



American University of Armenia
Հայաստանի Ամերիկյան Համալսարան
MORE THAN AN EDUCATION - A COMMITMENT

Capstone Thesis

**Sentiment Analysis To Predict Global Cryptocurrency
Trends**

By: Gasia Atashian,
Hrachya Khachatryan

Supervised by Arsen Mamikonyan

Submitted to the College of Science and Engineering

May 2018

Abstract

This project involves discovering relationships between forum talks and bitcoin price changes. We have shown that there is a big correlation between bitcoin price and bitcointalk.org forum posts. For that purpose we took the data from the Bitcoin Discussion of bitcointalk.org, performed sentiment analysis as well as relevant statistical analysis, topic modeling and then constructed neural network to predict bitcoin price values.

Contents

Abstract	3
Introduction	5
2. Related research	6
3. Data Gathering and Preprocessing:	7
3.1 Data Collection:	7
3.2 BitcoinTalk forums:	7
3.3 Historical Bitcoin Price data	8
4. Topic Models: LDA	10
4.1 Data Preparation and Transformation	10
4.2 Constructing a document-term matrix	11
4.3 Applying the LDA model	11
4.4 Results	11
5. Sentiment analysis	13
5.1 Sentiment analysis of forum posts	13
5.2 Bitcoin price change Peaks	13
5.3 Relation of sentiment analysis to price changes	14
6. Neural Network	15
6.1 Model Architecture	15
6.2 Model validation.	16
6.3 Feature Summary:	16
6.4 Model results:	18
8. Model Evaluation and Summary	23
9. Code Sources	23
References:	24

1. Introduction

Bitcoin is a cryptocurrency that is used to make online payments and has become very popular nowadays. Moreover this cryptocurrency is not only used for online payments, but also for investments. Hence it became very important nowadays to understand how the bitcoin price changes and what kind of factors influence its fluctuation. The rise of cryptocurrencies has changed the way of economic transactions greatly. Besides the bitcoin several other cryptocurrencies have come into existence as well. The increase of the bitcoin circulation also brought a lot of users into social media and online forums to share their opinions and reactions about this cryptocurrency. People who have common interests tend to make posts about certain topics. Moreover, as bitcoin is mostly traded on the Internet, they make their decisions about buying or selling the bitcoins mainly based on the information that they obtain from the same Internet, usually from online forums and social media. And the relationship of such forum discussions and bitcoin price fluctuations is not well studied today. Therefore this paper aims to find such relationships by making sentiment and topic analysis of Bitcoin discussion posts from bitcointalk.org and construct a predictive model.

In order to build our model we have downloaded data from bitcointalk, cleaned the data and used machine learning to figure out correlation between blog posts and bitcoin price. We have scraped the data from the mentioned source ([Bitcoin discussion](#)), then cleaned that to keep only the useful information by removing unmeaningful words as well as HTML tags, stemmed the words, (more on this in section 4.1). And after that we constructed a topic model with Latent Dirichlet allocation (LDA) [1], which is described in section 2. Then we made a sentiment analysis of our data. Afterward by having that information we analyzed which kind of impact does that have on bitcoin price fluctuations by building a neural network with multiple layers (section 6).

2. Related research

Some people made forum posts sentiment analysis, without considering the information derived from cumulative user posts' data gathered during a specified period [9], while others made research on online user comments.

For that purpose, topic modeling has been intensively studied as a technique for analysing user opinions and thoughts from their textual posts[10]. Topic modeling[11] is a text-mining method that gives a collection of prevailing topics and related keywords out of a large-scale document corpus. The topics give users an instant overview of the overall corpus, by eliminating the need to read through all the posts, which would be very difficult and time-consuming process.

Lately, collaborative filtering and topic modeling were integrated for generating scientific article suggestion systems on online community[12]. A Temporal Latent Dirichlet Allocation (TM-LDA) system was used to make a deep analysis of the online social media by employing an advanced Latent Dirichlet Allocation (LDA) topic modeling[13]. Also, application of the LDA approach to Chinese social reviews discovered the sentiments underlying some social events and services[14].

3. Data Gathering and Preprocessing:

3.1 Data Collection:

Data is a gathered information that comes from real observations. Data collection plays a vital role in each project, as the effectiveness and the cleanliness of the data directly affects the results. For this project, the data is mainly gathered from two resources; from the bitcointalk forums and from the historical bitcoin price exchange data.

3.2 BitcoinTalk forums:

Analyzing the different forums of social media, we mostly found bitcoin related data on Reddit, Twitter and BitconTalk. We tried to gather data from Reddit and do analysis on, however Bitcointalk forum was more related to our project as Reddit contained more generalized posts. Although there are many projects that did sentiment analysis on Twitter data, but we decided to use BitcoinTalk data to add more value to the overall research and project.

We gathered data from <https://bitcointalk.org/> forum, which consists of topics and each topic contains replies. Using the library Scrapy in python [4], we scraped the HTML data of each page from bitcointalk forums since April 23, 2011 6:24:16 PM until May 05, 2018 01:15:00 AM, which took almost 5 hours to complete. Then we separated each topic with its replies. We preprocessed the data by cleaning the message part and separating the quoteheader for the message. Initially, messages contained HTML noise, like ''' which we replaced it by its human readable form which is '.

We used `htmlpackage` in python, specifically the `unescape` function to clean the HTML noise and also the following mappings to further clean the rest:

```
map_clean= {'&#039;': '',
            '&quot;': '',
            '&nbsp;': ' ',
            '<br />': '\n',
            '<br/>': '\n',
            '<br >': '\n',
            '<b>': '\n',
            '&lt;': '<',
            '&gt;': '>',
            '&le;': '<=',
            '&ge;': '>='}
```

Our final data frame consists of the following columns:

Timestamp, topic_id, topic_title, message_number, message_author, message_text, quoteheader.

Timestamp is an integer in Unix time

Topic_id is the id of each topic as the Bitcointalk keep tracks of unique id topic

Topic_title is the title of the topic

Message_number is the queue where the message (reply) stands in the following topic

Message_author is the author of the message (reply)

Message_text is the extracted message (reply)

Quoteheader refers to the quoteheader if there is any.

