# Trading on Twitter: Using Social Media Sentiment to Predict Stock Returns

*Results show that the sentiment in tweets about a specific firm from users with less than 171 followers (median in their sample) had a significant impact on the stock’s returns on the next trading day, next 10 days and next 20 days. Sentiment in tweets from users with fewer than 171 followers that were not retweeted had the greatest impact on future stock returns.*

Collected all public tweets that contained the relevant $ symbol with an S&P 500 stock ticker from Twitter using a developer account. Retrieved about 3.5m tweets, of which 16.02% were retweets. They **excluded** all the tweets that contained more than one ticker symbol, because they weren’t sure if the information in the tweet pertained to one firm or all firms equally. This resulted in a total of about 2.5m tweets.

Used a word analysis strategy, each word in a tweet was matched to a dictionary of terms to determine its sentiment. Simply counted words using POS and NEG tags. It is an imperfect measure of sentiment, because it cannot detect subtle meanings in English, such as sarcasm or semantic word groups that combine positive and negative words (e.g. “not bad”).

Didn’t include emotions in their analysis (limitation).

CLT argues that positive and negative sentiment may have different effects (Liberman & Trope, 1998; Bar-Anan et al., 2006: fujita et al., 2006), so it is important to track both positive and negative sentiment, because they may have different effects.

Use three measures to measure sentiment: N/T, (P-N)/(P+N) and log(1+P/1+N). 🡨 **Interesting to test with.**

Examine whether the speed of information dissemination (i) reflected by the number of followers and (ii) retweet history is associated with future returns.

**Interesting way of trading:**We followed the approach of Tetlock et al. (2008) and constructed two equally weighted portfolios, one long, one short. At the close of each trading day, we analyzed the sentiment in that day’s tweets about speciﬁc ﬁrms using pos2 (we choose pos2 because it takes into account both positive and negative sentiment). We purchase ﬁrms in the top 10% and short sell the ﬁrms in the bottom 10%. Not all ﬁrms receive tweets each day, so the number of ﬁrms varies from day to day. We used three different holding periods (1 day, 10 days, and 20 days), and at the end of the holding period, we close out our long and short positions. Because we are simultaneously taking long and short positions, there is no need to consider market return as a control; any rise or fall in the market as a whole is controlled for by the simultaneous long and short positions.  
  
The 1-day holding period produces positive returns, but because the strategy executes trades every day, the returns become negative after including trading costs. The trading strategies using 10- and 20-day holding periods produce signiﬁcant positive returns, both before and after trading costs.

These returns compare favorably with those in Tetlock et al. (2008), who found trading strategies using sentiment in news stories produced returns of 23.17% before trading costs and–2.71% after trading costs(i.e.,a loss). In other words, the results in Table 8 show that a trading strategy with a 10- or 20-day holding period that balances long and short positions results in meaningful positive returns.

# 

**Also interesting as one might think otherwise on the following trading day. Do take into account that btc is 24/7, so the whole ‘next day’ trading strategy is probably not useful.**

We argue that a social contagion process is at work. Tweets spread positive or negative sentiment about a stock through the market and can inﬂuence prices, and thus the returns from trading those stocks. Sentiment can spread quickly; for example, network hubs like Jim Cramer can send a tweet that has positive or negative sentiment about a stock and his 650,000 followers immediately see its sentiment. If these followers act, the stock price can respond very quickly. Thus, we conclude that Twitter users with many followers have a market impact similar to traditional news media; the impact of the sentiment in their tweets disseminates rapidly and is quickly incorporated into stock price. However, there arenosigniﬁcantstockreturnsonfuturetradingdaysandthusitisdifﬁculttoproﬁt from a trading strategy based on them. In contrast, if a user with few followers sends a tweet, few people will see it, and even if they quickly act on its sentiment, the small number of trades will have little immediate impact on the stock price. Over time, however, as the sentiment diffuses through the market, the sentiment willgraduallyaffectthestockprice.Thesentimentintweetsfromuserswithfewer followers had a stronger impact on stock returns the next trading day and over the next 10–20 days compared to tweets from users with many followers. Because the change is gradual, there is an opportunity to proﬁt from this as a trading strategy. The diffusion is slower for tweets that are not retweeted, and thus they have the greatest impact on returns on future trading days and offer the greatest opportunities to proﬁt from a trading strategy based on this.

# Inﬂuence of Social Media over the Stock Market

Differentiate between type of investors. Could try looking for indicators of TA trades, which would be interesting.

