Mining r/Bitcoin Data

Collecting and Analyzing Big Data [S0K17a]
Prof. Neal Caren
Semester Assignment

Nura Kawa r0767214
Jasper Michel Schröder r0734367
(Both Master of Statistics and Data Science)

KU Leuven, Academic Year 2020-2021
Handed in on June 4, 2021

Contents

1 Introduction

Much attention has been drawn to the discussion forum *Reddit* amidst the frenzy regarding *GameStop* in the January of 2021. Retail investors gathered on the subreddit[1] *wallstreetbets* and collectively bought the shares of *GameStop*, a company in distress whose shares many institutional investors and hedge funds had bet on to go down. Because of the actions on the subreddit, however, the stock price of *GameStop* skyrocketed. As a result, some of those hedge funds suffered severe losses and were brought to the brink of collapse. This topic, therefore, has sparked a lot of interest and news coverage. In this paper, we investigate if such a relationship, i.e., between an asset's price and activity on a subreddit, exists for other assets. In particular, we have decided on the cryptocurrency *Bitcoin* in the time range January 2016 until December 2020.

Several reasons motivate the choice of *Bitcoin*. First, it is a relatively new asset and type of asset that captured a lot of interest and traction on the internet. Second, as one has to be tech-savvy to invest in or to trade *Bitcoin*, we assume that it is likely that there is a more prominent overlap with potential *Reddit* users than for a more 'common' asset such as a company's publicly listed shares. An important distinction has to be given at this point, however. For *GameStop*, there was a clear causal relationship; the action of the subreddit influenced the stock price. In this paper, we make the *a priori* assumption that if such causation exists, it goes from the price of *Bitcoin* to the activity on the subreddit.

The reason for this is the sheer difference in market capitalization. Throughout most of 2020, *GameStop's* market capitalization was below 0.5 billion USD and only surpassed the threshold of 1 billion USD at the beginning of 2021. At the peak of the frenzy, it also 'only' went up to roughly 21 billion USD (macrotrends 2021). For *Bitcoin*, however, the market capitalization at the beginning of 2016 was at approximately 5 billion USD, surpassed the threshold of 20 billion USD in April 2017 and shot up to its peak of 1.1 trillion USD in March 2021 (statista 2021). This enormous difference in market capitalizations leads us to assume that it is more likely that a causal effect goes from asset to subreddit and not vice versa.

In this paper, we extract insights on *r/Bitcoin* activity, with a focus on the following research questions:

1. What trends in forum behavior do the data reveal? How do these trends change over time?
2. Are *Bitcoin* price and volatility good predictor(s) of *r/Bitcoin* forum behavior?
3. What style, tone and topics characterize *r/Bitcoin* thread content?
4. Using features extracted from thread text, can we successfully identify *r/Bitcoin* threads posted on days with high *Bitcoin* volatility?

There are several aims in this process. Next to investigating if any meaningful interrelations between the price/volatility and thread activity occur, it is aimed to gain particular insight into the content and categorization of the texts from an NLP perspective. Also, we want to investigate if themes or phenomena mentioned in existing literature can be found or replicated in this analysis. As to a personal motivation from the authors, as this is the semester assignment of the course *Collecting and Analyzing Big Data*, we attempt to put as much as possible into action what we learned in class and transfer those skills to the task at hand.

The remainder of the paper is structured as follows. We start with a literature review, followed by a section dedicated to explaining how the data have been retrieved. Afterward, the paper is

[1] A so-called 'subreddit' can be seen as subpage of the discussion forum dedicated to a special topic.

split into two parts. In both parts, we first begin by laying out the methodology and then listing the results.

Part I deals with an analysis of the interrelation between the data retrieved from the *r/Bitcoin* subreddit and the price of *Bitcoin*. In detail, we start with an exploratory data analysis, then proceed with an analysis of thread activity, focusing on the distributions across authors and the dates the threads were posted. Following, a price analysis is conducted to compare how the price and volatility of *Bitcoin* correlate with metrics from the *Reddit* dataset. Finally, we attempt model building in which we use linear regression to (i) model the number of new threads on a day and (ii) model the number of comments of threads on a day.

In Part II, we investigate the contents of the texts in the *Reddit* threads. In particular, this encompasses computing thread features, in which we compute certain features based on the text in the data. After a brief reflection regarding data pre-processing steps such as feature engineering or tokenization, we investigate, amongst others, the Dale-Chall readability score, look into sentiment analysis and perform topic modeling. On top, by use of t-SNE, we plot the threads based on the topic models. In the final section of this part, we attempt a predictive model to model if a thread has been posted on a day in which there was high volatility for *Bitcoin*.

Then, we summarize the work, the results achieved in the paper and answer the research questions. The report is then finalized by showing the conclusions of the paper, as well as an outlook regarding future research.

2 Literature Review

Reddit is a valuable source of data and has been the basis for many articles in the existing literature. Among those, there are several from the social sciences. Many reasons make *Reddit* data, as well as the analysis of it, so intriguing. First, due to many *subreddits* that can range from very general topics[2] to very specific ones[3], *Reddit* itself, or rather, the subreddits it consists of are a source of rich datasets available, free to access and open to anyone on the internet. Also, depending on the subreddit, many contemporary topics are discussed, such as current US politics on *r/Politics*. Thus, information made available on *Reddit* goes with the ravages of time.

Also, for data science and data engineering practitioners, *Reddit* provides many opportunities for exploring, analyzing and dealing with big data. The data retrieval via web scraping or via an API and the analysis of those retrieved data are two interesting fields to explore. The retrieved data from *Reddit* provide a vast amount of semi-structured data, which provides the groundwork for analysis from many perspectives. Structured data is found in, e.g., the date/time of the thread or the comment, the number of upvotes and other variables. Unstructured data, such as textual data, title and text of the thread, can be analyzed as well as images, videos or GIFs. Also, with the use of the data, network data for modeling network structures can be created.

There exists a rich literature regarding the analysis of *Reddit* data, *Bitcoin*, and the methods employed in this paper. We present a few selected, highly interesting papers from existing literature. We think each highlights a certain part of the vast potential and exciting use cases of *Reddit* as a source of data.

One paper in which analyses from the perspective of the social sciences are conducted with

[2]For example, *r/pics*.

[3]For example, *r/Trailmeals*.

data from *Reddit* is presented by (De Choudhury, and De 2014). In that paper, the authors investigate the behavior of redditors, i.e., people active on *Reddit*, when they discuss mental health issues. The authors are stunned by how open and detailed redditors talk about their mental health problems, diagnoses, medications or other practicalities related to that topic, given that mental health/illnesses are still tabooed to a certain extent. The fact that *Reddit* is free to use, and even people without an account can see all of the threads, including their content, makes it even more remarkable that redditors share so much detailed information. In their analyses, the authors, therefore, investigate the factors that influence the behavior of redditors. They find that several factors play a crucial role. First, the support of other people; De Choudhury and De indicate that other people on that subreddit are generally kind, constructive and helpful towards redditors that make public a certain part of their mental illness. Second, redditors are free to choose how much information about themselves they reveal. Thus, everything from total anonymity with generic accounts up to full disclosure of their person is possible. This is primarily amplified by the usage of so-called 'throwaway accounts,' i.e., accounts used for only a single action, such as opening a thread. The authors find that throwaway accounts account for 60 percent of the thread activity. Ultimately, these options allow people to post as they can make it as (non-) anonymous as they wish. Further, *Reddit* is a good niche for the exchange of ideas and information. It has low to zero barriers to access. Many people make an effort to comment, and despite the anonymity, one can have the opportunity to exchange with other potentially like-minded people. The existence of subreddits thus acts as a premise for people to meet with a common interest. Lastly, the authors succeed in building a language model. They figure out that what is being said is directly related to the way it is said. The usage of pronouns, for example, changes drastically based on what is talked about.

Another paper in which the analysis takes a 'social science perspective' is (Buntain, and Golbeck 2014). In their paper, the authors approach *Reddit* by plotting the users of several subreddits and their interactions as a network, thereby mimicking social network structures. Nodes in the graph represent users, and interactions between users are shown as edges. The analysis conducted is then based on the structure, topology and informational value captured in that network. Buntain and Goldbeck find out that in the predominant number of cases, users take on an apparent role in their social network. Also, users are typically part of only one community instead of being a member of multiple communities. Thus, they are 'loyal' to their fixed community in which they are rooted.

As an example, members of one instance of such a community are labelled "True Bitcoiners" and are the point of study by (Knittel, and Wash 2019). They use the term to describe redditors genuinely committed to the *r/Bitcoin* subreddit, continuously post there, and share their enthusiasm about *Bitcoin*. Their threads are usually filled with one, or a combination of, recurring themes: (i) Proclaiming their trust and loyalty to *Bitcoin*, (ii) downplaying problems, often also coupled with overly mentioning the advantages of *Bitcoin*, (iii) spreading information on Bitcoin, (iv) making 'advertisement' for Bitcoin. From a social science perspective, these "True Bitcoiners" are engaged in negative integration, isolation and demarcation from other people not sharing their beliefs. Also, their behavior is characterized by a very pronounced, almost religious, conviction regarding their belief in Bitcoin. To them, *Bitcoin* is more than just a currency, an exchange method or an asset but rather something that has a long-lasting positive impact on society by empowering people worldwide. The authors comment that even during times when *Bitcoin* plummeted, they kept on posting and commenting, further spreading positive threads.

A very interesting combination between the analysis of *Reddit* data and the analysis of *Bitcoin*

in itself is employed in (Bukovina, and Marticek 2016). In their working paper, the authors investigate the relationship between the sentiment of threads on the *r/Bitcoin* subreddit and the volatility of Bitcoin. However, the 'volatility' they talk about is not related to the price (index) volatility but related to other *Bitcoin* metrics such as revenue per block transaction. They claim that *Bitcoin* is approaching a 'saturation stage,' meaning that a lot of the initial hype and uncertainty around *Bitcoin* is starting to fade. The market forces of supply and demand will increasingly play the biggest role in determining its price and revenue gained per mining transaction. Thus, it will become more evident over time which role *Bitcoin* plays in the realm of (crypto-)currencies and financial investments. They further conclude that the sentiment of the threads on *r/Bitcoin* only explains a minor part of *Bitcoin* volatility. This finding we find questionable because of the assumed direction of the relationship. We think that it makes more sense to revert this relationship, i.e., going from the *Bitcoin* price to its effect on *Reddit* threads. Of all the forces that influence the price of *Bitcoin*, we think that *r/Bitcoin* is a comparatively small force.

Finally, the article that builds the foundation for retrieving the data for this paper is *"The Pushshift Reddit Dataset"* by (Baumgartner et al. 2020). The article presents *Pushshift*, a dataset of immense size and volume that stores (meta-)data retrieved from *Reddit*. In that article, the exact working and mechanism of the data retrieval are outlined, and it is a very interesting paper for two reasons. First, it is the ease and flexibility when working with *Pushshift* as data is easily retrievable via a free-to-use API. Second, it provides rich documentation and is suitable to be used as background reading. Baumgartner et al. themselves lay out that their API has been used in over a hundred different research papers and enjoys wide popularity. Also, in this paper, we use the API to retrieve data from the *r/Bitcoin* subreddit (c.f. infra).

