Buzzword based text-features for financial time series prediction at the example of Reddit and Bitcoin

Mario Innerkofler
mario.innerkofler@student.tugraz.at
Graz University of Technology
Graz, Austria

ABSTRACT

In this elaboration we aim to come up with a reasonable collection of text based features for predicting future bitcoin market events. We collect comments from English speaking Subreddits and identify a suitable set of ubiquitous trading related buzzwords by applying a Word2Vec embedding. We define future market events with the triple barrier labeling method and regress on the text based features. We then detail the relation between occurrences of buzzwords and the future price and showcase the applicability via a small simulation run.

KEYWORDS

cryptocurrencies, nlp, bitcoin, word2vec, logistic regression, trading simulation

1 INTRODUCTION

Cryptocurrency markets form an attractive environment for the application of automated trading bots. Not only is it the high accessibility and availability, but massive volatility outburst provided chances of large returns in short time periods. One prominent assumption on market returns is that they are close to unpredictable. While this may be true in the long run, we could observe periods of hype and resentment in the crypto currency world. In 2017 the Bitcoin price for example increased by a factor of twenty. We aim to predict periods of notable market shifts by tracking the topics and discussions conducted on Reddit. To that end we keep track of word counts of commonly used trading terms which we derive by expanding a predefined set of such buzzwords. We then will use logistic regression to evaluate the impact of the individual average term frequencies computed over 4 hour windows. We then use a 4-fold cross validation scheme to evaluate the predictive quality of the derived features and use the prediction to fuel a naive trading simulation.

2 FINANCIAL TERMINOLOGY AND SETTING

We aim to keep the extent of this section to a minimum. However a certain setup is needed to highlight the difficulties of deriving text based features. First of all, let us introduce the quantities which define the market price. Typically we can not measure the market price at any given moment in a continuum, but rather are given equidistant discrete measurements of the market which are aligned with the beginning of a day, a month or a year. The time in between these measurements will be called the interval. In our case we work on 5 minute intervals which are aligned with the start of each day. Each measurement comprises five numbers. The open, high, low and close price and the volume. The open and the close price are measures of the market value at the beginning and at the end of the 5 minute period. The high and low price are the largest and smallest price observed within the particular 5 minute time frame. The volume simply is the amount of currency traded in this very time period. These 5 minute bundles of data are oftentimes called candles (or candle sticks) and each candle is oftentimes visualized like a boxplot. In the following figure we display the bitcoin ohlcv-data at our hands by means of a day-based candlestick chart.

Figure 1: Candlestick chart; 1 day intervals

For our purposes, the sequence of close prices will be used as a reference and we do not mind price shifts which occur within individual candles. The sequence of close prices will simply be referred to as the price. In order to get the price data we used the python library ccxt. We fetched the bitcoin data from 2020-09-01 00:00:00 until 2021-03-09 00:00:00 from the broker “binance" which we make sure is given in unix_time (UTC), just as our Reddit comments will be dated. Already from the above figure 1 it is visible that the covariance structure shifts over time and that there is a trend. Thus models based on the financial data need to take this into account but our investigation is not greatly influenced by this.

2.1 Triple barrier labeling method

Finally let us define what it is exactly what we want to predict. We are in a classification setting and roughly speaking we want to distinguish between rising, falling and sideways (i.e. close to constant) markets. We use the triple barrier labeling method described in the book by Lopez de Prado [1, p.45]. In order to define the label, we need to agree on two defining constants i.e. the gap ρ and the lookahead l. In our case we use a gap of $\rho := 0.5\%$ and a lookahead l of 1 hour (i.e 12 5-minute candles). Now the label y_t at time t is defined as follows. We consider values $x_t(1 \pm \rho)$ which defines the upper, respectively lower barrier (a price movement of $\pm\rho$ percent). Now let i run through $1 \dots l$. As soon as x_{t+i} breaches one of these horizontal barriers (i.e. exceed, respectively undermines $x_t(1 \pm \rho)$), we grant label 2 (uptrend) if that event occurred due to an upper barrier breach and a label 0 (downtrend) if the bottom barrier was breached. If no barrier is breached for any $i \in \{1, \dots, l\}$ the label 1

(sideways) is granted. This means the price went through a vertical barrier at x_{t+l} without touching any of the above/below limits. This labeling method exactly corresponds to the nature of filing limit orders which are stopped if certain price thresholds are breached. At the end of a time period (label 1) we are given the opportunity to make or elongate our trade.