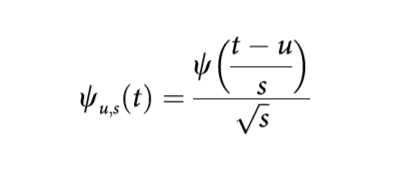
Use dependent variable vrisk (variation of risk), which collects the daily variation of the VIX index. In the logit model, this is a binary variable in terms of 1 and 0, so that if the variation is positive, the variable takes the value 1 and the value otherwise.

**Variables to look into: daily sentiment, daily experience, daily holding period, and number of followers>** The calculation of daily sentiment, daily experience, and daily holding period is as follows: the sentiment of each message was multiplied by the number of messages posted by the user who wrote that message on a certain day t, and then the calculation of daily average of all users is as a weighted average, resulting in a sentiment that is always negative (see Table1); the study multiplies the experience of each user by the number of messages that user posted on a certain day t, and then the study calculates the daily average of all users as a weighted average; the study multiplies the holding period of each user by the number of messages that user posted on day t, then calculating the daily average of all users as a weighted average. Acting this way, the experience, the sentiment, and the holding period for a certain day with high message activity exert a heavier effect than the experience, the sentiment, and the holding period of another day with low message activity. Finally, the study calculates followers as the average of the followers for each user that post one or more messages in a given day t.

Concentrating on technical investors, experience and holding period are the two main characteristics that combine in order to avoid a raise in market risk. Focusing on nontechnical investors, experience and sentiment appear as the two main explanatory features, and evidence shows that this type of investors can contribute to avoiding a raise in market risk even if the environment is strongly pessimistic.

# What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis

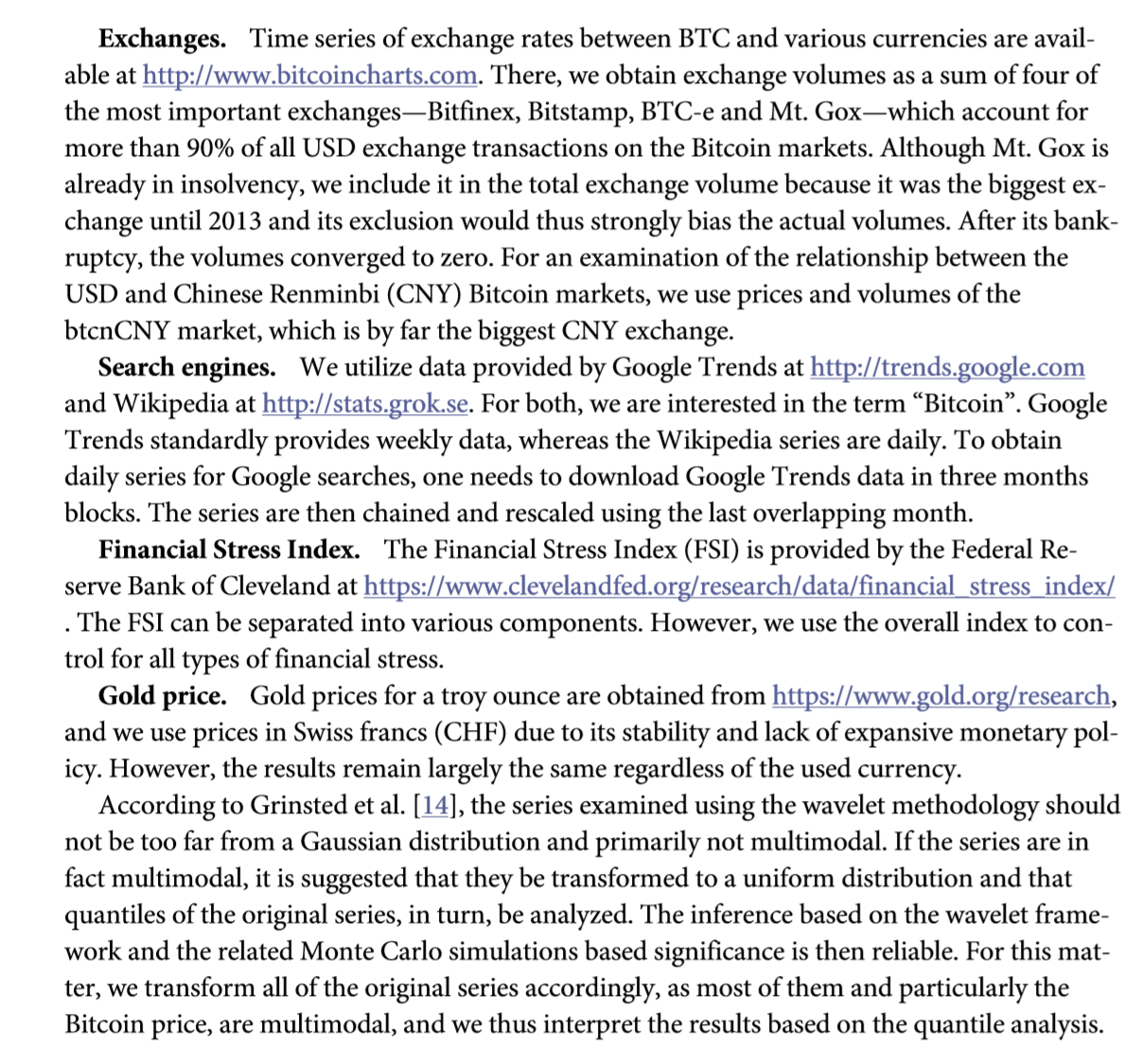
A wavelet is a complex-valued square integrable function generated by functions of the form:



<http://blockchaininfo> freely provides very detailed series about Bitcoin markets. On a daily basis, the following time series used in our analysis are reported:

* Total bitcoins in circulation
* Number of transactions excluding exchange transactions
* Estimated output volume
* Trade volume vs transaction volume ratio
* Hash rate
* Difficulty

Also use:



They don’t use any social media, which is my main focus. However, for future research I could add something like: **combine my data and the corresponding results with other factors such as x1,..xn.**

Another reason why statistical analysis might be more suitable in crypto than in normal stocks is because of the availability of data. This is because you’re not able to track all transactions that occur using the USD or other currencies, but for Bitcoin this is possible.

**Influential factors Bitcoin**First, although the Bitcoin is usually considered a purely speculative asset, we find that standard fundamental factors—**usage in trade, money supply and price level**—play a role in Bitcoin price over the long term. These findings are well in hand with standard economic theory, and specifically monetary economics and the quantity theory of money. Second, from a technical standpoint, the increasing price of the Bitcoin motivates users to become miners. However, the effect is found to be vanishing over time, as specialized mining hardware component have driven the hash rates and difficulty too high. Nonetheless, this is a standard market reaction to an obvious profit opportunity. A reversal is identified at the end of the analyzed period. Third, the prices of bitcoins are driven by investors’ interest in the crypto-currency. The relationship is most evident in the long run, but during episodes of explosive prices, this interest drives prices further up, and during rapid declines, it pushes them further down. This is well in hand with previous research on the topic [10,11]. Fourth, the Bitcoin does not appear to be a safe haven investment. Finally, fifth, although the USD and CNY markets are tightly connected, we find no clear evidence that the Chinese market influences the USD market. We speculate that such behavior is due to the analyzed data structure and its frequency, and trading algorithms which efficiently capitalize on potential arbitrage opportunities between different Bitcoin exchanges. Overall, the Bitcoin forms a unique asset possessing properties of both a standard financial asset and a speculative one.

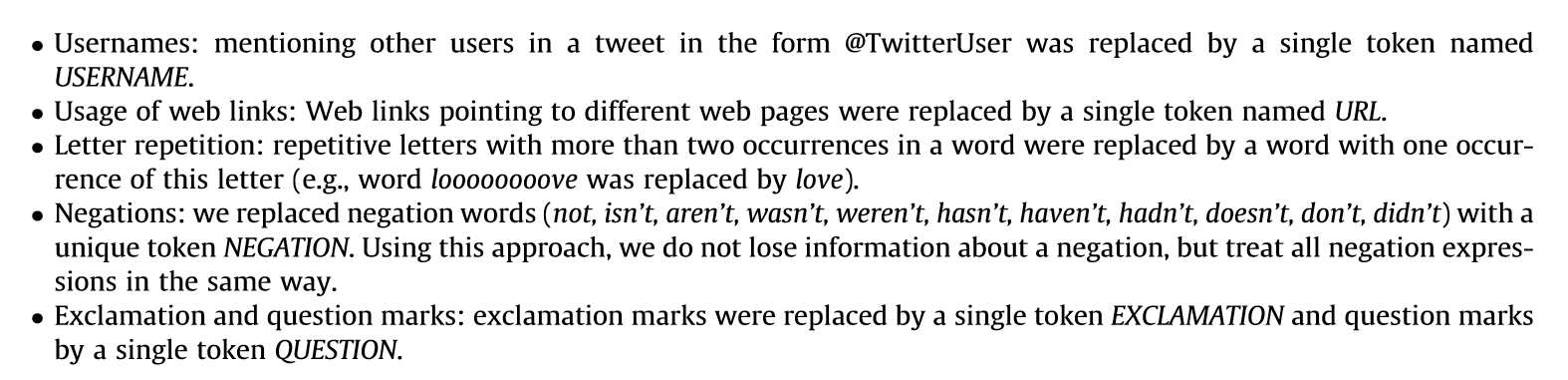
# Stream-based active learning from sentiment analysis in the financial domain

Paper analyses whether the sentiment express in Twitter feeds, which discuss selected companies and their products, can indicate their stock price changes.