3 Data Retrieval

The workflow started with the data retrieval since data from two different sources was needed; first, the data regarding the thread activity from the subreddit *r/Bitcoin* and the price development of *Bitcoin* itself. To access the data, two different APIs were made use of. The *Pushshift* API was employed to retrieve the data regarding the subreddit. In doing so, the following variables have been extracted; the id of the thread, its author, the number of comments, the score (the difference between up- and downvotes), the title of the thread and, if applicable, its text.

Due to the nature of the *Pushshift* API, not all of the data could be retrieved at once but had to be retrieved in chunks. First, we broke down the entire period of January 1, 2016, until December 31, 2020, into smaller subsets spanning half a year each. For each of the subsets, the timestamps of the start and end were recorded. A request has then been sent to the *Pushshift* API in which the name of the subreddit and those timestamps as after and before parameters. Within a single request, only 50 results were received back[4], which have been stored separately. Then after a short waiting period needed to not overuse the API, a new request has been sent, but this time the *after* parameter has been replaced by the latest timestamp of the posts in the previous request. This has been repeated until all of the timestamps in one period have been exhausted. This procedure has been done for all the subsets available, and finally, all of the data

[4] An open issue persists on *GitHub*, c.f. https://github.com/pushshift/api/issues/60. Although the documentation says that maximum 500 data points can be retrieved with a single request, in reality, only 100 can be received. In this application, as we were gathering a lot of data, we decided to limit it even further, reducing the size and spreading out the requests with a lag of 2 seconds.

received via the *Pushshift* API could be concatenated to one single file. It was very beneficial that the data came back in an immaculate and formatted shape.

While the data retrieval process has been straightforward up to here, two major problems come along. First, the API can only provide a snapshot of the data, the reason being the dynamic nature of the dataset itself. While some variables regarding a thread stay the same, such as id, or timestamp, many other variables are time-dependent and can or will change over time. The author account can get deleted, entire threads can be deleted, new up- and down-votes and the number of comments can change. Thus, the dataset is only seen as a 'snapshot.' Also, when sending a request, the API will send the last recorded entries and not scrape the entire subreddit again. Thus, there are two sources of time-varying dependencies. Naturally, however, the further a thread is in the past, the more likely it is that less new activity will be processed and that the API recorded everything correctly. For recent threads, this is not the case, so special caution is due.

Another problematic issue is that for the `score` variable, the difference between up- and downvotes, that variable is not continuously updated and, in some cases, stays at `1` for a very long time. The creators of the API themselves mention in the paper documenting the API (c.f. (Baumgartner et al. 2020)) that *Reddit* itself distorts the `score` variable to prevent spam bots from abusing their platform[5]. As a result, the `score` variable is not very reliable for data related to threads not several years old. All in all, however, the dataset is of good quality. There are some pitfalls to pay special attention to, but a lot of the data is still relevant and does not change over time, especially on an aggregated level.

The second source used was the *Coindesk* API, a very reliable source, free to use[6] that allows users to retrieve the price of *Bitcoin*. Regarding the granularity, the method employed in this paper was to get one price a day, labeled the *Close* price, although *Bitcoin* does not have a close price in the original sense. Nevertheless, it is very beneficial for the use case that the historical prices will not change over time.

4 Part I: Analyzing the Relationship Between Reddit Data and the Bitcon Price

4.1 Methodology

After the data have been retrieved, cleaned and set up, the following is done. We use several different analysis methods to get an understanding of the data. We begin with an exploratory data analysis to explore the dataset at hand. Mainly, we use several visualization and plots to explore the data. We also investigate the correlation of several variables in the data set. Following, we employ linear regression and present several models to decompose and model the number of threads using several variables in the data set. Following, we also decompose the number of comments by use of the number of threads.

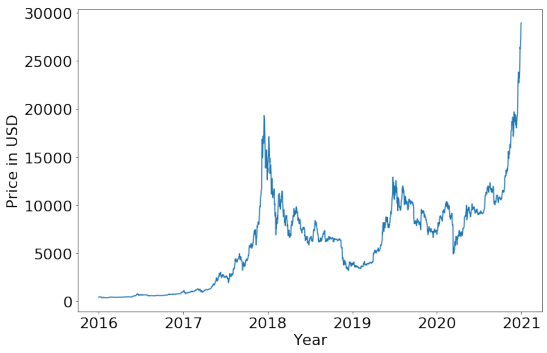
[5]Despite this warning, there exists another open issue, as documented on *GitHub*, c.f. https://github.com/pushshift/api/issues/14, as it is not clear yet how legitimate programmers are still unable to retrieve the valid score.

[6]They only ask the user not to abuse the service and to acknowledge their service by including a statement saying *"Powered by Coindesk"* which we happily take care of.

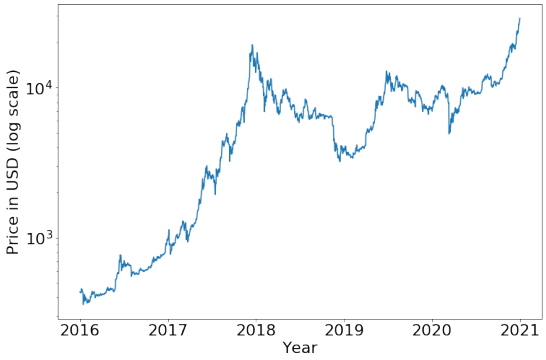
4.2 Exploratory Data Analysis

4.2.1 Bitcoin Price

We start with a descriptive analysis in which we present some of the most critical insights that can be derived from an exploratory standpoint. We begin by presenting the development of the price of Bitcoin, as seen in Figure 1. Two different scales are used, the 'linear axis' in Figure 1a and the 'logarithmic axis' in Figure 1b. The reason to do so is that the linear axis can be misleading as initial growth at low levels looks seemingly irrelevant. The logarithmic scale can, therefore, better reflect the magnitude of the relative changes.



(a) *Bitcoin* Price Index



(b) *Bitcoin* Price Index (Log Scale)

Figure 1: Development of *Bitcoin*Price Index

A few noticeable things can be observed in the plots. We see that in the years of 2016 and most of 2017, *Bitcoin* has been growing quite steadily with a few minor setbacks (best observable on the log scale). Then, towards the end of 2017 and approaching 2018, the *Bitcoin* price spiked, only to be followed by a rapid and long-running decrease in prices until 2019. In the middle of 2019, a relatively short and steady increase in the price was observed but followed by a setback until 2020. However, starting in the second quarter of 2020, a long and sustained increase in the price could be observed, characterized by more than exponential growth. Keeping this general pattern in mind will be of high importance later.

4.2.2 The Dataset

The dataset as described in the methodology section that we are working with has a dimensionality of 574623 observations and 7 columns. We begin by looking at the activity on the subreddit itself, i.e., the number of *new threads per day*, c.f. Figure 2. Several key characteristics can be seen here. First, from the beginning of 2016 until the end of 2017, there seems to be an almost perfect correlation between thread activity and the *Bitcoin* price as the two curves seem to be of the same shape. After the

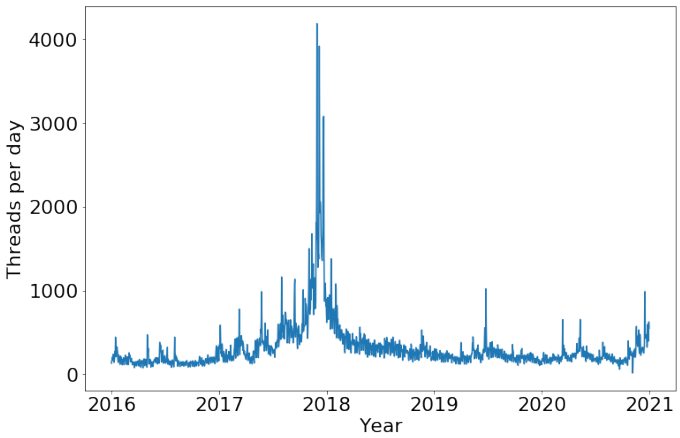


Figure 2: Thread Activity

crash of the *Bitcoin* price, the thread activity went down immediately. Also, it stayed at relatively low levels throughout the years of 2018, 2019, and 2020, and only minor increases can be observed. For example, the short spike in *Bitcoin* price in the mid of 2019 led only to a temporary increase in thread activity. Also, towards the end of 2020, we see that the thread activity picked up speed again. However, it is not as pronounced and we expect the actual number of the new threads in that time range to be higher (c.f. Methodology section).

The analysis so far leads to a hypothesis/explanation as follows. The number of threads can be decomposed into two distinctive elements. First, there is a baseline of threads issued by a steady community. We see that no matter how the *Bitcoin* price behaved, there was always a certain baseline of threads guaranteed. This can be thought of as the 'core' of people that follow Bitcoin, primarily because they are firm believers, interested in the technology, or whatever. This would be in line with the so-called "True Bitcoiners" as described by (Knittel, and Wash 2019). The second part can be regarded as a 'momentum activity.' If the price of *Bitcoin* rises or behaves dramatically, then it is, of course, evident that more and more of the core authors will also increase their number of threads, partly because there is more to report on. But also, it will call new actors to the stage, retail investors jumping in on the action, people becoming generally interested in *Bitcoin*, journalists, even trolls, to name a few.

A necessary, but not a sufficient condition that would corroborate this point is a highly uneven author structure, i.e., a lot of authors that give one or two threads, and a selected number of authors that is low(-er) but produces the bulk of all the threads in total. The study of the authors' distribution is, therefore, the element of the following section. In total, there are 170,526 distinct authors; a total of 46,343 threads (8.06% of the total threads) came from people whose accounts were deleted later on. Although one may be inclined to say that it seems likely that people got interest in what's happening with *Bitcoin*, posted something but then lost interest and deleted their accounts after a decrease, this reasoning lacks support. The reasons why an account is deleted are manifold, ranging from voluntarily deleting the account to not obeying the terms and conditions and being deleted as a result. Also, one knows nothing about the distribution of authors within the set of deleted threads. Hypothetically, it could even be the case that they all came from a handful of big authors.