3 THE DOMAIN OF REDDIT COMMENTS

Reddit is an online platform where users can participate in online discussions and exchange opinions. For reading on Reddit, no account is required but in order to submit one needs a free account. Reddit is structured in various isolated units (called Subreddits) which users can post submissions to. A submission usually starts a discussion of the particular sub topic address in the submission. Submissions can be commented on by users and also comments can be commented. In this way Reddit obeys a natural tree-like structure where each node is formed by a comment, the roots of said trees are the submissions. In our setup we only take a look at comments since they better capture opinions rubbing against each other. We ignore Godwin's law and to not truncate the trees at some height. Moreover we ignore the tree-like structure and the belonging to different Subreddits completely. We simply collect all comments, marked by a time-stamp, which were posted within our time period of choice. This pool of sequenced documents defines our text corpus. We used the PushshiftAPI to scrape the contents of nine Subreddits in the time from 2020-09-01 00:00:00 until 2021-03-09 00:00:00. Not only do we collect the text body and the date, but we also scrape some meta data such as the number of up-votes to be well prepared for advanced modeling approaches. The names of the Subreddits and the amount of comments on our disposal are displayed below in Table 1.

subreddit_shname	number of comments
Bitcoin	875173
BitcoinBeginners	198634
BitcoinMarkets	134693
CryptoCurrencyTrading	4384
CryptoMarkets	37332
Crypto_Currency_News	2402
SatoshiStreetBets	324322
btc	149753

Table 1: Subreddits; number of comments

3.1 Processing the comments

First of all, we send each document through a traditional NLP data-cleaning pipeline were the below operations are applied as ordered below.

(1) Drop all rows which contain a null-value (i.e. missing date, missing title, or similar)
(2) Set all letters to lower case
(3) Drop comments of the form "[deleted]" or "[removed]".
(4) Drop all links. This is done by dropping every token containing ".com" or "http".
(5) Drop all line breaks, tabulators and similar.
(6) Drop every string which is not a letter such as numbers, punctuation, symbols and emojis.
(7) Reduced repetitions of more than 2 letters to a double-letter occurrences (e.g. moooon -> moon). We do the same for repetitions of more than 2 groupings of two letters (e.g. hahahah -> haha).
(8) Replace each occurrence of the word "btc" by "bitcoin" and identify each occurrence of "cryptocurrency" or "cryptos" by "crypto".
(9) Drop stopwords. We use the gensim variable STOPWORDS as a basis which we slightly modify by including and dropping and some other words which may be of (no) importance. The exact list of stopwords can be retrieved by using our code basis.
(10) Apply textblob's lemmatization algorithm.
(11) *Optional: gensim's implementation of Portes's stemming algorithm, we did not use this.*
(12) Drop all tokens with less than 3 letters
(13) Remove leading/trailing and multiple blank spaces
(14) *Drop all documents with less than 3 letters, this is superfluous by (12). It may however be useful when other thresholds are applied*
(15) Drop duplicates of postings (automated alerts, spam and similar)

At this stage we obtain a corpus which we refer to as the clean Reddit corpus.

3.2 Treatment of outliers

Next up we investigate the frequency of comments which provides further insight into the quality of the data. From Reddit we scrape the attribute "created_utc" which provides us the time at which a comment was posted. We roll a one hour window over the data and collect at each comment the number of comments being posted within the last hour. We obtain the following time series.