In total we have 1046382 rows, with 25047 unique topic_title (topic_id)

	timestamp	topic_id	topic_title	message_number	message_author	message_text	quoteheader
0	1524542495	3381878	Prices will some day fluctuate less?	1	Johannson	Do you think that someday the prices for Bitco...	NaN
1	1524545968	3381878	Prices will some day fluctuate less?	2	no0dlepunk	Yup, once bitcoin becomes accessible to everyo...	Quote from: Johannson on April 24, 2018, 04:01...
2	1524547244	3381878	Prices will some day fluctuate less?	3	scorpionso	I think so , if btc going to substitute with e...	NaN
3	1524570746	3381878	Prices will some day fluctuate less?	4	hase0278	It can actually, so long as bitcoin gets evenl...	Quote from: Johannson on April 24, 2018, 04:01...
4	1524571076	3381878	Prices will some day fluctuate less?	5	horrifiedx1	to be a currency or a means of payment i think...	NaN
5	1524571184	3381878	Prices will some day fluctuate less?	6	Naman1111	Yes, Right now the market is not as matured as...	NaN

Figure 3.1. Bitcointalk forums data.

Having the data ready, we can proceed to the next step.

3.3 Historical Bitcoin Price data

We have used the GDAX python library [5] to gather bitcoin history data. The API gives 200 points on each request, we had to gather the data part by part and then merge them.

We collected the data from 2014 until 2018 in an interval of 5 minutes.

The data contained six features, timestamp, low, high, open, close and volume.

Timestamp refers to the time in Unix time which we converted it to human readable date-time.

Low is the lowest price the bitcoin has been in the interval, same as high with the highest price.

Open is the opening price, which means the initial price in that interval, same as closing with the closing price.

Volume refers to the traffic of transaction in the given time.

The following figure is our data:

	time	low	high	open	close	volume
timestamp						
2018-05-19 11:35:00	1526729700	8244.08	8244.40	8244.08	8244.40	2.240500
2018-05-19 11:40:00	1526730000	8244.39	8245.00	8244.40	8245.00	3.743579
2018-05-19 11:45:00	1526730300	8244.99	8269.34	8245.00	8269.34	24.687894
2018-05-19 11:50:00	1526730600	8269.34	8274.00	8269.34	8274.00	9.368873
2018-05-19 11:55:00	1526730900	8270.84	8368.31	8273.99	8368.31	257.014080

Figure 3.1. Bitcoin price data.

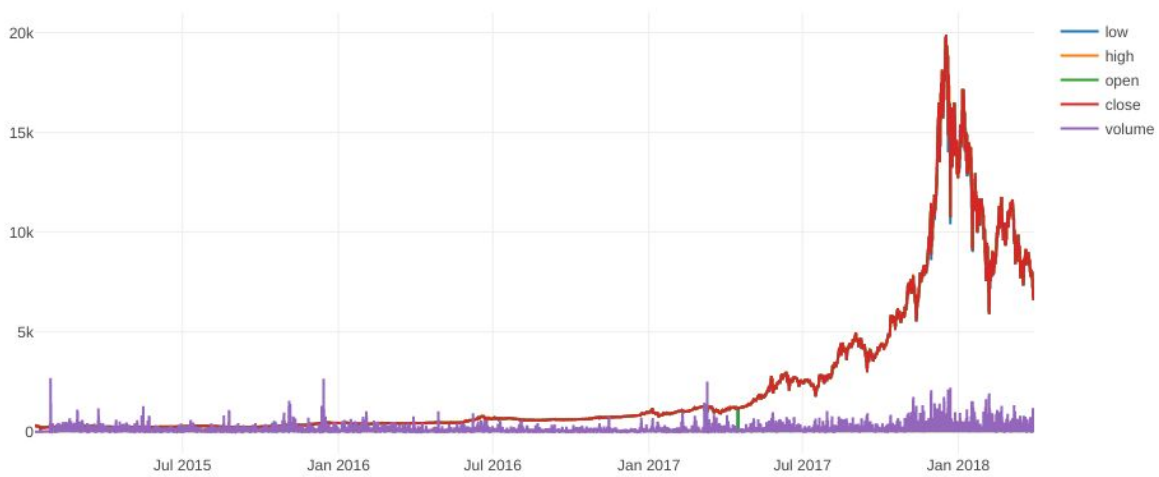


Figure 3.2. The plot of bitcoin price historical data.

As the most interesting part was from 2016 to 2018, we filtered the historical data, as well as the forum posts data to that range, and built our model based on that.

4. Topic Models: LDA

As our main goal is to find relations between forum talks and bitcoin price changes, we also considered topic modeling in order to include topic distributions of the posts as an additional feature in our model to have better results. Topic modeling is described as a way of finding a group of words (topics) from a set of documents, which can be a collection of sentences, that best shows the information of each of the documents. It can also be considered as a method of text mining – a way of obtaining recurring patterns of words in textual context [15]. There are many techniques of such modelings, however, the widely used one is the Latent Dirichlet Allocation (LDA), which we used in our model to separate our posts into topics [3]. The LDA model discovers the different topics that are contained in a particular document and the portion of the topic that is present in a document. We took bitcoin price value per 5 minutes as labels of our model; hence as input we took the set of all the posts that were made during every 5 minutes. So the LDA model was constructed on the set of documents, which are represented as bags of posts per 5 minutes. We built a model with 15 topics. Then for each document we got the weights of the topics, and added those weights as additional features to our input data for the neural network.

4.1 Data Preparation and Transformation

Data preparation is very important in order to build a meaningful topic model, because documents may contain a lot of nonsensical words which will interfere generating useful topics. So the following transformations were performed to prepare the data.

- Tokenizing: converting a document to its atomic elements. In our case, we are interested in tokenizing to words [1].
- Stopping: removing meaningless words. Certain parts of English speech, like conjunctions (“for”, “or”) or the word “the” are meaningless to a topic model. These terms are called stop words and were removed from our token list [1].
- Stemming: bringing words, that have equivalent meaning, to the same term. For example, the words “stemming”, “stemmer”, “stemmed”, should be interpreted as the same; hence stemming reduces those words to “stem”. So this is very important, because otherwise our model would view those words as separate entities and will reduce the frequency of that model which will influence the decision of the topics. For this purpose

the python implementation of Porter stemming algorithm was used to stem the words, which performs different operations to remove unnecessary parts from the words by having lists of English suffixes and prefixes in accordance with their usage rules [2].