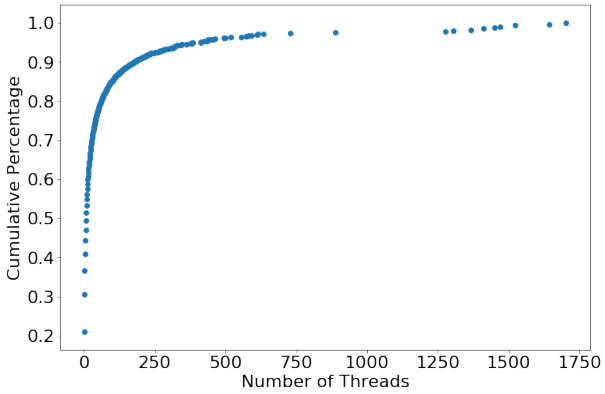


Figure 3: Cumulative Percentage of Threads

Thus, we instead present the distribution of authors and the threads they produce. Figure 3 to the left is used for that undertaking. On the x-axis, the total number of threads is plotted, and on the y-axis, the cumulative percentage of threads in total. The way to interpret is that a point shows that authors that produce up to that level of threads are responsible for the given percentage of threads. It can be seen that the threads from authors that write only one thread already accounts for slightly more than 20% of threads in total. Authors that write one or two threads account for a little bit more than 30% of the threads in total; eight is 50%. The top 10 authors only provide 2.5% of all the threads. Following, we investigate if a classical *Pareto law* of the 80-20 nature is present, which in this case would translate into "20% of all authors are responsible for 80% of the threads in total (and vice versa)." Such distribution cannot be seen directly, but we find that 80% of all authors are responsible for 30.5% of the contents. Vice versa, this means that 20%

of all the authors write 69.5% of all the threads, which is somewhat deviating from the classical 80-20 ratio, but impressive nonetheless.

Initially, one may think that this contrasts with the figure presented, given that the second point is already at the 30.5% level. However, that is not the case. Instead, it shows the severe imbalance in the dataset. The authors that only write one or two threads represent 80% of the total number of authors. Also, 95% of the authors are responsible for 50% of the threads in total, which makes 50% of the threads coming from only 5% of the total authors.

We present Figure 4, which over time shows if an author from one of the three categories (1, 2 or 3 Threads created) created a thread, when did that happen. The term can thus be interpreted as a count argument. It is observable that the bulk of the threads coming from people that are one- or two- or three-time authors of threads post in the time where the *Bitcoin* price is spiking, especially pronounced for 2018. Also, we see certain smaller spikes in mid-2019 and towards the end of 2020, coinciding with periods where the *Bitcoin* price was also going up a lot. Additionally, we present Figure 5 to show when threads from deleted accounts were issued. There, we see an obvious pattern. The new threads are almost perfectly correlated with the spike of *Bitcoin* towards 2018. However, after that, the activity by people who later deleted their account dropped to zero, with a few occasional and rare spikes in later years. This further gives rise to the explanation that many people just left the platform altogether after the price plummeted drastically in 2018.

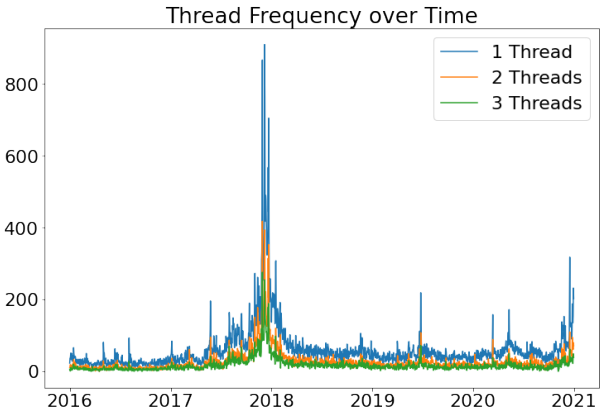


Figure 4: Threads by authors that post 1,2, or 3 threads in total

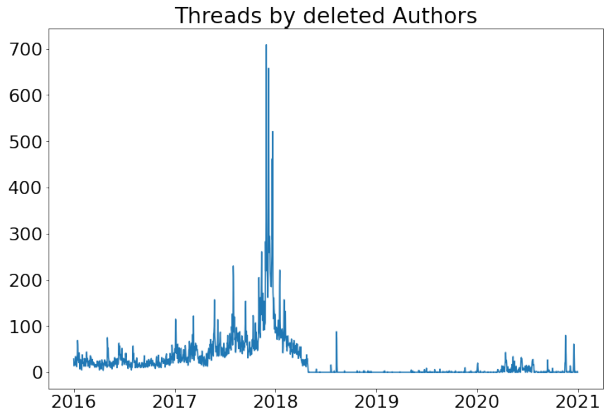


Figure 5: Threads by deleted accounts

So, in all likelihood, they were not interested in the matter in general but wanted to get in on the action, nothing more. Yet another possibility is that these threads came from bots that tried to scam while markets were at an all-time high and were therefore deleted. In addition, we present an analysis of when the authors with the most threads were active, presented in Figure 6 below:

It is pretty interesting to see that also the threads issued by the most active users were very time-dependent and that they have not been publishing threads continuously but rather post a lot within a short time span and then drop altogether (excluding *Daveliuz* here). This could indicate that they tried very hard for a short period to get a good standing in the subreddit but couldn't establish it. Shortly after, the entire interest dropped, and they stopped publishing. Thus, the presented authors here cannot be attributed to the "True Bitcoiners." Therefore, we gauge that the community of these "True Bitcoiners" post a lot as a group, but each individual does not stand out due to their high number of threads posted.

Overall, we now have established the proof that a lot of the additional thread activity comes from people who are just interested during the spikes of *Bitcoin* and are jumping in on the action and do not have a long-term interest in the topic itself. However, it is interesting to see that the spike in 2017 is so much more pronounced than, e.g., in 2019 and the end of 2020. A reasonable explanation would be that people at the end of 2017 were so exuberant, which led to devastation afterward.

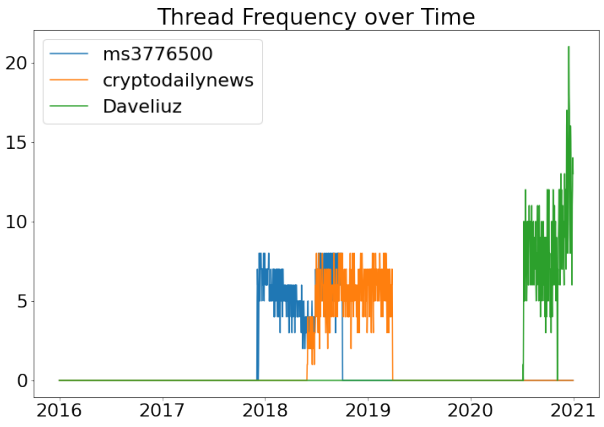


Figure 6: Thread activity by three most active authors

We have established an evident disproportionate distribution of the weights, with many threads coming from a small number of selected authors. The remaining authors, rather, the bulk of the authors, posts up to 8 times, but rarely more often than that. Also, a lot of the one-time posting occurs in combination with rising *Bitcoin* prices. We could see a clear correlation there. In general, higher *Bitcoin* prices lead to more activity on that platform in terms of new threads. Moreover, it is fascinating to see this 'democratic' forum, as everyone can join, post, and there is no starting capital or the like required. The access is thus, more or less, guaranteed to everyone interested, but only the ones genuinely interested will stay on that platform. Next to the attention received from rising *Bitcoin* prices, trust is also a big issue. This was remarkable after the prominent drop in prices at the beginning of 2018. Increases in *Bitcoin* after that did not lead immediately to an increase in threads, but it took a lot of time for authors to pick up again and start commenting. Probably, a lot of trust was lost in the process.

We have also seen a massive imbalance regarding the number of comments per thread. However, it lies in line with intuition. One expects there to be a lot of threads not having a lot of comments. Also, take the dynamics into account. Usually, the threads on the subreddit are sorted based on one of two metrics. Option 1 is sorted by the `score`, where a 'winner takes it all' dynamic sets in. The threads on top will be seen by most people, which guarantees a lot of interaction and comments. Option 2 is that the threads are sorted by the time they were created (newest first). As a result, there exists only a brief period when a thread is in the best position to be seen by a lot of people. After that period, it will fade away somewhere and will probably not be seen by a lot of people.

4.3 Extended Price Analysis

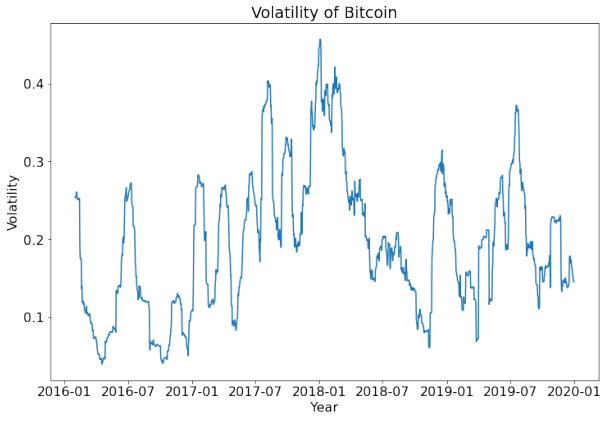


Figure 7: Volatility of Bitcoin

We now extend the price analysis and take several other variables into account as well. As the price development has already been shown in Figure 1, we now present the volatility of the price development, presented in Figure 7. It is interesting to see that the volatility itself is also swinging wildly. This indicates that the price can swing a lot, and even the metric capturing the price swings can swing itself. Based on Figure 7, one can see that the volatility is running in cycles. Usually, there

exist short periods in which volatility is pronounced low or high, and a clear 'build-up' or 'going-down' connecting the two phases is observable. The only time where a long and sustained decrease in *Bitcoin* volatility is noticeable is from 2018 until 2019, the year where *Bitcoin* started close to its record high but only plummeted afterward. The results so far, therefore, indicate the riskiness and instability that came along with an investment in Bitcoin.

For further analysis, we study the correlation between the following variables: (i) the *Bitcoin* price, (ii) the change in *Bitcoin* price, (iii) the relative change in *Bitcoin* price, (iv) the *Bitcoin* volatility, (v) the new threads per day, (vi) the change in new threads per day, (vii) the percentage change in threads per day, (viii) the new comments per day, (ix) the change in comments per day, and lastly, (x) the percentage change in comments per day. Figure 8 shows the correlation heatmap.