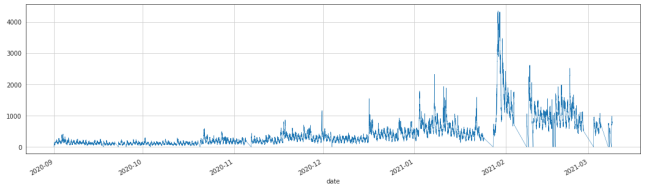

Figure 2: Frequency of comments

We spot that there are longer periods where no comments were posted, followed by a large spike. This behavior raises suspicion towards the correctness of the time stamps. Pushshift.io reported problems with database overload due spam and this may be the cause of these irregularities [https://bit.ly/3rcXoPS]. The exact periods of times where no comment was posted within the last hour can be found in the appendix, Table 2. Based on this, we decide to end our time window of consideration early on 2021-01-27 00:00:00 which is right before the gap prior to the largest spike. We include the previous irregularities since the subsequent gaps do not seem as drastic. In a more elaborate approach one could make an attempt

of spreading the comments over the previous time. For instance one could use the tree-like structure of comments where child nodes must occur later in time. We then could investigate the mean team between comments and a response to determine a fair spread over time. The most difficult issue may be to find the times where the root notes of each comment tree are located. Here one could in a naive approach simply use a random order where again the spaces are statistically motivated.

4 FEATURE IDENTIFICATION

The fundamental idea is to track the occurrence of trade-related topics in the Subreddits. One source of information which we used for this task is the PhD Thesis of Phillips, see [2]. Thereby a dynamic topic model was used. It is capable of identifying topics treated by the community and furthermore can take into account the shift of phrases being used when a certain topic is talked about.

4.1 An initial set of buzzwords

We however are not interested in discovering topics of our community but rather impose an up front idea of words being associated to trading related topics and work from there. We do not take in consideration a potential shift of words used to talk about a topic. We argue that within a period of a few months this effect is negligible. In order to lay down the foundation of the trading topics, we ask domain experts for typical (slang) words being used in such discussions. A list of buzzwords ['long', 'buy", 'moon, 'pump', 'bullish', 'sell', 'hodl' (sic!), 'hold', 'dump', 'bearish', 'bear'] was returned. In fact also the words "drop" and "rise" were mentioned. We did not include them since they very likely are also used in other contexts, also within this community. Their use may potentially boost the false positive rate. However on the other hand we want to achieve a good coverage of the topics by detecting some other typical words, i.e. maximize the rate of discoveries.

4.2 Extending the set of buzzwords

To that end we most successfully used a word2vec embedding of the clean Reddit corpus with a latent dimension of $n = 25$. Prior to learning, we removed the top 10% of the most frequent words and words which occur only 2 times or less to reduce the noise and feature space dimension. We then print the top 5 nearest neighbors of each buzzword with respect to cosine similarity of this embedding and select words with a cosine similarity of at least 0.75. We only include up to 5 similar words since we want to maintain a fair balance of information on all the intricacies hidden in the semantics of these words. Moreover for the modeling task we do not want a too large feature space (in particular we do not want collinearities). With this technique we were able to obtain an extended list of buzzwords being ['short', 'sell', 'trade', 'invest', 'move', 'accumulate', 'moonn', 'pluto', 'mar', 'dump', 'rally', 'pnd', 'moonshot', 'spike', 'bearish', 'overbought', 'overextended', 'hopeful', 'optimistic', 'buy', 'liquidate', 'rebuy', 'hold', 'trade', 'hold', 'hodling', 'hodl', 'sell', 'buy', 'invest', 'liquidate', 'pump', 'whale', 'rally', 'pnd', 'bullish', 'overbought', 'oversold', 'euphoria', 'euphoric', 'bull', 'timing', 'squeeze', 'cycle', 'bullrun']. Some words provide new semantics while others are slight variants of already known words which went under the radar by slight grammatical variants which the lemmatization could not take care of. What we do not know at this stage is that some of these words are very rarely used, imposing a strict limit on their predictive power. Now we use a K-means clustering with respect to the cosine similarity of the word2vec embedding to visualize our words within the semantic space. The number of clusters to be found is 3.