4.2 Constructing a document-term matrix

Now at this stage, we have a tokenized, stopped and stemmed list of words. Then we constructed a list of lists, one list for each of our original documents, which means we grouped all posts for each subject into one list [3].

As was already mentioned, in order to build an LDA model we need to get the frequencies of the terms within the documents. To do that we constructed a document-term matrix, which will contain the frequencies of the words for each of the documents. First, we assigned a unique integer id to each unique token then collected word counts and relevant statistics. As a result we got an object called corpus, which represents a list of vectors equal to the number of documents. In each document vector is a sequence of tuples. For example one of the vectors can be: [(0, 2), (1, 1), (2, 2), (3, 2), (4, 1), (5, 1)], which represents one of the documents. The tuples are (term ID, term frequency) pairs.

4.3 Applying the LDA model

Now we have a document-term matrix and we generate an LDA model using gensim library [3]. Our LDA model is then stored. This model has a lot of functions that provide us with the needed information. For instance, we can review our topics with the `print_topic` and `print_topics` methods. Also we can get the topics for each of our documents. Hence by having this model, we can understand which topics were most popular in particular periods of time and which kind of impact does that have on the performance of the neural network predictions.

4.4 Results

The obtained 15 topics are represented as array of words with their weights. So the overall picture of the topics is the following:

```

(0, '0.044*country" + 0.033*ban" + 0.024*china" + 0.022*government" + 0.020*news" + 0.018*economy" + 0.014*economic" + 0.014*world" + 0.010*crisis" + 0.010*state')
-----
(1, '0.098*gold" + 0.037*fork" + 0.022*silver" + 0.020*hard" + 0.014*network" + 0.014*mate" + 0.014*lightning" + 0.012*proper" + 0.012*skill" + 0.011*segwit')
-----
(2, '0.162*market" + 0.043*stock" + 0.037*crypto" + 0.018*cap" + 0.018*cryptocurrency" + 0.015*trade" + 0.014*investor" + 0.014*asset" + 0.013*billion" + 0.012*value')
-----
(3, '0.020*know" + 0.019*people" + 0.019*say" + 0.017*like" + 0.016*one" + 0.014*go" + 0.013*don" + 0.012*thing" + 0.011*even" + 0.011*really')
-----
(4, '0.219*usd" + 0.088*position" + 0.034*bch" + 0.029*co" + 0.019*positively" + 0.014*ratio" + 0.013*close" + 0.011*toã" + 0.010*pessimistic" + 0.010*trade')
-----
(5, '0.026*invest" + 0.026*money" + 0.023*good" + 0.020*buy" + 0.019*make" + 0.017*investment" + 0.017*coin" + 0.017*time" + 0.015*profit" + 0.015*people')
-----
(6, '0.030*business" + 0.026*work" + 0.015*blockchain" + 0.015*job" + 0.013*family" + 0.012*wallet" + 0.011*success" + 0.009*key" + 0.009*help" + 0.009*social')
-----
(7, '0.047*mine" + 0.038*supply" + 0.032*transaction" + 0.029*demand" + 0.026*fee" + 0.023*exchange" + 0.020*coin" + 0.017*cost" + 0.015*rise" + 0.013*high')
-----
(8, '0.024*entry" + 0.022*assurance" + 0.021*image" + 0.015*theã" + 0.013*click" + 0.013*bitcoin" + 0.011*nope" + 0.010*consistent" + 0.010*ofã" + 0.009*lift')
-----
(9, '0.031*price" + 0.029*market" + 0.020*go" + 0.014*see" + 0.012*time" + 0.010*back" + 0.010*sell" + 0.009*buy" + 0.008*panic" + 0.008*news')
-----
(10, '0.081*price" + 0.033*year" + 0.032*rise" + 0.019*go" + 0.018*time" + 0.015*happen" + 0.015*high" + 0.014*reach" + 0.013*see" + 0.011*value')
-----
(11, '0.073*campaign" + 0.025*signature" + 0.022*management" + 0.016*discipline" + 0.016*join" + 0.012*primary" + 0.012*divide" + 0.011*vote" + 0.011*register" + 0.010*withdrawal')
-----
(12, '0.096*ico" + 0.050*project" + 0.034*icos" + 0.024*scam" + 0.024*facebook" + 0.012*twitter" + 0.012*watch" + 0.010*game" + 0.010*youre" + 0.010*kyc')
-----
(13, '0.034*currency" + 0.021*people" + 0.021*crypto" + 0.021*bank" + 0.018*money" + 0.014*country" + 0.014*world" + 0.013*government" + 0.010*make" + 0.009*cryptocurrency')
-----
(14, '0.025*index" + 0.024*org" + 0.024*topic" + 0.023*budget" + 0.023*bitcointalk" + 0.020*emotion" + 0.017*charity" + 0.014*content" + 0.014*merit" + 0.012*pray')
-----

```

Figure 4.1. LDA topics.

Based on these topic representations we can see that indeed there are meaningful topics in the documents. We can notice that the discussions of the forum are about important topics which indeed can influence the price fluctuations. We can even label these topics. For instance, if we look at topic number 2, based on the represented words, we can conclude that this topic is mainly about stock investments in the market. As another example, we can take topic number 5. In this case we can even make a complete meaningful sentence with the provided words. An example of such sentence can be: "It is good time to invest money and buy coins to make a profit". Another interesting topic is the 10th. We can see that this topic is about future price rise.

To sum up the results of LDA topic modeling, we can say that, the topics of the forum discussions are indeed related to bitcoin. Hence, it is logical to make a hypothesis that the discussions around the relevant topics may play a significant role in bitcoin price fluctuations. Therefore as we became confident about the propriety of the topics, we decided to include the weights of the 15 topics for each document as an auxiliary feature to our neural network (see section 6).

5. Sentiment analysis

5.1 Sentiment analysis of forum posts

After dividing our documents into topics and constructing the LDA model, it is logical to understand how positive the posts of each document were. To that purpose we use TextBlob library to evaluate documents from our data set. TextBlob gives two values polarity (from -1 to 1), that shows the positiveness of the data and subjectivity (from 0 to 1), which shows how subjective the data is [6]. Next, these two values also were added to the input data of the predictive model as auxiliary features.

