communicate via HTTP, the front- and back-end actors are StockActor and DataSource respectively. The front-end Actor sends registration messages to the back-end Actor when the page is loaded. When they are received, the sender is registered in an observers list. From hereon, each message is pushed to the front-end when they are acquired. The different messages are the following: Tweets, Transactions, EMA and SMA points.

5 Web front-end

5.1 Play Framework

We decided to separate the project into front- and back-end communicating with HTTP. For the front-end, we used Play, an open source web application framework.[10]

It implements the Model View Controller design pattern to develop reactive applications in Java or Scala. The communication is developped using Scala Actor classes from the akka library[11], which abstracts the implementation of number of threads and concurrency.

To allow a more reactive testing, we used Typesafe reactive platform[12] during the development of the web application. It provides a browser-based environment with hot-reload which allows to compile and run the new source code without restarting the server.

The source files are built using SBT, with Typesafe it is done seamlessly. It also allows us to monitor the amount of used ressources, mainly the number of instanciated Actors.

The web interface is coded in HTML, Less and coffeescript.

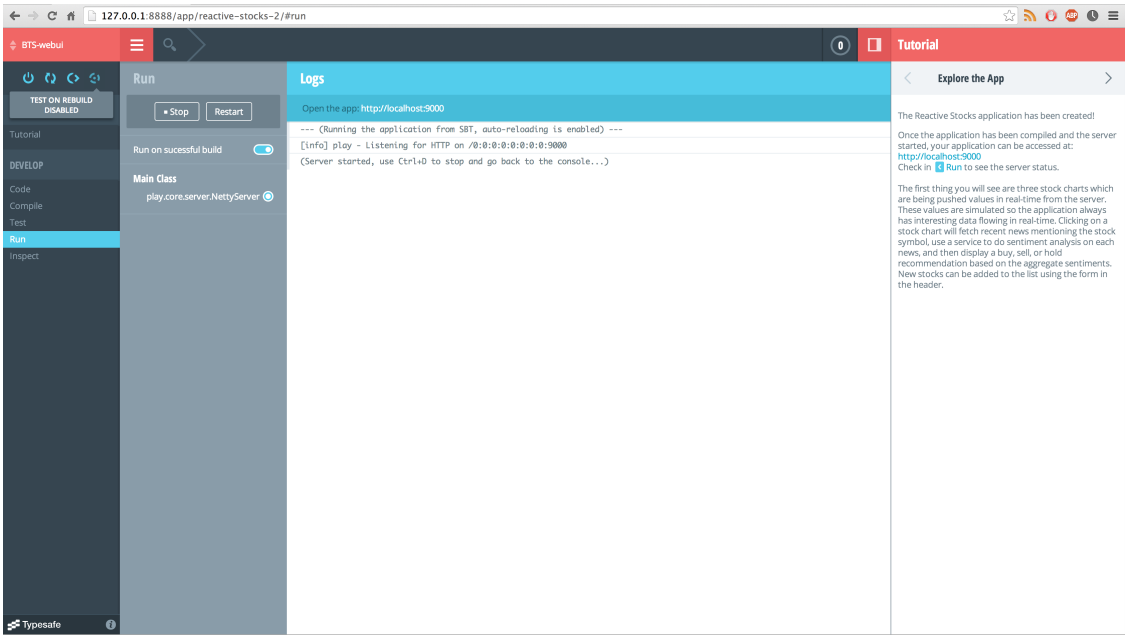

Figure 3: Typesafe Activator

[10] http://www.playframework.com
[11] http://akka.io
[12] http://typesafe.com/activator

5.2 Stock exchange graph

The graph displays 3 curves: the current price of the Bitcoin in yellow, the Simple Moving Average in red and the Exponential Moving Average in blue, these values are in USD. The displayed Bitcoin price is actually the price at each transaction, for this reason, the points are not at regular time intervals. The plot is refreshed every second.
It is possible to zoom in and out of the graph by selecting the amount of points that are displayed from the most recent transaction.
The graph has been implemented using the Flot Plot library.