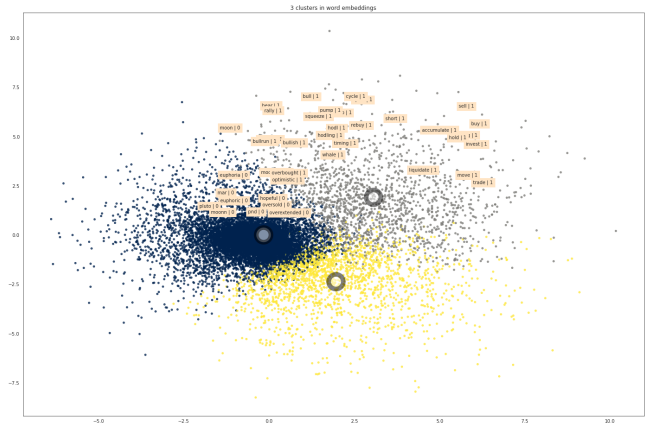

Figure 3: Clustering of buzzwords

Prior to this visualization we only knew that the thus selected nearest neighbors of each predefined buzzword were close. Now we observe that in fact all these words are located at a similar position. This means that perhaps they in fact all treat a similar topic within our community which hopefully (but quite surely) involves trading. Now we look at the cosine similarities of pairs of these words.

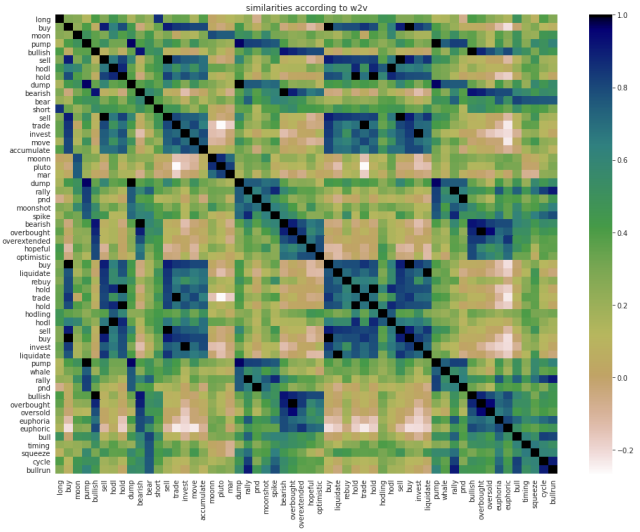

Figure 4: Cosine similarities of buzzwords

We see that certain pairs of words are regarded as highly similar. This may indicate that some aspects of our topics are well covered by several words. On the other hand some cosine similarities are very small which we interpret as different contexts being covered

by our choice of buzzwords. It is worth noting that words on a similar language level appear in similar context. For example the words 'sell' and 'buy' have a similarity of over 0.9. Also the words 'bullish' and 'bearish' show great cosine similarity.

4.3 Other means of extracting buzzwords

In other attempts we tried to use pretrained embeddings (Glove-25, Glove-100, fasttext-wiki-news-subwords-300). Two issues were encountered. The first issue is that much of the slang words were not included in the respective dictionaries. Thus neither is the particular slang trained into the model, nor can it be expected to return prominent slang words which we are not aware of. One attempt of only working on the intersection of the dictionaries has been made but the results were not as desired since the trained contexts did not match. Another drawback of these pretrained models is that the training data is not recent enough and does not catch up with the (kind of new) vocabulary used in our community.

Another technique applied was LDA. As mentioned earlier, such a model aims to identify topics in an unsupervised manner which is not of our prime concern. In the approach of Phillips such an attempt provided topics which were market related, [2]. However in our investigation this did not turn out like this and the topics seemed to rather be facilitated by different languages, garbage expressions and similar non semantic properties.

5 CONNECTING TO THE MARKET

Now having available a large set of words which appear in similar context we aim to establish a connection between the prevalence of our buzzwords to the financial domain. We leave it up to the model to decide which word should be attributed to which market trend by which proportion. This attribution than can be interpreted as a soft clustering of the words where the topics are the market trends.

5.1 Feature engineering

In order to measure the prevalence of the topics, we proceed for each buzzword as follows. We use the term frequency to count the number of times a buzzword occurs in comments. We compute a 4 hour rolling average over the term frequencies. We do so in order to factor out the number of overall comments being submitted and thus consider only the presence of the buzzword within all the text of the last 4 hours. We now use buckets of 5 minutes to even out the sample count and calculate a representative point via the median over this 5 minute bucket. This defines our feature for one buzzword at one time step. We do so in order to reduce the amount of input data and even out the amount of information going into a single prediction. We argue that within one 5 minute bucket the median well represents the 5 minute bucket since within this time frame the mean frequency counts hardly vary. Furthermore in this way it is easier to align to the ohlcv data. Alternative backward fill (of the unknown label) and forward fill (of the data) could be applied when aligning the different frequencies in the time series. We now column-wise replace values which have a z-score larger than 1.5 by a linear interpolation. This is yet another countermeasure to outliers due to time stamp impurity. We then apply min-max scaling columns wise. The following displays the thus obtained multivariate time series.