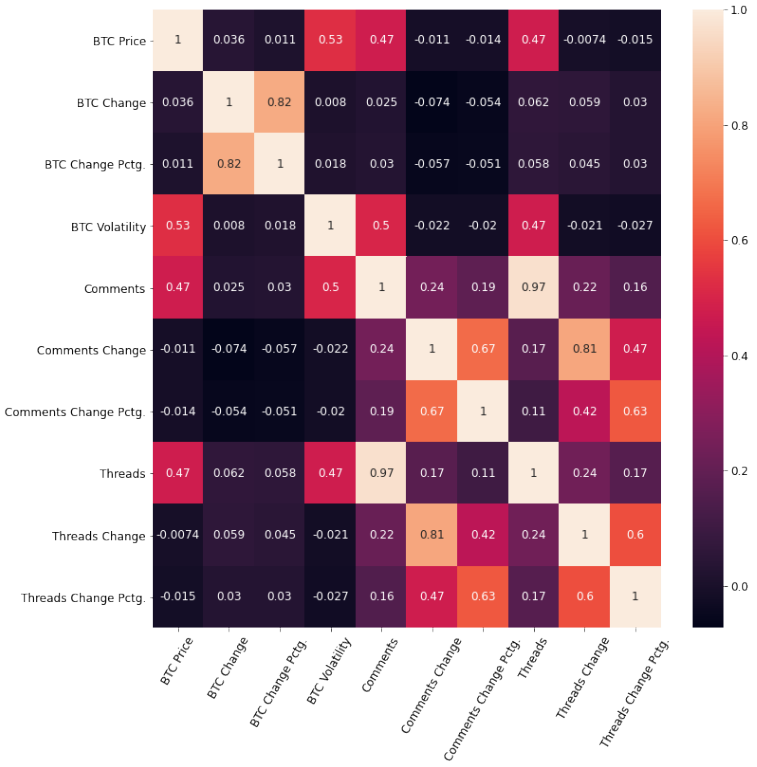


Figure 8: Correlation Heatmap

It is interesting to see that the *Bitcoin* price correlates positively with the volatility. This goes against expectations as there is usually a negative correlation between price and volatility, the so-called leverage effect (Ladokhin 2009). This is very interesting to see, mainly because the correlation is quite substantial. This indicates that the higher the price, the bigger the swings will be (in relative terms). However, what is expected, is that the price of *Bitcoin* is positively correlated with the number of threads and the number of comments. This is in line with expectation as we would expect that higher *Bitcoin* prices lead to more enthusiasm. Also, a positive correlation between the volatility and the number of comments and threads is observable. This is expected due to two reasons. 1) we know that volatility and price are correlated, so are price and comments/threads. As a result, there must be some correlation between the volatility and comments as well. 2) If the market is in turmoil, and a lot of up- and downswings of high magnitude occur, then there will be more people commenting, we believe. Next to that, a very high correlation between the number of comments related to threads of a certain day and those very threads is observable, which is interesting. Also, this positive correlation (although to a lesser extent) exists when one looks at the absolute values and at the relative change.

4.4 Model Building

We now focus on investigating if the number of threads on a day, NT_t, can be decomposed and accordingly predicted by using several key variables. We start by presenting four initial models that use the following variables; the *Bitcoin* Price Index, *BPI*, and three different time components, i.e., the time linear, squared and cubed. The linear regression models are presented

in the following:

$$NT_t = \beta_0 + \beta_1 \times BPI_t + \varepsilon_t \tag{1}$$
$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \varepsilon_t \tag{2}$$
$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \varepsilon_t \tag{3}$$
$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \beta_4 \times T_t^3 + \varepsilon_t \tag{4}$$

where β_0, the intercept, reflects the baseline interest from the 'incumbents,' β_1 captures the effect of the price of Bitcoin, and $\beta_2, \beta_3, \beta_4$ capture the linear, squared, cubed time trends. The linear regression models are trained on only the data from 2016 until 2019 since the data from 2020 may not be sufficiently updated. The models with the fitted parameters can be found in Table 1. Coefficients that were statistically significant at a significance level of $\alpha = 0.05$ are highlighted in bold.

	Model Formulation	AIC	R^2
1	$NT_t = \mathbf{141.796} + \mathbf{0.039} \times BPI_t + \varepsilon_t$	20257	0.2215
2	$NT_t = \mathbf{323.830} + \mathbf{0.075} \times BPI_t - \mathbf{0.482} \times T_t + \varepsilon_t$	19861	0.4105
3	$NT_t = \mathbf{92.218} + \mathbf{0.065} \times BPI_t + \mathbf{0.426} \times T_t - \mathbf{0.001} \times T_t^2 + \varepsilon_t$	19677	0.4823
4	$NT_t = 14.1 + \mathbf{0.066} \times BPI_t + \mathbf{1.259} \times T_t - \mathbf{0.002} \times T_t^2$ $+ \mathbf{5.79x10^{-7}} \times T_t^3 + \varepsilon_t$	19652	0.4920

Table 1: Model Summary

Interestingly, the intercept in Equation 4 is no longer statistically significant. One may think that this refutes the idea of a 'baseline' of incumbents who always comment. However, using the intercept as a proxy for baseline activity is not a proper choice. Even if there is a core of people always active on the platform, it is by no means clear that that core will also always give the same output. On top, the baseline community can grow, and their activity will also depend on the price of bitcoin. Then, throughout all models, it has been seen that there is a significant effect of the price of *Bitcoin* on thread activity that stabilizes around 0.065 to 0.066. This means that for every increase of 16 USD, the number of new threads per day rises by 1. Also, a time effect of cubic nature could be established. That is, first the number of comments goes up, then it goes down only to pick up speed again later (albeit at a much slower pace), which is somewhat in line with intuition and would explain the effect of people euphoric first until 2018, then lose interest in the trust in 2018 but slowly start rebuilding interest in 2019. Also, one can see that adding the time effects led to a continuous decrease in AIC and a continuous increase in R^2, indicating a beneficial model quality. With model 4, a little less than 50% of the total variation in thread activity can be explained by the factors used. Although there remains a lot of unexplained variation, this model is a reasonable first approximation. Other (latent) factors that could account for unexplained variation are, for example, the general popularity of *Reddit* and the subreddit, news coverage, or the like.

We extend the initial model(s) by additionally incorporating the volatility. The new model presented now looks as:

$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \beta_4 \times T_t^3 + \beta_4 \times \sigma_t + \varepsilon_t \tag{5}$$

and accounts for the bitcoin's volatility, here depicted as σ_t. After fitting, the model description is as follows:

	Model Formulation	AIC	R^2
5	$NT_t = -36.318 + \mathbf{0.061} \times BPI_t + \mathbf{1.124} \times T_t - \mathbf{1.72x10^{-3}} \times T_t^2 + \mathbf{5.27x10^{-7}} \times T_t^3 + \mathbf{327.6} \times \sigma_t + \varepsilon_t$	19640	0.4971

Table 2: Model Summary (extd.)

So, we see that we could further increase the R^2-value. However, as it is clear that the metric will not decrease upon adding variables, it is not that convincing yet. Still, we also observe that the AIC decreased, indicating that the increase in model performance is worth the increased model complexity. On top, although the parameter values change, the significant cubic time effect and price of *Bitcoin*, as well as a non-significant intercept, stayed the same. As the last step, we extend the models by using interaction effects, i.e., the interaction between the *Bitcoin* price and the time variables.

$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \beta_4 \times T_t^3 + \beta_4 \times \sigma_t + \beta_5 \times (BPI_t \times T_t) + \varepsilon_t \tag{6}$$

$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \beta_4 \times T_t^3 + \beta_4 \times \sigma_t + \beta_5 \times (BPI_t \times T_t) + \beta_6 \times (BPI_t \times T_t^2) + \varepsilon_t \tag{7}$$

$$NT_t = \beta_0 + \beta_1 \times BPI_t + \beta_2 \times T_t + \beta_3 \times T_t^2 + \beta_4 \times T_t^3 + \beta_4 \times \sigma_t + \beta_5 \times (BPI_t \times T_t) + \beta_6 \times (BPI_t \times T_t^2) + \beta_7 \times (BPI_t \times T_t^3) + \varepsilon_t \tag{8}$$

After fitting the models, we get the following:

	Model Formulation	AIC	R^2
6	$NT_t = -11.827 + \mathbf{0.207} \times BPI_t + \mathbf{0.739} \times T_t - \mathbf{0.002} \times T_t^2 + \mathbf{10^{-6}} \times T_t^3 + \mathbf{159.807} \times \sigma_t - \mathbf{1.73x10^{-4}} \times (BPI_t \times T_t) + \varepsilon_t$	19376	0.583
7	$NT_t = 16.348 + \mathbf{0.891} \times BPI_t - \mathbf{1.719} \times T_t + \mathbf{4.587x10^{-4}} \times T_t^2 - \mathbf{2.5x10^{-6}} \times T_t^3 - \mathbf{310.258} \times \sigma_t - \mathbf{1.69x10^{-3}} \times (BPI_t \times T_t) + \mathbf{7.9x10^{-7}} \times (BPI_t \times T_t^2) + \varepsilon_t$	19135	0.648
8	$NT_t = \mathbf{0.03} + \mathbf{1.511} \times BPI_t - 4.749 \times T_t + \mathbf{2.349x10^{-3}} \times T_t^2 - \mathbf{1.298x10{-}6} \times T_t^3 - \mathbf{347.244x10^{-2}} \times \sigma_t - \mathbf{3.474x10^{-3}} \times (BPI_t \times T_t) + \mathbf{2.96x10^{-6}} \times (BPI_t \times T_t^2) - \mathbf{7.7x10^{-10}} \times (BPI_t \times T_t^3) + \varepsilon_t$	19069	0.664

Table 3: Model Summary (further extd.)

We see better model quality metrics upon adding the variables. In detail, we see that the AIC went down continuously. This means that in each step, the model performance seems to be justifying the additional model complexity. Also, the R^2 went up to 0.664 for the final model, which means that 66.4% of the entire variation in the dependent variable, NT_t can be explained by all of the variations in the independent variables. However, a problem observed in the models before now becomes increasingly evident, especially in the step from 7 to 8. The R^2 and AIC behave favorably upon augmentation with other variables, but the parameter values change drastically. Also, this problem is amplified by the fact that the new model's parameters are barely interpretable anymore. Thus, even though model quality improved, it did so at the expense of lack of explainability. On that note, one could claim that it is better to use other methods instead of Linear Regression.

Lastly, one can hypothesize that the number of comments can be further included. However, there is a problem. Rather, the number of new threads is responsible for comments and not the other way around. Thus, we present a new regression model that models the number of comments as a function of the number of threads, presented here:

$$Comments_t = \beta_0 + \beta_1 \times NT_t + \varepsilon_t \tag{9}$$

	Model Formulation	AIC	R^2
9	$Comments_t = \mathbf{100.152} + \mathbf{10.1375} \times Threads_t + \varepsilon_t$	23371	0.937

Table 4: Model Summary (new)

This model looks pretty good, as both parameters are highly significant, the R^2 is also very high, 93.7% of the variation in comments is explainable by the variation in the number of threads. In Figure 9, we plot the exact number of comments and the number of comments predicted by this model. In general, the predictions come very close to the actual observed number of comments. Granted, a few discrepancies occasionally occur, for example, in the fall of 2016, as well as the winter of 2018. The root mean squared error is reported as 855.35, which is relatively high, possibly due to the squaring effect, which is worsened because of the mispredictions. The MAE, in comparison, is a bit lower, reported at 539.34. So, overall it seems to be quite good, but a few spikes in the number of comments cannot be predicted. However, that seems to be inherent to the nature of the problem.