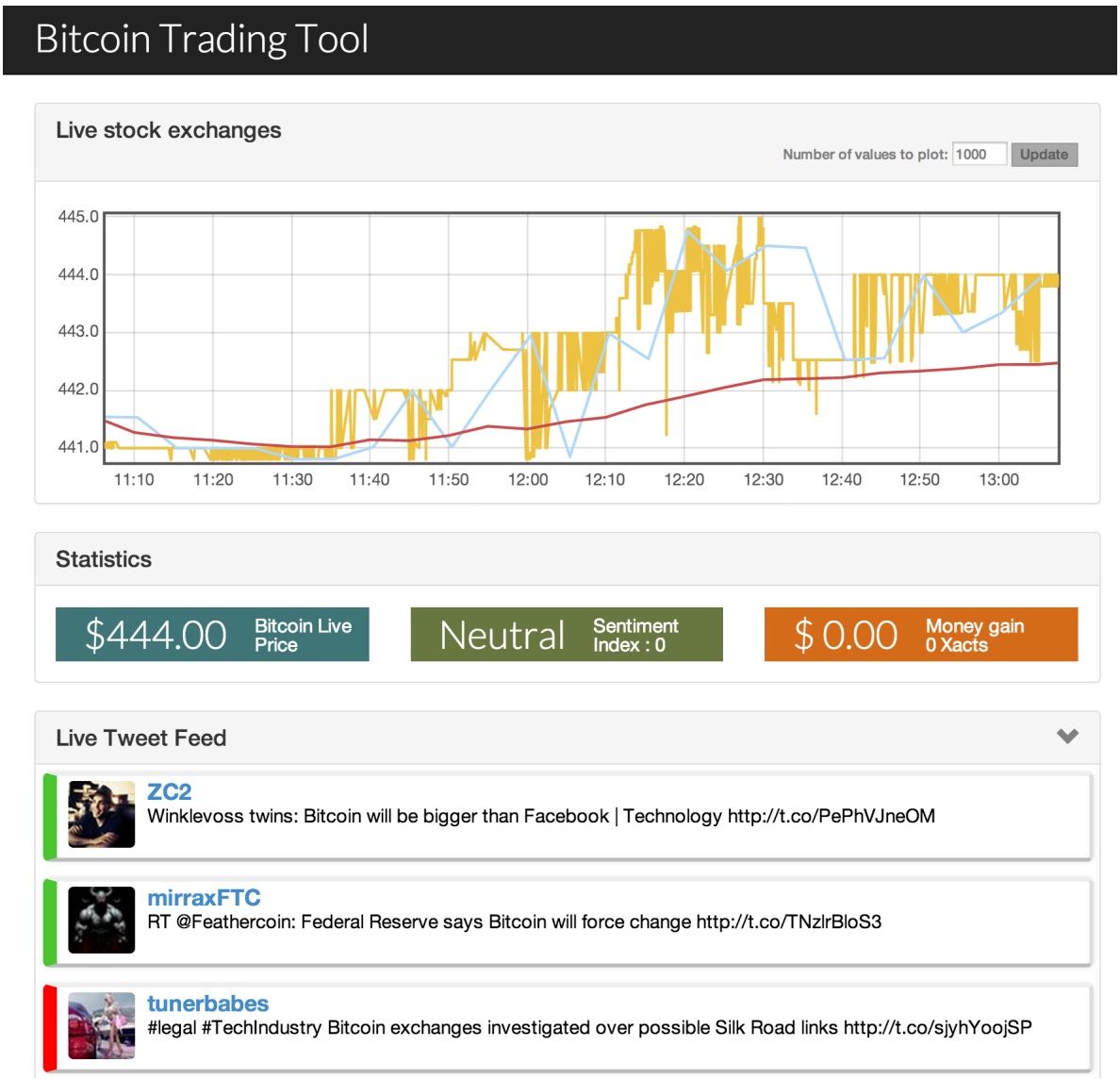

Figure 4: Web App Graphical Interface

5.3 Statistics

This bar shows the current price of the Bitcoin, which is the price at which the most recent transaction was executed. The current sentiment is displayed according to the

classified Tweets of the last 24 hours. The sentiment index is the sum of the classified Tweets, +1 for positive and −1 for negative. The mood of the market is "Positive", "Neutral" or "Negative" if the index is respectively superior to 10, between −10 and +10 and inferior to −10. We also display the money gain according to our predictions using moving average algorithms improved with the sentiment analysis. It takes into account the fees of transactions (0.2%).