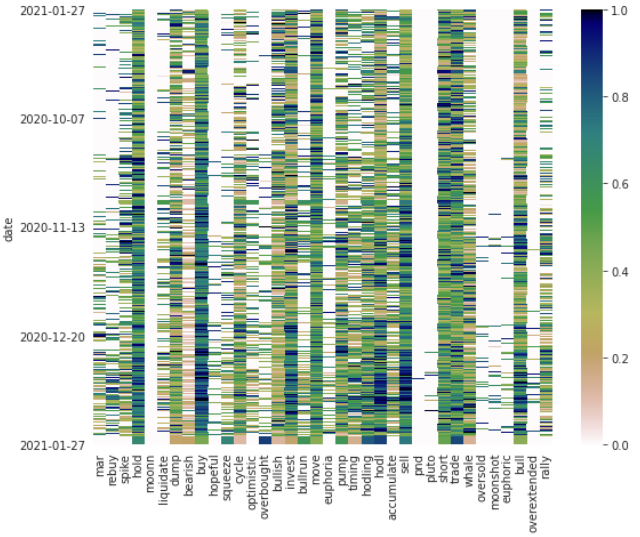

Figure 5: Text based features over time

We can observe that some buzzwords generally are more frequently used than others and that over time different intensities are observable. Some Buzzwords very rarely occur. By having applied min max scaling this can only be due to the existence of a few very large data points. This in turn can be reduced to the observation that some words simply are very rarely used.

5.2 Selecting representative features

The next step is to perform a backward reduction of features in order to get rid of collinearities. In some sense this can be seen as selecting the centroids of our soft topic clustering. To that end we train a logistic regression model on the scaled data and look at the sum of absolute values of coefficients. By having scaled the data, the regression coefficients indicate the increase in predicted probability by unit change of the feature. By aggregating the magnitude we thus claim that this represents the overall importance for our prediction task at hand. We drop the least important feature, retrain the model and iterate until a suitable threshold (0.2) is no longer undermined by any feature. This technique provides us the following reduced list of features ['rebuy', 'hodling', 'moonshot', 'euphoric', 'pluto', 'squeeze', 'pnd', 'buy', 'hodl', 'sell', 'oversold', 'hopeful', 'overbought', 'bull'].

5.3 Interpreting the impact of buzzwords

Now let us take a thorough look at the individual feature importance of this reduced set of predictors. The logistic regression model grants us insight into the individual importance of mean frequency counts for predicting a certain market situation.