TextBlob example:

Text	Polarity	Subjectivity
look way sub even perfect price buy majority people see sub coin	1 (positive)	1 (subjective)
price go halve people literally retard	-0.9 (negative)	1(subjective)
argument support fact general trend price dip well	0.05 (neutral)	0.5 (less subjective)
zero loss formula review software peter morgan muler	0 (neutral)	0 (not subjective)

5.2 Bitcoin price change Peaks

We followed a general algorithm for finding local minimum and maximum in the bitcoin price change data. According to our algorithm, a point is considered a maximum peak if it has the maximal value, and was preceded (to the left) by a value lower by delta, and followed (to the right) by a value lower by delta. The following plot shows the result of detecting maximum and minimum points on bitcoin price change data. In next section we will see the relation of the sentiments to such peaks



Figure 5.1. Bitcoin price peaks.

5.3 Relation of sentiment analysis to price changes

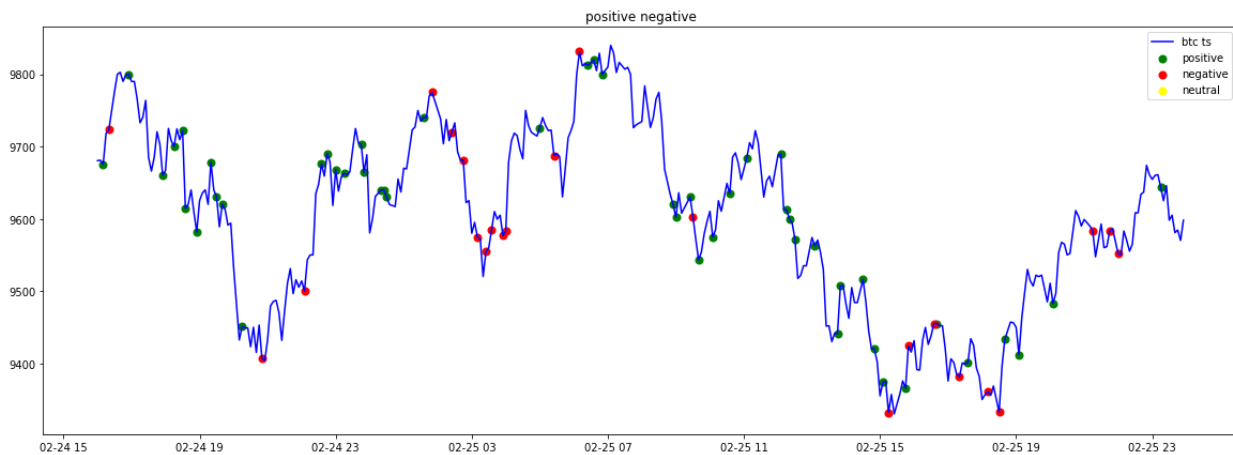


Figure 5.2. Sentiment analysis and price change relation.

From the figure above we can notice some correlation of sentiments with price changes. For instance, in some regions, the positiveness is correlated with the increases in the price, and negativeness is correlated with the decreases. However there are also some regions where this correlation is not observed. The obtained correlation factor is not so high it is about 3%. All in all, we added the sentiments to the neural network in order to see how it helps to make better predictions.

6. Neural Network

6.1 Model Architecture

Next, we started to construct a neural network that predicts bitcoin price based on our data, which is a set of user posts with some auxiliary features. We built Keras neural network [7]. As the network required the inputs to be vectors, we decided to use StarSpace word embedding in order to map words to a vector in a space that has high dimensions. These word embeddings give semantic and syntactic information of the words. For instance, the words, which are similar, are close to each other in this space and the words which are dissimilar, are far apart [8].

Before training the model we labeled the data. We tried different methods of labeling. First we calculated the labels as the logarithm of the ratio of the average price of bitcoin in next 5 minutes to the average price of previous 5 minutes: $\log\left(\frac{\text{Average Price Next 5 Min}}{\text{Average Price Prev 5 Min}}\right)$. However, after training our model we got very bad results, our model was not learning anything and the loss was not decreasing. Then we took the label as the normalized values of bitcoin price for each 5-minute data and got better results. Moreover, we tried different ways of normalizing and the results for each method are represented in section 7.

The main input to the model was the collection of the posts for every 5 minutes, represented as a sequence of words which we got from StarSpace in a vector form. But to spice things up, we also added auxiliary inputs to the model, such as polarity and subjectivity, whose correlation to the data was 3%. Then we included the volume of transactions within each 5 minute as another feature, whose correlation was 12%. And finally we added the 15 weights of the topics of our LDA model as new auxiliary features. By having such amount of features we examined their different combinations in the neural network and represented all obtained results in section 7. To combine the main input data (the posts) with the auxiliary features we first passed our vectors to the LSTM layer. LSTM transformed the sequence of the vectors into a single vector, containing information about the entire sequence. Then we fed into the model our auxiliary input data by concatenating it with the LSTM output. Next we stacked three deep densely-connected networks with different amounts of neurons and finally applied the main logistic regression layer with sigmoid activation as our labels range from 0 to 1.

6.2 Model validation.

Our final filtered data contains 351666 forum replies, which we transformed into 5-minute bags, which made a total of 141981, and divided randomly to three sets: train, test and validation with the respective ratios: 80%, 10%, 10%. To measure the performance of our model we used the values of loss function measured in mean squared error, as well as the R squared score which measures how the data points are close to the fitted values. Assuming that the best naive model without training it on the data is to predict the mean value of the initial label, we can compare our model to the mean model and measure how well it does. The R-squared score is calculated by using the following formula:

$R^2 = 1 - \frac{TSS}{RSS}$ where TSS is the sum of square of difference between the real value and the predicted, and RSS is the sum of squares of difference between the real value and the mean value.

In other words, R squared is used as the indicators of the goodness of our model. The values of the indicators, as well as the plots of the losses, are represented in section 7. Trying different architectures and different combinations of features, we obtained different results which are shown below.