5.4 Live Tweets

The Live Tweet Feed displays the most recent Tweets relevant to Bitcoin. The sentiment is illustrated by a red (or green) border at the left of the tweet to represent a negative (or positive) tweet. Neutral tweets are hidden because it concerns the vast majority of them (due to bots, spam, ads, etc.). The user can correct the sentiment of a tweet simply by clicking on it and by choosing the correct label in the pop-up that appears on the screen. When a user corrects a tweet, the tweet content along with the correct sentiment are sent to a remote PHP script that append a new line in the training set. When this file becomes big enough, we can then retrain the sentiment analysis tool and enhance the accuracy. The user can also block spams by clicking on the `Spam!` button. Spams are produced by fake accounts handled by bots that continually post ads on the social media. The use of PHP here is critical since it is not possible to write/read files in JavaScript/JQuery.

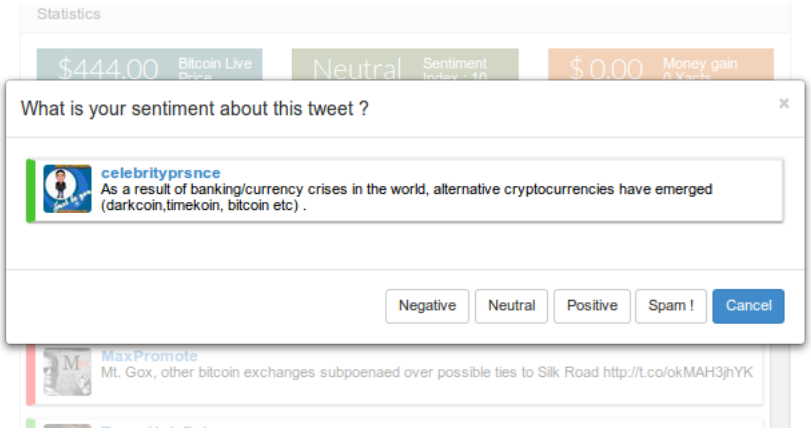

Figure 5: Tweet Index Correction

6 Conclusion

Working with bitcoins has proven challenging. The methods used for sentiment analysis at the beginning were quickly found out to be inacurrate. It was a case of trial and error to find the best classification method. For the time-series models, having the right parameters has proven itself crucial to produce satisfactory results. Another difficulty arose when it was time to link all the modules of the project, to link all the results and output them on a usable visual interface.
A detail we wanted to investigate was the correlation between price trend and sentiment in social media. One important factor is to find if the sentiment really influences the price or if it was the other way around. As it turns out, it can go both ways. Some positive mood often indicates a future rise in prices, as well as a sudden drop in the price can later generate an important quantity of negative tweets.

References

[1] Nakamoto, S., Bitcoin: (2008) A Peer-to-Peer Electronic Cash System

[2] Droke, C. (2001). Moving averages simplified. Marketplace Books.

[3] Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. The Journal of Finance, 47(5), 1731-1764.

[4] Pring, M. J. (1997). Martin Pring's Introduction to Technical Analysis. McGraw Hill.

[5] Leung, J. M. J., & Chong, T. T. L. (2003). An empirical comparison of moving average envelopes and Bollinger Bands. Applied Economics Letters, 10(6), 339-341.

[6] Carr, T. K. (2007). Trend Trading for a Living: Learn the Skills and Gain the Confidence to Trade for a Living. McGraw Hill Professional.

Github Repository: github.com/cheseaux/BitcoinTradingSystem