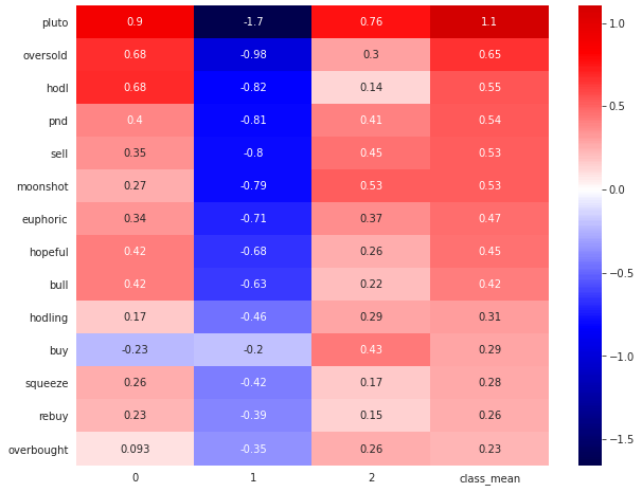

Figure 6: Feature importance of selected buzzwords

A blue color means a frequent use of this buzzword diminishes the likelihood of a future market movement in the direction indicated by the column labels (0 - down, 1 - sideways, 2 - up). The red color indicates an increase of the estimated probability. We can spot that all our buzzwords are used in higher proportion over the last 4 hours prior to up/down movements and a less dense use of buzzword is helpful for forecasting sideways movements. The difficult distinction is between up and down movements. This is quantified by similar values in column 0 and column 2 in each row. Some relations are quite straightforward to interpret. For example the word "buy" is indicative of an upwards movement, as well as "moonshot". The word "hodl" (a slang word for "hold on for dear life") is used prior down movements which reflects the evident hope of market recovery and not making a nervous selling decision. Somewhat insensible is the downward association of "oversold" and "bull" because these words would rather be associated with up trends. One thing which we need to point out is that the labels are not uncorrelated. So it may well be that only the predictions which are made already in an up/down phase are correct and the up movement is actually note detected early. This can (and should) be addressed and measured in a more advanced approach.

5.4 Assessing the quality of predictions

Now let us shift our view more to the prediction capabilities of our purely text-based model. We now split our dataset into 4 pieces of same duration and use cross validation. Each training set is further reduced by leaving out 7 days at the beginning and end to avoid information leaks. This technique, called embargoing, is described in the book by Lopez de Prado, see [1, p.105]. We then recompose the predictions on each test-fold and evaluate it via means of classification assessment. We look at the ROC-curve for each class (one vs. rest).

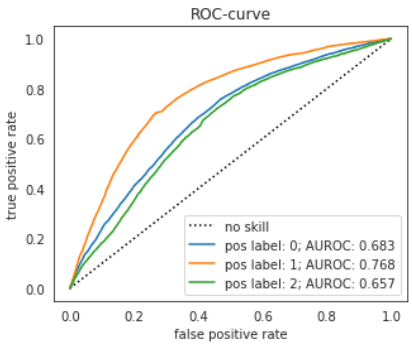

Figure 7: ROC curve

It is notable that each class can be identified better than random. It seems to be easiest to detect sideways movements. Regarding the up/down movements we still are able to distinguish between up/not up and down/not down in a reasonable manner. However what we need to bear in mind is that a confusion of ups with downs is a more expensive mistake regarding a thus derived trading strategy. In order to get insight into these ambiguities, we use a confusion matrix (normalized by the number of ground truth labels in each row).

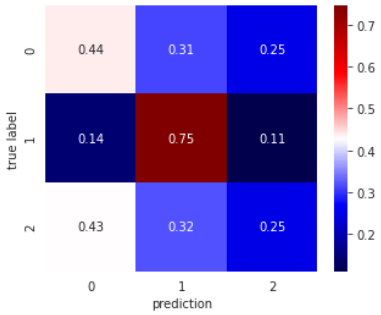

Figure 8: Confusion matrix

It becomes evident that indeed an overwhelming amount of sideways predictions are correctly predicted. Regarding the true up/down movements we see that the 0-label is quite safely predicted. In our trading strategy we will use the 0 and 1 label to decide when to leave a trade. In this sense we are quite secure when it comes to ending our trade at the right time. With the 2-label we see that we have a bad case of confusion with a lot of 0-labels. This means when the market is predicted to go up, it more often than not goes down. In this way we will open a trade in a downward market which of course is the polar opposite of what we actually want.

5.5 Converting the results into a trading strategy

Now the goal of this elaboration is to produce a strategy which uses the text information to decide whether opening a trade makes sense. In a real world application there are many more complications such as the order amount, finite amount of capital, fees and many more

hazards. Our simulation thus is to be taken with a grain of salt. How we go about this is that we bucket all predictions in one-hour slices, average the predicted label and round to the closest integer. If we still observe a 2 we open a trade (supported by predictions over 1 hour). As soon as this signal generator produces a 0 or a 1, we close the trade. A trade also is forcefully closed whence the stop loss /take profit margin of 0.5% is met. This is exactly captured by the ground truth labels. Each trade is made with a 1000$ investment and we end up with the following portfolio evolution.