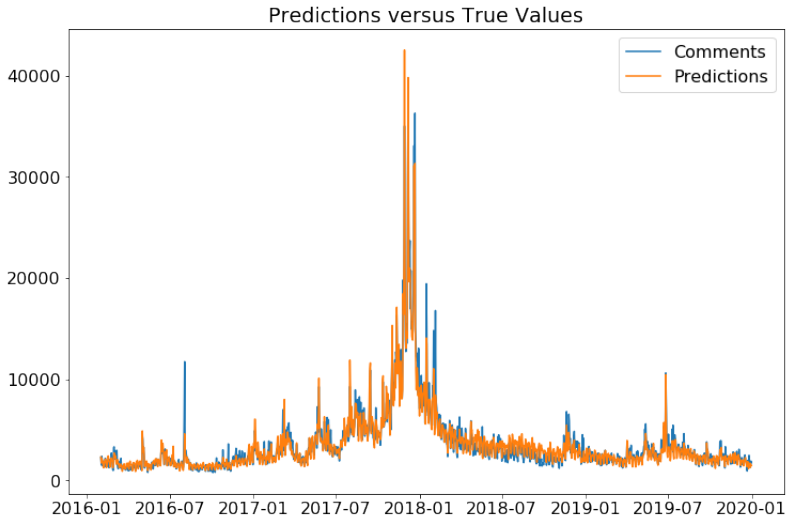


Figure 9: Predictions

5 Part II: Analyzing the Contents of Threads

5.1 Methodology

5.1.1 Data Pre-Processing

The first action in the data pre-processing is the cleaning process. Cleaning is performed in several steps. *Step 1*: We first remove all documents where the text is missing, then set all words

to lowercase. The resulting text is input to a function that computes features. *Step 2*: After the feature engineering step (see next section), we further clean the text by removing punctuation and trailing whitespaces, removing English language stopwords and identifying and removing bitcoin-related stopwords. This is input for a sentiment computation function. *Step 3*: The final cleaning step is to stem the words and tokenize them before performing topic modeling. Stemming is not strictly necessary, but it can improve the predictive ability of features extracted from topic models. Tokenization transforms a document into a list of its "tokens", which can be words or n-grams. This representation becomes the input to a topic model. Figure 10 gives an example of the cleaning process on a short text.

```
Original Text:
    "You're a grand old flag, You're a high-flying flag,
    And forever in peace may you wave"

Step 1: Feature Engineering
    'you're a grand old flag your a high-flying flag,
    and forever in peace may you wave'

Step 2: Sentiment Computation:
    'your a grand old flag your a highfli
    flag and forev in peac may you wave'

Tokenized Text: Topic Modeling
    ['grand', 'old', 'flag', 'highfli',
    'flag', 'forev', 'peac', 'wave']
```

Figure 10: Text cleaning. Step 1: lowercase, input for Feature Engineering. Step 2: remove punctuation and stem, input for Sentiment computation. The final output is a set of tokens, where stopwords are removed; this is input for Topic Modeling.

The next act is *Feature Engineering*, i.e., defining features based on the present variables, which will be of importance later on. In the present case, we define features based on the text of the threads that summarize or shed light on a certain part of the text. Unlike articles or manuals, *Reddit* threads have no rules defining their structure and only a few guidance on content [7]. Generally, one posts to *r/Bitcoin* to either ask a question or share information related to the topic *Bitcoin*. Thus, our strategy is to obtain as detailed features as possible to determine a thread's structure. Using python library `py-readability`, we obtain for each document 21 features; a detailed list can be found in Appendix B. The features include characters, syllables, words, and part-of-speech counts. Additionally, we compute the readability of the thread, which measures the level of difficulty needed to read a text. Several measures of readability exist; we selected the Dale Chall readability metric due to its simplicity and lack of assumptions on the text. It is defined as

$$\text{Dale Chall Readability } = 0.1579\left(\frac{\text{Difficult Words}}{\text{Words}} * 100\right) + 0.0496\left(\frac{\text{Words}}{\text{Sentences}}\right)$$

If the percentage of difficult words is greater than 5%, then the score is adjusted by adding 3.6365. The adjusted score represents the "Reading Grade of a reader who can comprehend

[7] For more/detailed information, c.f. `https://www.redditinc.com/policies/content-policy`)

your text at 4th grade or above" (Readability Formulas n.d.). The exact score breakdown can be found in the chart below:

ADJUSTED SCORE	GRADE LEVEL
4.9 and Below	Grade 4 and Below
5.0 to 5.9	Grades 5 - 6
6.0 to 6.9	Grades 7 - 8
7.0 to 7.9	Grades 9 - 10
8.0 to 8.9	Grades 11 - 12
9.0 to 9.9	Grades 13 - 15 (College)
10 and Above	Grades 16 and Above (College Graduate)

Figure 11: Dale-Chall Readability Score Interpretation (Readability Formulas n.d.)

A score of 4.9 and below means that a text can be understood by someone in Grade 4 and below. A score of 10 and above means that a text can be understood by someone with a university education. The `py-readability` implementation assigns a score of 0 to any text with fewer than 30 words. Later on, we determine that this does not cause an issue in the analysis.

As the last step, we perform *Sentiment Computation*; that is, we calculate the sentiment of the text offered in the thread. This will be seen as another feature. An essential dimension of social media posts such as *Reddit* threads is their sentiment or measure of the writer's tone. We compute sentiment using VADER (Valence Aware Dictionary and sEntiment Reasoner), an open-source tool that looks for a set of lexical features in a text with rules of stylistic conventions for expressing sentiment intensity. VADER is "specifically attuned to sentiments expressed in social media" (Hutto, and Gilbert 2015) [8]. We perform sentiment analysis after cleaning and stemming the documents. The sentence's sentiment in Figure 10 is 0.7579, which is very positive - quite in line with intuition.

5.1.2 Topic Modeling

A simple and common method for text analysis is *tf-idf* representation, where document information is described by weighted word frequencies. This information can be used as features for unsupervised learning of the corpus or to augment the predictive models of *Bitcoin* volatility. However, this representation leaves out important information about the corpus: inter- and intra-document relationships. We, therefore, turn to topic modeling, which generates a probabilistic representation of a corpus. We chose *latent Dirichlet Allocation* (LDA) (Blei et al. 2003), a three-level Bayesian hierarchical model that models each document in a corpus as a mixture over a set of topics. LDA represents a corpus as a set of normalized topic weights. An advantage of LDA over other topic models is that the number of parameters does not grow with the training corpus size; this avoids overfitting. A disadvantage is that the number of topics must be selected *a priori*. While there exist hierarchical topic models that alleviate this issue by constructing a hierarchy of levels of topic abstraction, they can be tricky to tune (Koltcov et al. 2021) and implement.

[8] The corresponding GitHub repository and documentation can be found under `https://github.com/cjhutto/vaderSentiment`

We fit an LDA topic model to the cleaned, stemmed text for several values of k, the number of topics, and select k with an "elbow rule" common to unsupervised learning: select the k where the rate of decrease in training error (here, it is measured as model perplexity) begins to decline. Additionally, we consider the interpretation of the produced set of topics before selecting a final model.

5.1.3 Visualization

We visualize the data before predictive modeling. First, we plot univariate and multivariate visualizations of some corpus features as part of exploratory analysis. This provides valuable information when building predictive models. Next, we visualize a multi-dimensional representation of the corpus to get an idea of its structure, i.e., how are documents clustered.

We use *t-SNE*, or t-distributed stochastic neighbor embeddings (van der Maaten, and Hinton 2008), to get a two-dimensional visualization of the LDA topic weights. *t-SNE* is a nonlinear dimensionality reduction technique for visualizing large datasets. It models pairwise dissimilarities in the feature space as a joint probability distribution, where distances between points are measured with respect to a Gaussian distribution. It then finds a 2- or 3-dimensional representation of this probability distribution (replacing the Gaussian with a student t-distribution) that preserves local similarities of the feature space. The result is a map that reveals the local and global structure of a corpus. By coloring each represented document by its dominant topic, we can potentially infer the relationships between topics.

5.1.4 Predictive Modeling

It is an interesting exercise to build a model that predicts whether a thread was posted on a day with high or low *Bitcoin* volatility. By definition, 30-day rolling volatility is time-dependent. However, in this exercise, we will make the (painfully) simplifying assumption that whether *Bitcoin* has high or low volatility on a specific day is an independent event.

We use our obtained features and topic weights to build a Logistic Regression model that predicts whether a thread was posted on a day where *Bitcoin* had high volatility. Given the highly correlated nature of the predictors, we perform PCA and select a few Principal Components as input to the Logistic Regression model.

5.2 Thread Features

By their definition, many of the engineered features are highly correlated, and the LDA topic weights are nearly uncorrelated. From Figure 12, we see strong positive correlations between words, syllables, characters, and other lexical features. We see a strong negative correlation between `type_token_ratio` and lexical features. We see weak correlations between LDA topic weights and between `comments`, `sentiment` and `dale_chall`, the Dale-Chall Readability score. Finally, we see a close to zero correlation between *Bitcoin* volatility and all features. This indicates that the features will be poor predictors of high/low volatility.

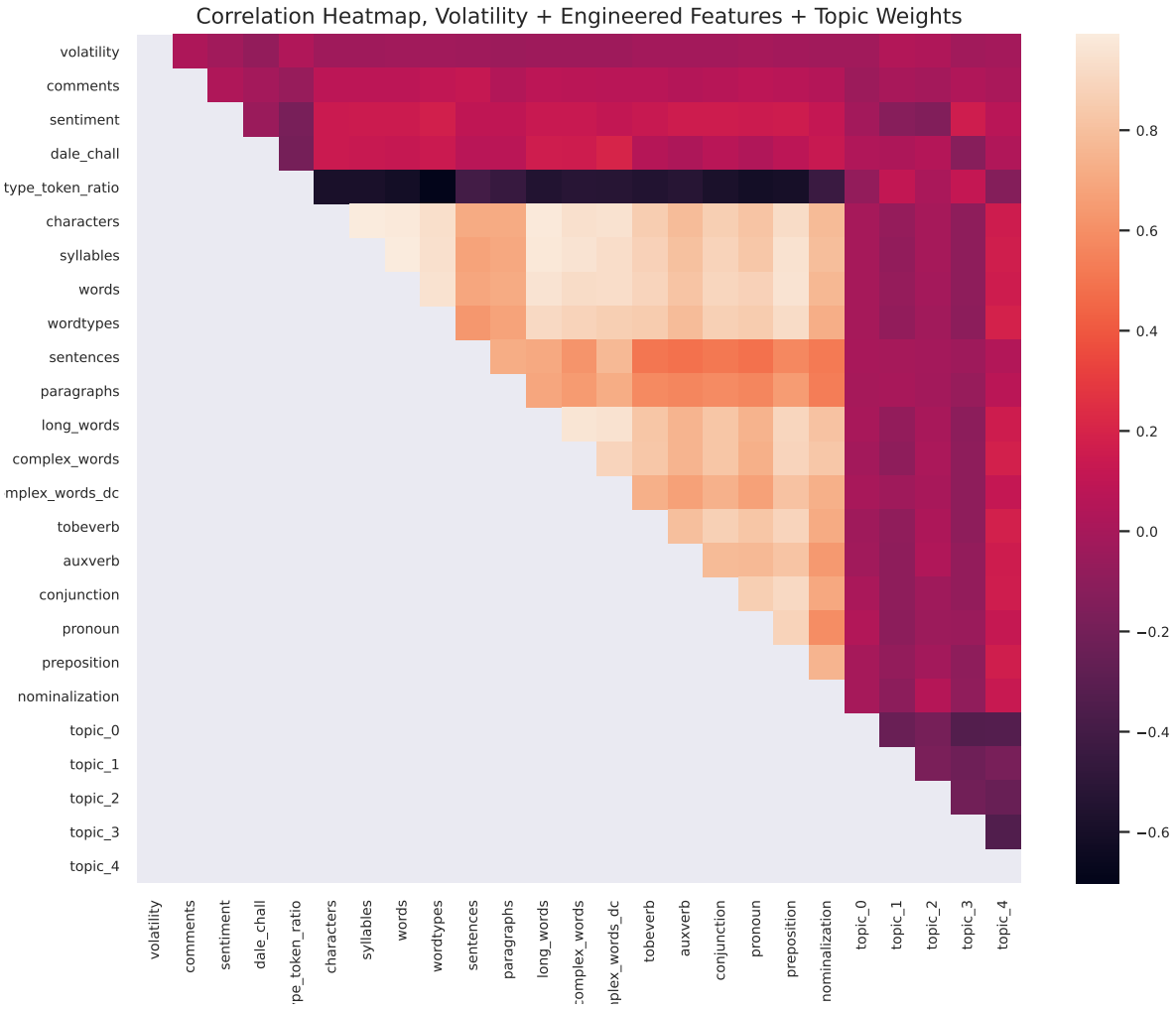


Figure 12: Correlation heatmap of *Bitcoin* volatlity, engineered features, and the topic weights. Note low correlation of features to volatility and high correlation within engineered features.