6.3 Feature Summary:

- StarSpace converted our words into 100 dimensionality vectors, followed by an LSTM of 32 neurons and three fully connected layers which takes the output of LSTM and the auxiliary variables as its input. The input and output size of the fully connected layers is the same size as the sum of LSTM output and number of auxiliary variables.
- Word padding refers to the dimensionality of the initial sentence input; for example having 150 word padding with a sentence containing 100 words, means adding empty words to the sentence to make the dimensionality 150, whereas the sentences that have more than 150 words, will cut it up to 150 words.

The following is the histogram of the number of word in our messages. It goes up to 2000 words.

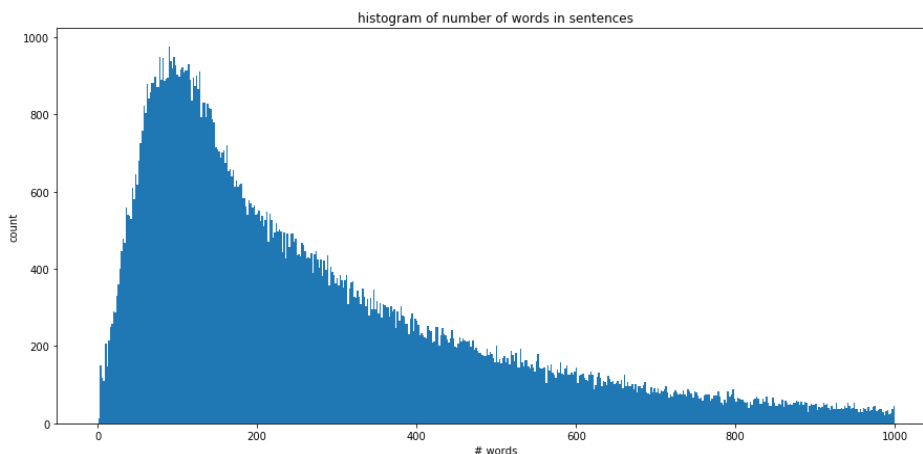


Figure 6.1.
Histogram of
number of words
in sentences

Our experiment showed that using padding of 750 words reduce the complexity of the algorithm and doesn't affect the loss.

- Sigmoid is the activation function which is used in the fully connected layers as well as the last layer.
- LDA topic refers to the probability vector of each forum to belong to one of the 15 LDA trained topics.
- Price label, we used two types of normalization. In the Min Max case we normalized the price to fall into the range 0 to 1. And in the Uniform min max case, we took the logarithm of the price and then normalized it to fall in the range 0 to 1 which gave almost a uniform distribution.

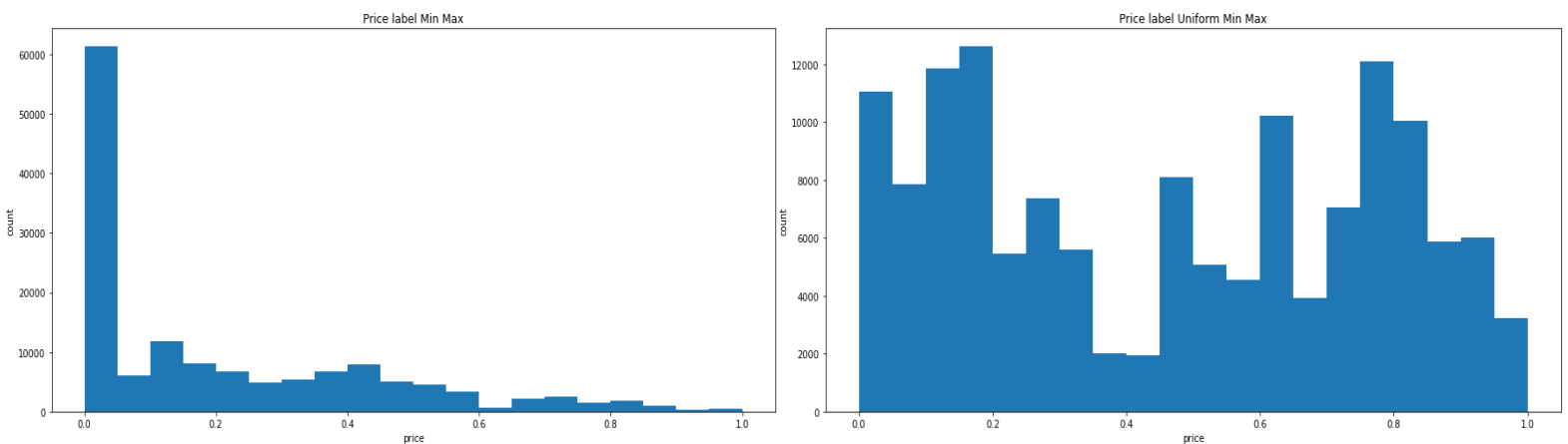
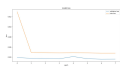
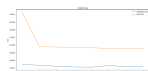
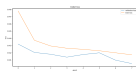
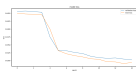
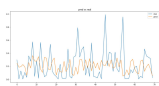
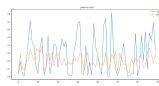
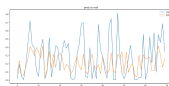



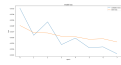
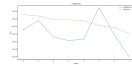
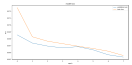
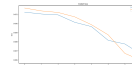
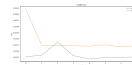
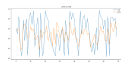
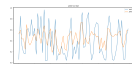
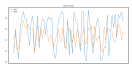
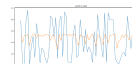
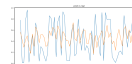
Figure 6.2: Price label histograms:Price label min max vs price label uniform min max.

6.4 Model results:

This experiment is done to try different word paddings in case of the non-uniform price labeling (min max), and one experiment using LDA topics.