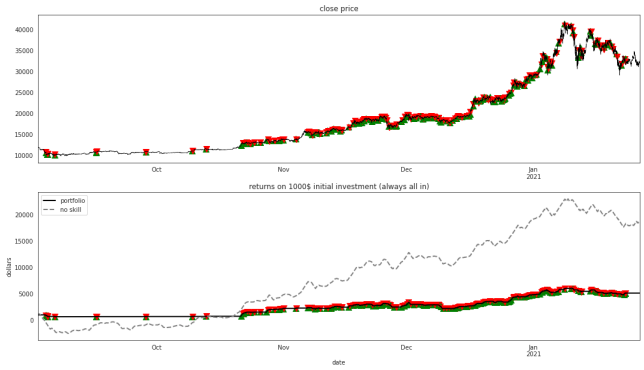

Figure 9: Trading simulation

The upper curve is the price of bitcoin over our time frame. The below figure contains the value of our account and we draw a no-skill strategy for comparison. That is, we place an order every hour no matter what and accept stop losses / take profits as need be. The green triangles indicate when our strategy enters a trade and the red ones mark when we left it. At the end of the period we had a value of about 5123$. The no-skill strategy made 18104$. In order to thoroughly assess if our strategy is better, these two numbers (and the entire simulation) are not sufficient. For example one may not necessarily want to maximize the wins, but look on the ratio of wins and losses of each trade which indicates how safe the strategy is. On the other hand we may want to take into consideration a finite amount of capital which we have on our disposal, so a trade may not always be affordable. On this note, we round up this topic and draw some final summarizing conclusions.

6 CONCLUSION

We went through various phases of handling text and extracting information from it. We started with a blank sheet of paper and first needed to decide on data sources, the time frame and the tools we need to get our hands on the data. For simplicity reasons, Reddit and an open source library for financial data provided a good playground. We then improved the quality of our data by traditional text preprocessing but also needed to fix the temporal oddities. We then claimed certain words of being indicative for certain market situations and expanded from there by using a word2vec embedding. This allowed us to identify further buzzwords being used in similar contexts. We then tracked the occurrences throughout time of these words. These time series underwent some smoothing operations in order to make them suitable as inputs for a logistic regression. This linear model was trained and the coefficients granted us insight into the composition of impacts of these word counts on individual market situations. Rounding off, we looked a bit closer at the predictive power of the model and did a small simulation to put our finding into (over simplified) work.

6.1 Tools and time

For all our experiments we use python 3.8 and a standard home PC. All models are trained on a 6-core AMD CPU and training times for each model are under 15 minutes. The scraping of the comments is a time consuming endeavor and takes approximately 24 hours.

Concluding, we hope that this small journey was informative and entertaining to the reader. We now want to point towards the code base which is accesible via the following link https://github.com/JustYeti32/reddit_btc. Feel free to expand on it or use it to address your tasks.

REFERENCES

[1] Marcos Lopez de Prado. 2018. *Advances in Financial Machine Learning.* John Wiley Sons, New York, NY.
[2] Ross Christopher Phillips. 2019. *The Predictive Power of Social Media within Cryptocurrency Markets.* Ph.D. Dissertation. University College London.

A PERIODS WITH MORE THAN ONE COMMENT WITHIN ONE HOUR

	dense block ends at	< 1 comment/hour for
0	2020-09-17 06:40:01	0 days 11:34:54
1	2020-09-21 15:35:41	1 days 03:52:15
2	2020-09-24 23:21:38	0 days 11:38:54
3	2020-10-19 16:58:22	0 days 18:37:11
4	2020-11-05 00:38:13	1 days 17:43:01
5	2020-12-03 17:57:16	0 days 09:50:02
6	2021-01-24 13:06:23	3 days 02:50:28
7	2021-02-03 20:17:50	4 days 04:03:33
8	2021-02-08 04:22:18	0 days 11:21:54
9	2021-02-09 21:03:47	0 days 06:27:48
10	2021-02-16 15:02:45	0 days 06:22:41
11	2021-02-16 23:28:12	0 days 07:14:16
12	2021-02-17 11:47:15	0 days 06:50:28
13	2021-02-27 06:08:52	3 days 10:16:06
14	2021-03-05 01:36:42	2 days 13:40:31
15	2021-03-07 18:01:11	0 days 07:02:31
16	2021-03-08 05:26:51	0 days 10:48:22

Table 2: Subreddits; periods of sparse comments