A very interesting metric is the aforementioned *Dale-Chall Readability Score*. The median Dale-Chall Readability score of the threads is 10.34, which is "Grades 16 and above (College Graduate)" level, c.f. Figure 11. Given the focus on Bitcoin, this is unsurprising. Below, in Figure 13, we provide some examples from the training data of threads with the minimum score, as well as a below- and above-median score.

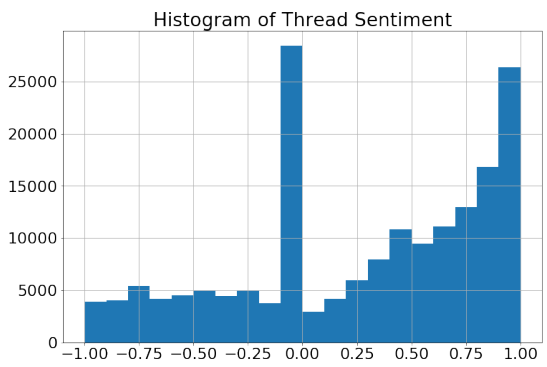
Based on the examples, the score seems in line with intuition. As indicated, a score of 0 is automatically assigned to texts with fewer than 30 words. We found that 1.38% (2462 threads) were given a score of 0. We find that the fraction is small, and therefore will not bias results. Indeed, Figure 12 shows a near zero correlation between `word_count` and `dale_chall`.

```
Minimum (score 0):
------------------
'thanks, sorry for being dumb'

Below Median (score < 10.34):
-----------------------------
'Sent coins to a new wallet on this site, have identifier but password
i though was right is not correct. as a last resort is anyone able to crack
if i give identifier and password i thought i used? .65 btc in acct'

Above Median (score >= 10.34):
------------------------------
'With all the talk that increasing blocksize would reduce the full node
count, why not make positive changes in the opposite direction?
Are any devs working on something that would  *increase* the full node
count?  And if not, why are they willing to  hamstring the network to
avoid a, say, 30% reduction in full node count?'
```

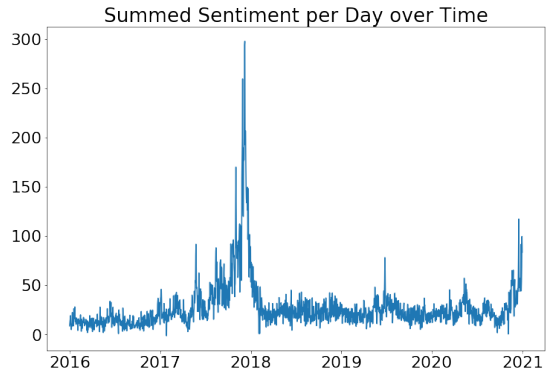
Figure 13: Training data examples of minimum, below-median, above-median and maximum Dale-Chall Readability scores. The first can be understood by someone in "Grade 4 or Below", given the simplistic structure and short length. The second more advanced, but with simple grammar. The above-median example uses difficult words such as "node" and "hamstring".

We present the sentiment of *Reddit* thread texts. After cleaning and stemming the text of the threads for which there are texts available. We employ the *vaderSentiment*, c.f. supra, to get the sentiment score. Based on that, we can make a histogram of the sentiment scores of each thread, presented in Figure 14a. We see that it is predominantly positive; a lot of the threads have a positive score, also a lot above the 0.5 threshold. Further, there are a lot of threads for which there is a slightly negative connotation. However, two things could happen. First, the visualization may be misleading because of the way the binning choice for this histogram. Second, it is not evident if a score mildly below 0, such as -0.05 or -0.1, is enough to be labelled as 'negative.'

As a second visualization, we present a new metric, summed sentiment, in Figure 14b. We compute the sentiment of each thread, group by day and then sum the score per day. Several very interesting things can be observed. First, we see that this metric skyrocketed in the huge *Bitcoin* frenzy at the end of 2017, followed by a rapid decrease. Although at a smaller magnitude, similar rapid increases can be found in mid-2019 and at the end of 2020. It is very interesting to see that no matter the decrease in Bitcoin, the summed sentiment never goes below 0, except for a single day. That was on January 26, 2017, which had nothing to do with the period of decreasing *Bitcoin* prices. Also, it is more important that even during the period where *Bitcoin* fell drastically, the summed sentiment still stayed above 0.



(a) Sentiment Histogram



(b) Summed Sentiment

Figure 14: Sentiment Visualization

Thus, in combination with findings from above, we now explain this phenomenon as follows. If *Bitcoin* rises, then a lot of attraction and attention is brought to the subreddit *r/Bitcoin*. This, in turn, leads to increased thread activity but also increased total positivity on the subreddit. As soon as the price starts to plummet, however, people lose interest and abandon the subreddit altogether instead of remaining on the platform and venting their anger. The people that remain and continue to post are then the ones interested in *Bitcoin* itself, with a substantial share of them probably belonging to the "True Bitcoiners" (c.f. Knittel, and Wash 2019).

5.3 Topic Modeling

We fit LDA topic models with k topics in the set of integers in $[4, 14]$ and measured the perplexity of each model, shown in Figure 15. There are two "elbows" in the plot: at 8 and 10 topics. However, the nominal perplexity values are all close, in the range [191, 196.5]. Since the reduction in perplexity by selecting more topics is relatively small, we opted for an additional criterion in selecting k: interpretability.

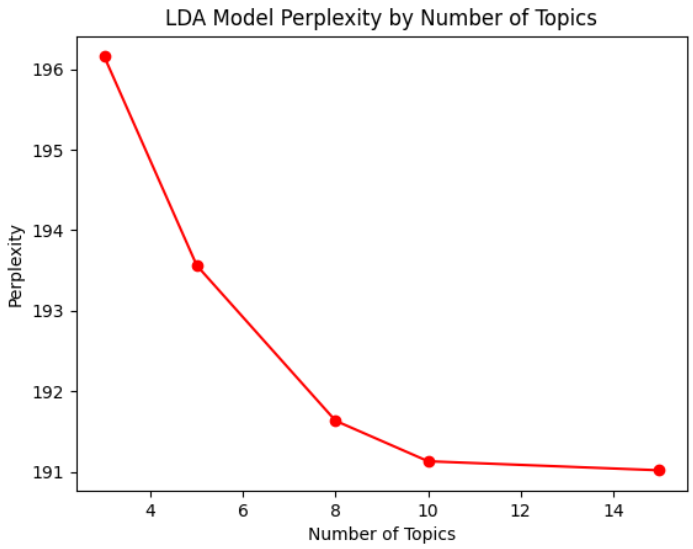


Figure 15: LDA Perplexity vs. k

For each topic model, we computed the top 10 most common words present in each topic (in LDA, topics are modeled as a mixture of words). Figure 16 shows the results for the LDA topic model with 5 topics. We found this model to be most interpretable; thus, we select it as our final model.

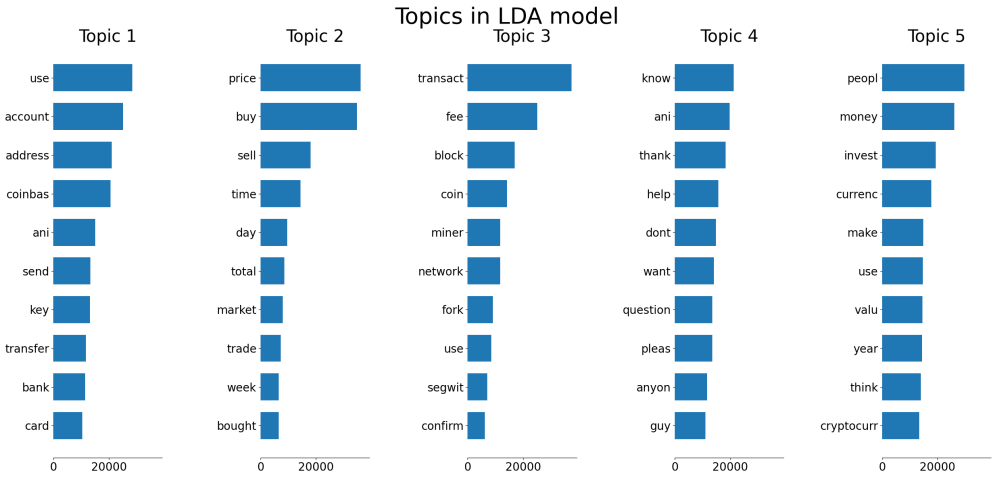


Figure 16: Most Common Words in Topics

Based on this final model, we can clearly distinguish five topics. Topic 1 is related to exchange/payments. It is likely to belong to the threads that deal with the usage of *Bitcoin* for payment and exchange purposes. Topic 2 is related to the 'financial market' side of things, i.e., the aspect of *Bitcoin* as a purely financial asset. It is related to trading and investing and the basic interrelation with the 'financial market' itself. Topic number 3 relates to the *Blockchain*, the underlying technology that *Bitcoin* is based on. Topic 4 is a mite hard to disentangle. However, given that so many words are present that relate to a question context, it is likely that this topic relates to all the threads in which questions are asked. Finally, topic 5 can be regarded as a 'philosophical topic.' This is related to threads where the generalities about cryptocurrencies and *Bitcoin* are discussed on a very high-level basis.