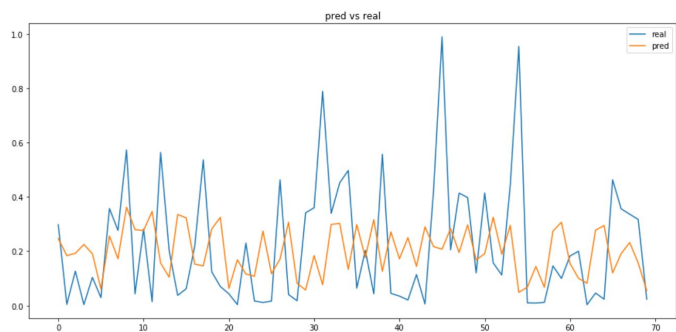
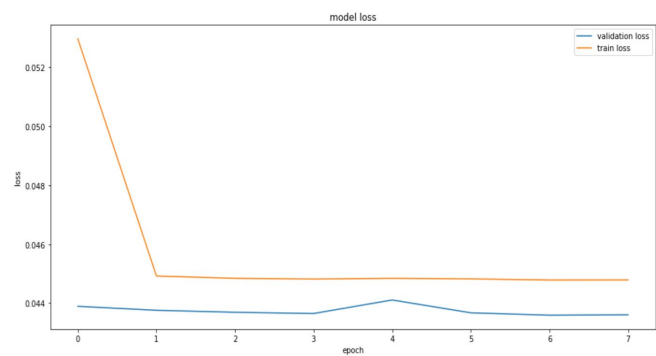
	Model 1	Model 2	Model 3	Model 4
Word padding	1500	750	750	150
Price label	Min Max	Min Max	Min Max	Min Max
LDA topics	False	False	True	False
polarity	True	True	True	True
subjectivity	True	True	True	True
volume	True	True	True	True
epochs	8	8	8	15
RMSE Train set	0.044	0.044	0.041	0.028
RMSE Test set	0.045	0.046	0.042	0.029
RMSE validation set	0.043	0.0436	0.040	0.030
R squared Train set	0.163	0.161	0.224	0.470
R squared Test set	0.172	0.167	0.225	0.446
R squared Validation set	0.165	0.165	0.225	0.444
# hours to train on CPU	2.8	0.33	1	0.33
Train_val loss				
predictions				

This experiment is done to try different combinations of auxiliary variables in case of the uniform price labeling (uniform min max).

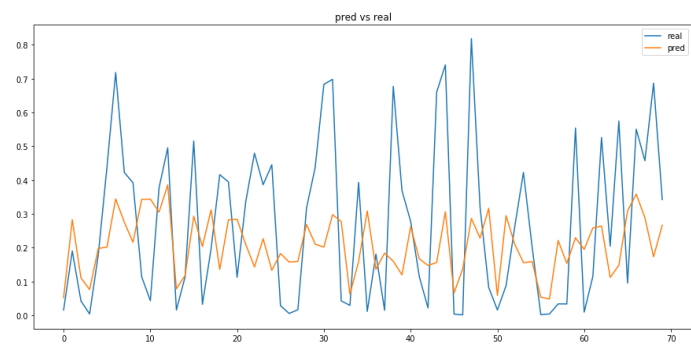
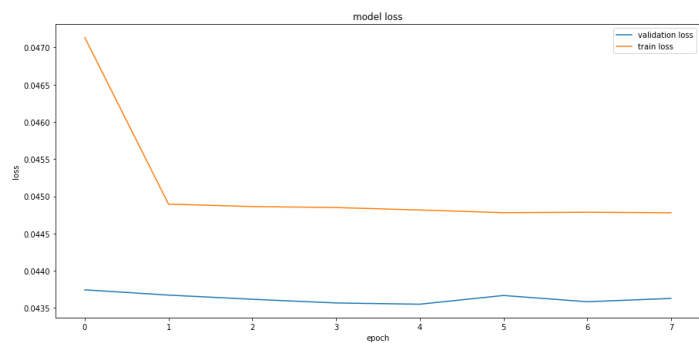
	Model 5	Model 6	Model 7	Model 8	Model 9
Word padding	1500	750	750	750	750
Price label	Uniform min max	Uniform min max	Uniform min max	Uniform min max	Uniform min max
LDA topics	False	False	True	False	False
polarity	True	True	True	True	False
subjectivity	True	True	True	True	False
volume	True	True	True	False	True
epochs	8	8	8	8	8
Loss Train set	0.072	0.722	0.065	0.085	0.072
Loss Validation set	0.072	0.071	0.653	0.086	0.072
RMSE Train set	0.072	0.071	0.065	0.086	0.072
RMSE Test set	0.072	0.072	0.066	0.087	0.073
RMSE validation set	0.072	0.071	0.065	0.086	0.072
R squared Train set	0.208	0.212	0.283	0.056	0.204
R squared Test set	0.214	0.218	0.282	0.056	0.211
R squared Validation set	0.211	0.214	0.283	0.049	0.206
# hours to train on CPU	1.25	0.83	0.66	0.53	1.2
Train_val loss					
predictions					

The following plots are dedicated to the Train and Validation loss depending on epoch (left side part plot), and predictions on a part of test data set (right side part plot) for each model.

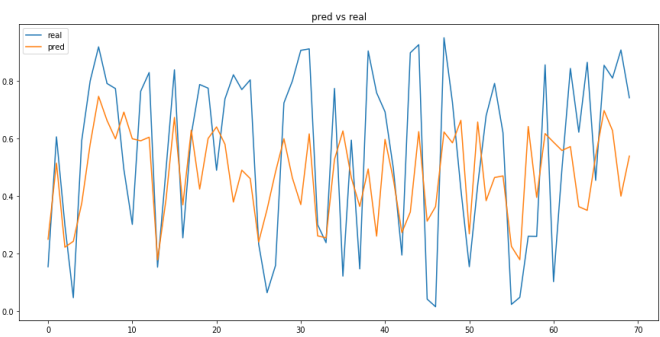
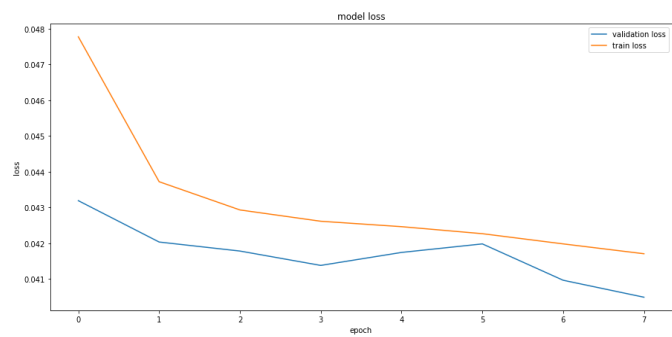
Model 1:



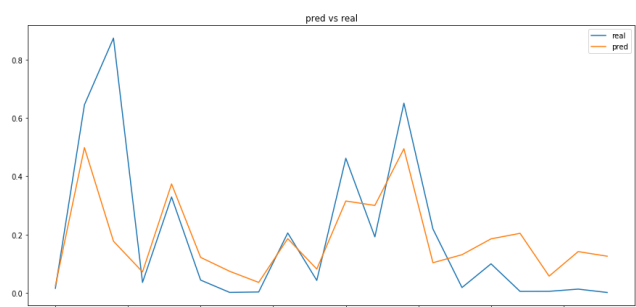
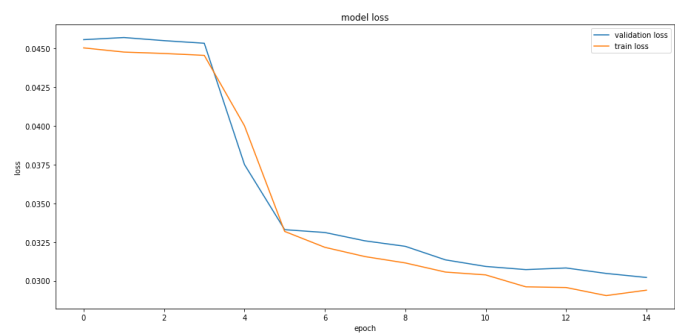
Model 2:



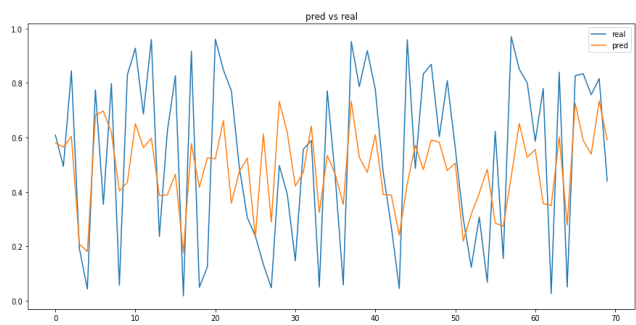
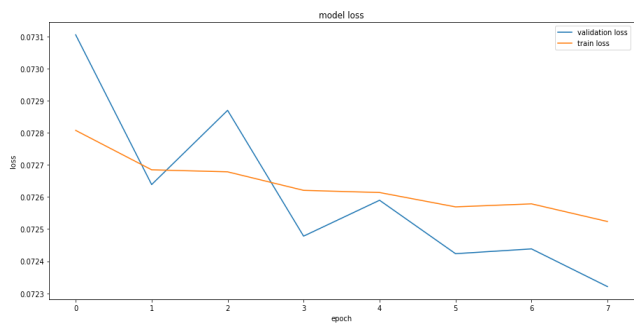
Model 3:



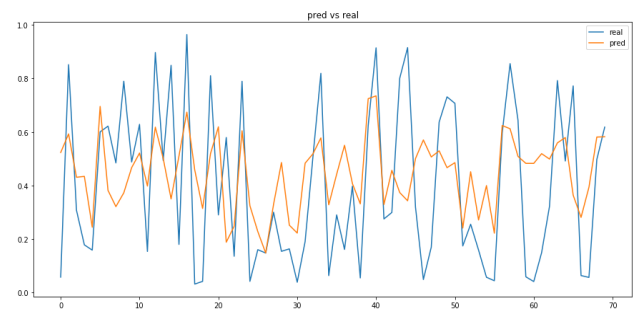
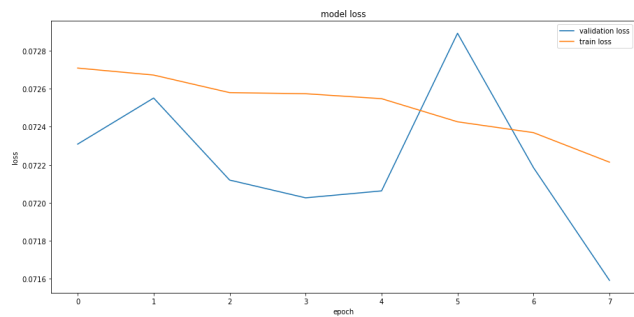
Model 4:



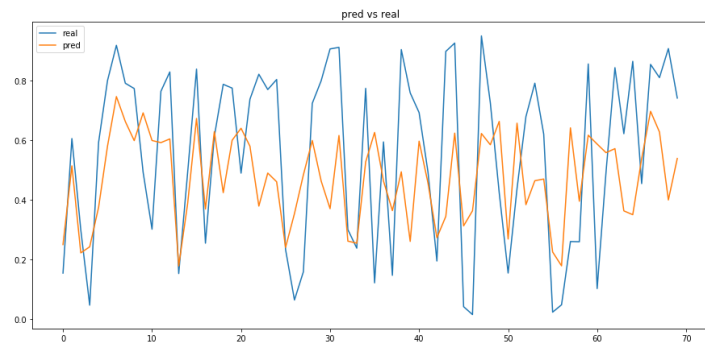
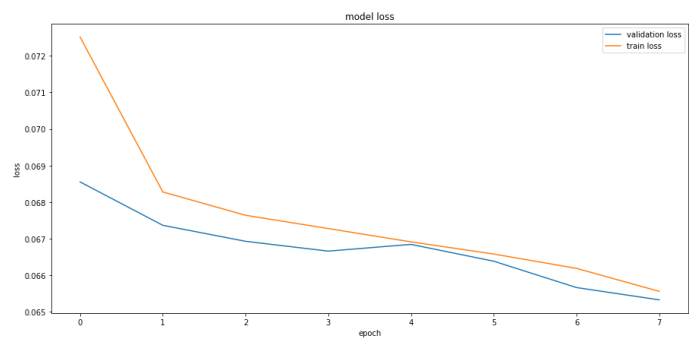
Model 5:



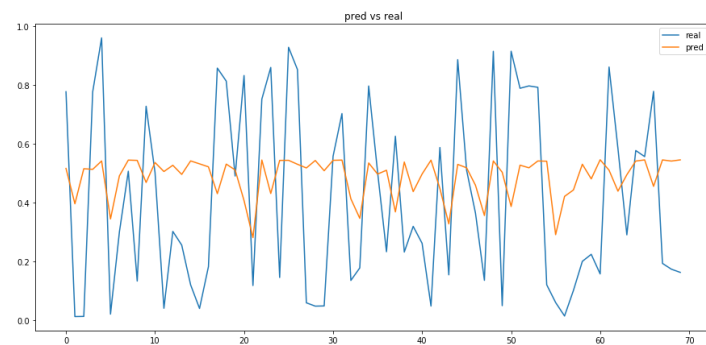
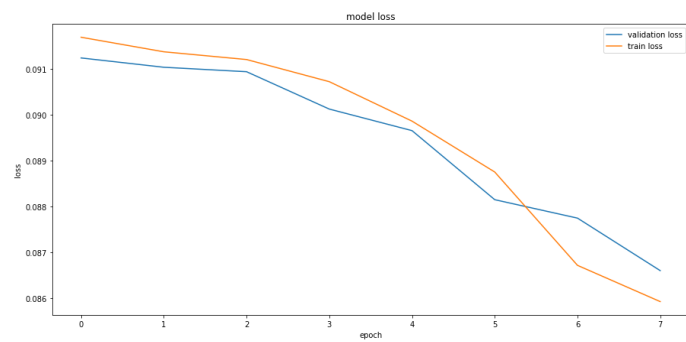
Model 6:



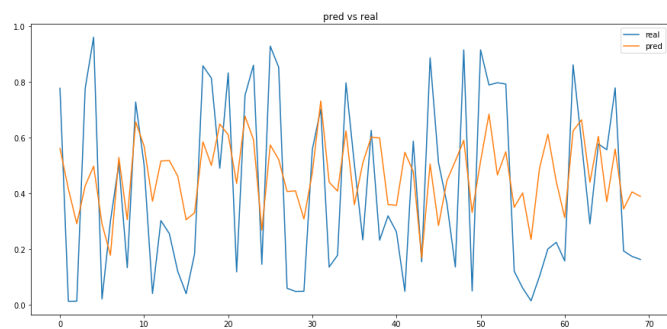
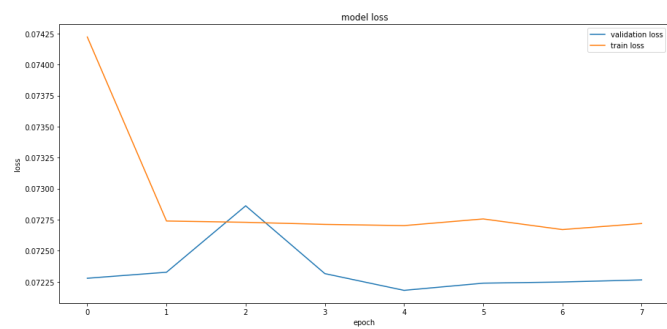
Model 7:



Model 8:



Model 9:



8. Model Evaluation and Summary

From the obtained results we can conclude that the model works pretty well taking into account that the data was not so much and was taken only from one forum. We can see that the highest R squared score was obtained in case of having only 150 words in each document, however, this is not a good prediction because the large amounts of posts were omitted in this case. Next, we can see that changing that value from 750 to 1500 does not make much difference, because there are not so many documents with a length larger than 750. Min max price labeling gives better R squared score, and less loss, however, the distribution of the price is right skewed, which means we have more chance to be near the real value by predicting the value near the left peak.

Although uniform min max gives more loss and less R squared, but statistically it is more accurate to do the experiment on. Using uniform min max labeling, with 750 we have R squared score about 0.22. However, when we added also LDA topics' weights as a feature the model worked better and we got R squared score 0.28. By this result we can say that our hypothesis made in section 4.4 was true; indeed LDA topic weights improved the R squared score. Hence we can say that topic distribution of the posts influence price fluctuations. And we should also notice that in case of including LDA topics the uniform normalization gave a higher score than the other one. Moreover, the results of adding the auxiliary variables of polarity, subjectivity and volume also increase the wellness of the model.

In conclusion, from the plots of the losses of our model we can see that in addition to not bad accuracies of value predictions, the model predicts the moments of increases and decreases of the price very well. And for further improvements we will try to test our model on larger dataset from different forums as well as social media.

9. Code Sources

All the source codes are provided in the following GitHub repository:

https://github.com/Gasia44/Capstone_Project

References:

1. Blei D. (2003). *Latent Dirichlet Allocation*. Retrieved from: <http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
2. Porter M. (2006). *The Porter Stemming Algorithm*. Retrieved from: <https://tartarus.org/martin/PorterStemmer/>
3. Barber J. (n. d.). *Latent Dirichlet Allocation (LDA) with Python*. Retrieved: https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html
4. Scrapy 1.5 documentation. (n.d.). Retrieved from <https://doc.scrapy.org/en/latest/>
5. D. (n.d.). Danpaquin/gdax-python. Retrieved from <https://github.com/danpaquin/gdax-python>
6. Tutorial: Quickstart¶. (n.d.). Retrieved from <http://textblob.readthedocs.io/en/dev/quickstart.html>
7. Chilamkurthy, S. (2017, January 05). Keras Tutorial - Spoken Language Understanding. Retrieved from <https://chsasank.github.io/spoken-language-understanding.html>
8. Getting started with the Keras functional API. (n.d.). Retrieved from <https://keras.io/getting-started/functional-api-guide/>
9. Kim, Y. B., Lee, S. H., Kang, S. J., Choi, M. J., Lee, J., & Kim, C. H. (2015). Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4524693/>
10. Linton M, Teo EG, Bommers E, Chen CY-H, Härdle WK. Dynamic Topic Modelling for Cryptocurrency Community Forums. 2016.
11. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of machine Learning research. 2003;3(Jan):993–1022.
12. Wang C, Blei DM. Collaborative topic modeling for recommending scientific articles. Retrieved from <http://www.cs.columbia.edu/~blei/papers/WangBlei2011.pdf>
13. Wang Y, Agichtein E, Benzi M. TM-LDA: efficient online modeling of latent topic transitions in social media. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 2012:123–31.
14. 28. Xianghua F, Guo L, Yanyan G, Zhiqiang W. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. Knowledge-Based Systems. 2013;37:186–95.
15. KDnuggets. (n.d.). Retrieved from <https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html>