5.4 Visualization of Threads with t-SNE

Figure 17 displays the t-SNE representation of each thread's final LDA model topic weights in the dataset, colored by dominant topic (legend: 1-blue, 2-orange, 3-green, 4-red, 5-purple). Clusters with one single color represent threads with one strongly dominant topic. Clusters with multiple colors are in between. It seems that many clusters are overlapping, indicating that a two-dimensional representation does not suffice to show the entire structure[9]. Nonetheless, it could also suggest that some threads are a combination of two topics. For example, someone might ask a question (Topic 4) but be very specific about a certain topic, such as *Blockchain* (Topic 3). In that case, words from both topics would be present, explaining the overlap. We can infer some local and global structure in the threads. Threads with dominant topics 1, 4 and 5 seem to have a distinct grouping. Threads of topics 2 and 3 are dispersed between other topics.

[9]It is expected, however, that such a rich example of data as textual data is can impossibly be presented sufficiently well on a two-dimensional sphere.

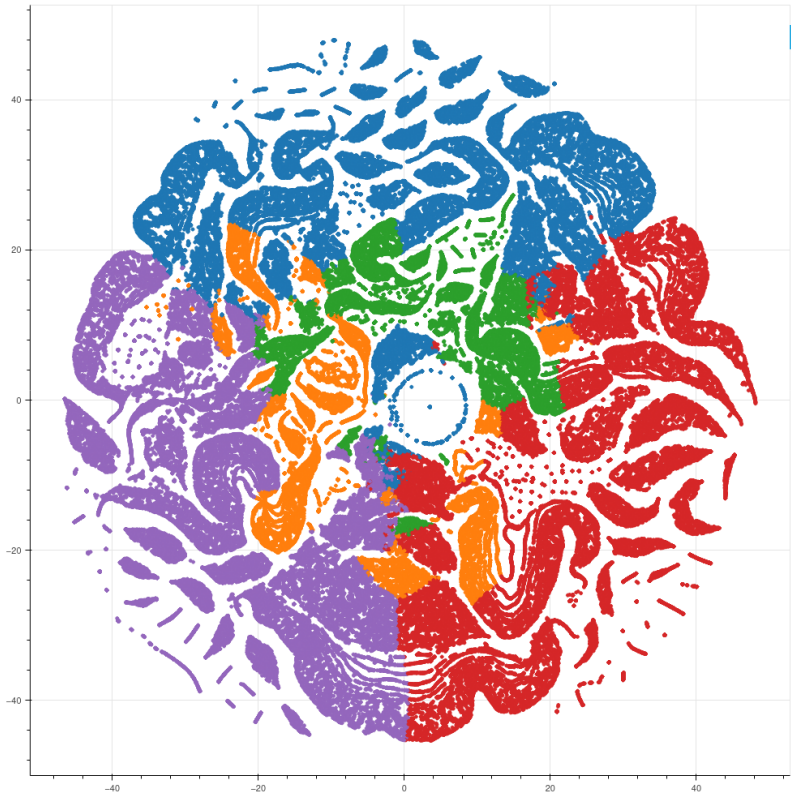


Figure 17: t-SNE of documents colored by dominant LDA topic. Legend: 1-blue, 2-orange, 3-green, 4-red, 5-purple.

To get a better picture, we create another t-SNE view with a random sample of 50,000 threads.

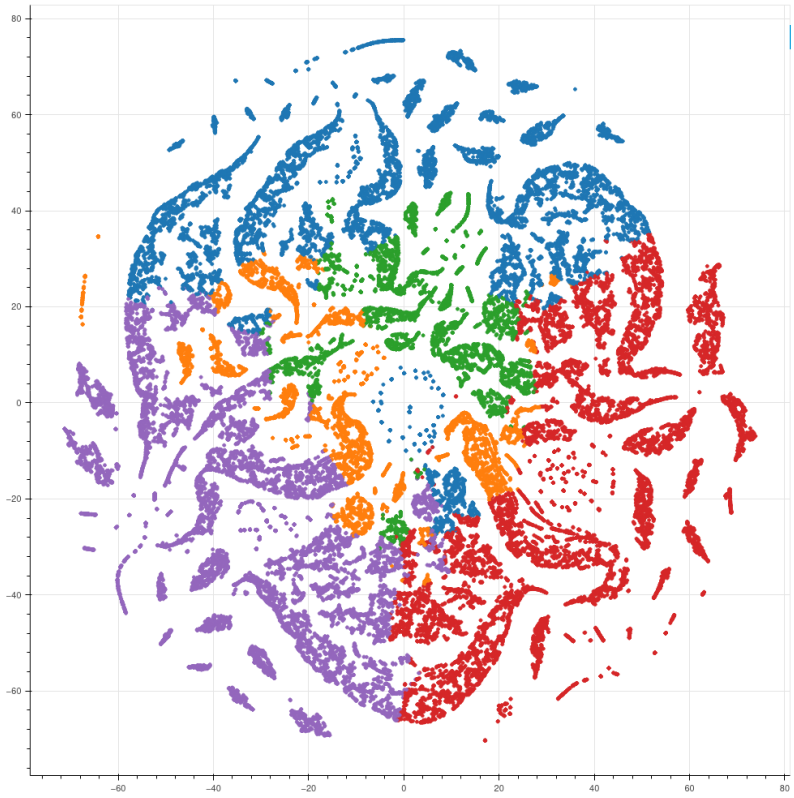


Figure 18: t-SNE of 50,000 randomly-sampled threads colored by dominant LDA topic. Legend: 1-blue, 2-orange, 3-green, 4-red, 5-purple.

Figure 18 reveals a similar structure of overlapping topics in a higher-dimensional space. We see more easily in the low-dimensional representation clusters that instead of distinct sphere-like shapes are very curved and are not one giant cluster for each model only. However, this is understandable since it is highly probable that threads can combine one or more of the topics we modelled. As one example, think of someone asking a question about *Blockchain*, in which they use precise vocabulary but still use the typical words associated with asking questions. Also, when taking domain knowledge of *Reddit* into account, this seems very sensible. Individual threads might ponder on something that has already been said or mentioned in a thread before; they may shed light from a different perspective on it or the link. As a hypothetical example, think of someone who posts something on the energy usage/consumption of *Blockchain* and someone else picking up on that thread but embedding it in a 'philosophical context,' combining the ecological implications with secondary societal impact or the like.

5.5 Predictive Modeling

We build a Logistic Regression model to determine whether a thread was posted on a day when *Bitcoin* volatility was high ($\geq 30\%$). We opt for this model due to its simplicity and interpretability. Despite modifying the model to adjust for issues in the data, we do not yield a satisfactory predictive model that determines whether a thread was posted on a day with high *Bitcoin* volatility.

The response variable of the model is an indicator of high volatility, or `volatility >= 0.3`. The predictors are the 24 features (19 engineered features and 5 features of topic weights). Each row in the training set represents one thread. Each row in the test set indicates whether the thread was posted on a day of high *Bitcoin* volatility. In this model, we assume that whether *Bitcoin* has high volatility is an independently-occurring event; we consider each thread independent. This assumption may not hold up in practice, and we will discuss this later. However, for this point, the presumptions are taken for simplification of the analysis.

When proceeding to the model specifications, several issues had to be taken into account. We found two challenges after exploring the data: imbalanced classes in the response variable and highly correlated predictors. The former problem can be dealt with by under- or over-sampling or by class weighting; we try two formula of class-weighting. The latter problem can be dealt with by dimensionality reduction or by sampling from the predictors. Given the correlation heatmap in Figure 12, we found the best strategy is to use PCA, which changes the coordinates of the features to yield a set of uncorrelated "principal components" or projections of the features onto a new basis.

Figure 19 shows that six Principal Components explains 95% of the variance in the features. At the same time, simply keeping one component explains 70% of the variance. We conclude that the proper number of components to keep lies between 1 and 7. While retaining information is essential in creating a discriminative model, too much information can lead to overfitting. Thus, we experiment with keeping a different number of Principal Components in the Grid Search step.

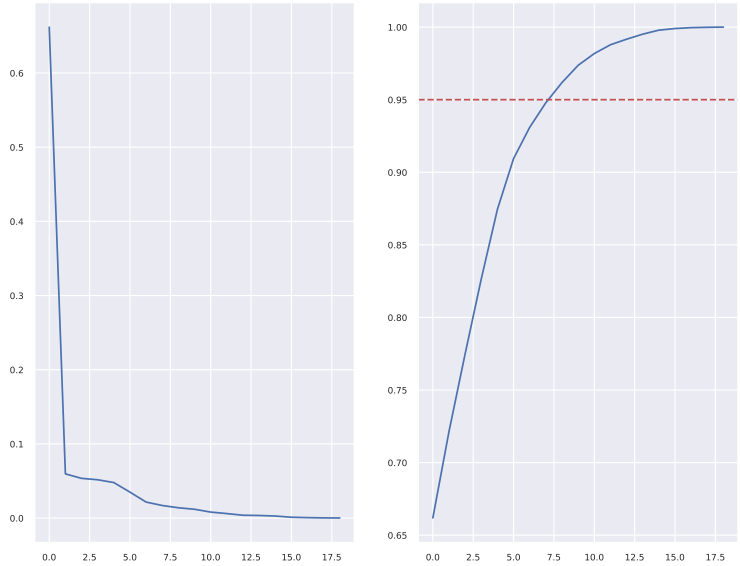


Figure 19: Left: PCA screeplot, Right: Percent Variance Explained from k Principal Components.

The training set had only 24% of its values as 1 (high volatility). Not using any class-weighting leads to severe problems, however. In the exploration phase, we fit an unweighted Logistic Regression model and yielded an accuracy of 75%. However, consider the following classification report:

	precision	recall	f1-score	support
0	0.75	1.00	0.86	21733
1	0.47	0.00	0.00	7066
accuracy			0.75	28799
macro avg	0.61	0.50	0.43	28799

Figure 20: Unweighted Logistic Regression Model Classification Report. Note the 0 recall score of class 1.

This model is heavily biased towards class 0 - not high volatility. We, therefore, adjust the model with balanced class weighting, only to obtain a model with 50% accuracy, but a higher F1 score for class 1:

Accuracy: 0.5018229799645821

	precision	recall	f1-score	support
0	0.79	0.47	0.59	21733
1	0.27	0.61	0.38	7066
accuracy			0.50	28799
macro avg	0.53	0.54	0.48	28799

Figure 21: Class-weighted Logistic Regression Model Classification Report. The accuracy is equal to random guessing.

Thus, we have seen that due to the imbalanced data, we can get a 75% accuracy by always predicting 0 but having 0 recall for class 1. This is, of course, not what we want to have. When accounting for the imbalanced dataset, we get down to 50% accuracy, which naturally, is also not good. After initial exploration of different modeling options, we propose the following model pipeline to streamline the modeling process:

1. ***Train/test split***: Split the data into train/test sets of 80%/20% of the data. The next steps are performed on each set of data.
2. ***Scaling***: For each feature, subtract its mean and divide by its standard deviation. This step is necesary to perform PCA.
3. ***PCA Decomposition***: Compute the PCA representation of the features and retain the scores of components. The number of components to keep is selected via grid search.
4. ***Model Fitting***: Fit a Logistic Regression model to the retained Principal Components. The regularization and class weight parameters of the model are selected via grid search.

Using a grid search, we explored the performance of the following model parameters: number of Principal Components $[1,3,5,7]$, Logistic Regression regularization$[0.0001, 0.01, 0.1]$, and Logistic Regression class weighting $[0:1.33, 1:4.027, \text{balanced}]$. "Balanced" class weights w for class j are defined as w_j =n_samples / (n_classes * n_samplesj), where n_samplesj is the number of samples in class j. The other class weighting scheme is defined as w_j = n_samples / n_samplesj. The best-performing model that became our 'final model' had the following specifications: number of Principal Components: 7, Logistic Regression regularization constant: 0.0001, balanced class weights. The classification report is below.

	precision	recall	f1-score	support
0	0.78	0.54	0.64	26940
1	0.28	0.54	0.37	9058
accuracy			0.54	35998
macro avg	0.53	0.54	0.50	35998

Figure 22: Final Logistic Regression Model Classification Report.

Unlike the previous part of the paper, we do not show the obtained model coefficients. Principal components yield no interpretation, as they are linear combinations of lexical features. Furthermore, as the model's accuracy is low, there is little point in interpreting its coefficients.

Despite addressing the issue of highly correlated predictors and class imbalance, our models perform poorly. Several reasons would explain it. First, that the predictors do not adequately explain the variance in the response. Simply put, it's likely that thread content and *Bitcoin* volatility are not related. Second, a linear model is too simplistic to describe a potentially non-linear relationship. In our analysis, we consider each thread an independent event. In reality, this is too simplistic of an assumption. Thread features could be nested within subtopics or users. Finally, *Bitcoin* volatility is time-dependent, based on the definition of volatility. If *Bitcoin* was highly volatile on one date, it is likely that volatility of the next day is close. Thus, our assumptions of independence of threads and *Bitcoin* volatility, as well as our choice of predictors for the response, were inadequate to yield a discriminatory predictive model. Also, one must take into account that the volatility is a 30-day rolling volatility. As a result, it may very well happen that price events 15 to 30 days ago still have such an impact that it is considered a high-volatility day. However, these days may not have any impact on a thread posted on a certain day.

6 Summary & Discussion

Starting with data describing *r/Bitcoin* activity, *r/Bitcoin* thread content, and *Bitcoin* price index, we performed an in-depth analysis to gain insights on *r/Bitcoin* and its relationship with *Bitcoin* price and volatility. The gained insights can best be summarized by answering the research questions:

1) What trends in forum behavior do the data reveal? How do these trends change over time? Thread activity, measured by the number of new threads posted per day, consists of a set of consistent, active members and momentum activity, which correlates with *Bitcoin* price. Specifically, as *Bitcoin* price increases, so do the number of new threads and comments per thread. Much of *r/Bitcoin* thread activity comes from users with temporary interest in the topic, who subsequently neglect or delete their accounts as soon as the price development is unsatisfactory.

2) Are *Bitcoin* price and volatility good predictor(s) of *r/Bitcoin* forum behavior? Interestingly, *Bitcoin* price and volatility have a positive correlation. We find that *Bitcoin* price and volatility, combined with a linear, squared and cubed time component, are acceptable predictors of the number of new threads per day. This corroborates our findings from exploratory data analysis: the core of *r/Bitcoin* activity trends stem from a small, devoted fan base. Fluctuations in activity correlate with *Bitcoin* price and volatility. Although these predictor variables can explain up to 66% of the total variation in the number of threads, they lack clear and coherent interpretability. On the other hand, the number of threads on a specific day is an excellent predictor of the number of comments that these very threads will in total receive. Our models are not without limitation; in future research, we would consider more complex models to capture nested relationships and incorporate other external variables.

3) What style, tone and topics characterize *r/Bitcoin* thread content? *r/Bitcoin* threads are, on average, written at a level understood by a college graduate. Thread tone is predominantly positive, especially on an aggregated level. While selected threads may indicate negative sentiment, the entirety of threads is positive, in some cases even exuberant. Topic modeling reveals a five-topic categorization of threads into: (1) exchanges/payments (2) financial market (3) blockchain (4) question and answer (5) philosophical discussion. A low-dimensional representation of the threads colored by dominant topics reveals a structure with many small clusters of threads, likely pertaining to a particular sub-topic. Many clusters contain multiple prevalent topics, showing that topics and threads are strongly interrelated and overlapping. We conclude that *r/Bitcoin* threads are rather sophisticated, and the users seem to maintain a relatively positive community. *r/Bitcoin* appears to be a place where *Bitcoin* enthusiasts conduct fruitful discussions.

4) Using features extracted from the thread text, can we successfully identify *r/Bitcoin* threads that were posted on days with high *Bitcoin* volatility? A linear classifier with lexical features and topic weights as input does not successfully discriminate *r/Bitcoin* threads posted on days with high volatility. This indicates that thread content is likely uncorrelated with *Bitcoin* volatility. Furthermore, the assumption that each *r/Bitcoin* thread is independent of other threads is too simplistic. Future research should thus look into more sophisticated model

building, which properly accounts for violated assumptions. Another exciting avenue worth exploring, which also happens to be more viable, is to define a rolling 20- or 30-day moving average and to predict, based on thread contents, if the *Bitcoin* price on that day is above or below the moving average. This would offer better conditions for independence and thus poses an attractive field worthy of exploration.

7 Conclusion & Outlook

By focusing our analysis on four research questions, we obtained helpful insight into the relationship between *Bitcoin* and *r/Bitcoin* activity, as well as a general understanding of *r/Bitcoin* thread content. However, this is only the beginning of what can be unearthed from the collected data. Revisiting the second research question, we can model the *Bitcoin* and *r/Bitcoin* activity relationship with latent factor analysis; "True Bitcoiners" and "attention" would be underlying (latent) factors that influence both the number of threads and the number of comments. An extended analysis using Structural Equations Modeling appears to be an avenue worth exploring in this regard. We can further extend our third and fourth research questions to more complex studies, such as considering alternative topic models to LDA and engineering more lexical features to perform a more thorough analysis. With additional user data, it would be of further interest to perform user segmentation to identify the set of "True Bitcoiners." The challenge of this problem is to obtain enough data of consistent posting behavior. With the highly fluctuating popularity of *Bitcoin* and the coming and going of users, more time and effort will be needed to reveal these dedicated users.

Also, *Bitcoin* is the incumbent regarding cryptocurrencies. Thus, a promising extension of our paper would be to investigate the same relationships with other cryptocurrencies such as *Ethereum* and *Ripple* and their corresponding subreddits. As these are considerably smaller[10], we would expect to see a more tight-knit community and less thread activity stemming from attention due to price movements as the users of these subreddits will probably be active predominantly for reasons such as the firm belief in a specific technological approach underlying a given cryptocurrency.

On a personal reflection, we have noticed the high quality of the *Reddit* dataset as a source of data, as well as the ease with which we could retrieve it thanks to the Pushshift and *Coindesk* APIs. Thus, it has been a valuable experience of working with these APIs and to further put data science techniques from a broad range of domains into practice.

[10] At the time of the writing (May 2021), the member statistics for r/Bitcoin, r/Ethereum, r/Ripple are 3 million, 982 thousands, and 316 thousands, respectively.

References

(1) Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The Pushshift Reddit Dataset. *Proceedings of the International AAAI Conference on Web and Social Media 14*, Accessible via: https://ojs.aaai.org/index.php/ICWSM/article/view/7347, 830–839.

(2) Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res. 3*, Accessible via: https://dl.acm.org/doi/10.5555/944919.944937, 993–1022.

(3) Bukovina, J., and Marticek, M. Sentiment and Bitcoin Volatility, MENDELU Working Papers in Business and Economics 2016-58, Accessible via: https://ideas.repec.org/p/men/wpaper/58_2016.html, Mendel University in Brno, Faculty of Business and Economics, 2016.

(4) Buntain, C., and Golbeck, J. In *Proceedings of the 23rd International Conference on World Wide Web*; Accessible via: https://doi.org/10.1145/2567948.2579231, Association for Computing Machinery: Seoul, Korea, 2014, pp 615–620.

(5) De Choudhury, M., and De, S. (2014). Mental Health Discourse on reddit: Self-Disclosure, Social Support, and Anonymity. *Proceedings of the International AAAI Conference on Web and Social Media 8*, Accessible via: https://ojs.aaai.org/index.php/ICWSM/article/view/14526.

(6) Hutto, C., and Gilbert, E. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*.

(7) Knittel, M. L., and Wash, R. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*; Accessible via: https://doi.org/10.1145/3290607.3312969, Association for Computing Machinery: Glasgow, Scotland UK, 2019, pp 1–6.

(8) Koltcov, S., Ignatenko, V., Terpilovskii, M., and Rosso, P. (2021). Analysis and tuning of hierarchical topic models based on Renyi entropy approach. Accessible via: https://arxiv.org/abs/2101.07598.

(9) Ladokhin, S. (2009). Forecasting Volatility in the Stock Market. *Working Paper. Vrije Universiteit Amsterdam*, Online. Retrieved via: https://beta.vu.nl/nl/Images/werkstuk-ladokhin_tcm235-91388.pdf Last Access: Apr 29, 2021.

(10) macrotrends GameStop Market Cap 2006 - 2021 | GME, Online. Accessed via: https://www.macrotrends.net/stocks/charts/GME/gamestop/market-cap. Last Access: May 20, 2021, 2021.

(11) Readability Formulas The New Dale-Chall Readability Formula, Online. Accessed via: https://www.readabilityformulas.com/new-dale-chall-readability-formula.php Last access: May 25, 2021.

(12) statista Market capitalization of Bitcoin from April 2013 to May 17, 2021 (in billion U.S. dollars), Online. Accessed via: https://www.statista.com/statistics/377382/bitcoin-market-capitalization/ Last access: May 20, 2021, 2021.

(13) van der Maaten, L., and Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research 9*, Accessible via: http://jmlr.org/papers/v9/vandermaaten08a.html, 2579–2605.

A Python Code

The Python code used for all steps in the paper, as well as the data retrieved via the API retrieval together with helper files created in the process have been published in the GitHub repository `jasperschroeder/BigDataClass`. Access to Prof. Caren has already been granted. In case of any difficulties in accessing, please send an e-mail to `schroejas@gmail.com`.

B Thread Features

We compute 21 features from `py-readability` for each thread. Below is an example of the computed features of the example text used in Figure 10:

```
readability
----------------------------------
dale_chall                 8.038189

sentence info
----------------------------------
characters_per_word        3.666667
syll_per_word              1.000000
words_per_sentence        18.000000
sentences_per_paragraph    1.000000
type_token_ratio           0.777778
characters                66.000000
syllables                 18.000000
words                     18.000000
wordtypes                 14.000000
sentences                  1.000000
paragraphs                 1.000000
long_words                 2.000000
complex_words              1.000000
complex_words_dc           4.000000

word usage
----------------------------------
tobeverb                   0.000000
auxverb                    1.000000
conjunction                1.000000
pronoun                    3.000000
preposition                1.000000
nominalization             0.000000
```

The package offers more features, including different readability metrics and information on sentence beginnings, which we opted not to use. The features all depict lexical information about sentence structure and part of speech tagging.