Static Twitter data analysis problem, explored in order to determine the best Twitter-specific text preprocessing setting for training the SVM sentiment classifier. In the static setting, the Granger causality test shows that sentiments In stock related tweets can be used as indicators of stock price movements a few days in advance, where improved results were achieved by adapting the SVM classifier to categorize Twitter posts into three sentiment categories of positive, negative and neutral.

The experiments in analyzing stock market sentiments of a particular company show that changes in positive sentiment probability can be used as indicators of the changes in stock closing prices.

Interesting way of preprocessing the data (note: I’m not sure how they treated multiple exclamation marks as from a semantic viewpoint: ‘I’m mad!’ isn’t as bad as ‘I’m mad!!’):



Replacing negation is found to be especially useful.

Their data is hand-labeled and thus specific for their problem. Sentiment data would be annotated by a program, making it more generalized (perhaps not as functioning).

Method of ‘Syntactic clustering of the web’ was used to remove tweets who are very similar (duplicates).

# Ideas

It would be interesting to build some form of a model that ranks words and users (by e.g. meta-data and historical accurateness) and uses this to predict the price of bitcoin going up and/or down (i.e. can also try regression).

A stereotypical sentiment analysis would also be interesting, where we could e.g. use a RNN to predict future sentiment and price of a coin based on the average sentiment of all users. We could also add a weighing factor/ranking factor here that prioritizes certain users based on e.g. meta-data.

Saving the sentiment of a user would also be interesting as a sentiment of -1 might not be the same for other users. So averaging the sentiment overtime might increase the model’s performance. 🡨 **Gemiddelde van een user pakken en dan verschil.**

Use the replacement of negations etc. see previous chapter.

# Aantekeningen

* Valideren: kijk naar een sample van tweets waarbij je als mens probeert te beoordelen of de score die door het systeem wordt gegeven, logisch is of niet.
* Specifieke woorden die voor BTC wellicht interessant zijn (denk aan FOMO).
* (-1, 1) score zijn vaak regel gebaseerd.
* ML is logischer om classificatie te doen: pos, neut, neg.
* ‘Too cheap’ is an indicator for it going up, so should be measured as positive.
* Can also be used as a counter:
  + Either fine-tuned to the absolute most influential people and use that as a pos/neg index. Pos 🡪 heavy bullish, neg 🡪 heavy bearish.
    - https://www.reddit.com/r/CryptoCurrency/comments/8gmj0g/top\_60\_most\_followed\_crypto\_twitter\_accounts\_who/
  + Mass population is actually the other way around. Because if everyone thinks it’s going up, then it will most likely go down.
* Stance (opvatting tov een onderwerp): Onderwerp: Stance towards bitcoin.
  + Meta-data
  + Sentiment 🡨 heel basisch.
  + **Eventueel impact op btc meta-data (**n. transactions etc) meten.
  + **Data ophalen tijdens project en als test data gebruiken.**
  + Woordgebruik 🡨 Keywords samenvoegen (frequentie) 🡨 Wegingen als Information Gain en TFIDF, nemen info van de klasse mee.
  + **Labels creeren op dagbasis en dan classificatie van risky, neutraal, not risky.**
  + Classificatie van wel/niet-risky. 🡨 short-term vs **long-term**.
  + Hoe ga je valideren dat iets een risky/niet risky investment is? Wat is je golden standard? Kan je in het verleden zien dat bitcoin risky/niet-risky was.
* Focussen op risky vs niet risky.
* Sentiment index
* Features moet je wel theoretisch kunnen motiveren (meta-data van gebruiker vs prijs).
* Twitter API geeft alleen tweets die hooguit 7 dagen teruggaan.
* Van de afgelopen 7 dagen users scrapen en daar individueel tweets scrapen.

# Questions

* I’d like to discuss the notes I have (ideas etc.). 🡨 See ‘aantekeningen’.
* Determine strong influencer 🡪 determine social graph 🡪 weighting of his followers (determine bots etc.). Something like a cumulative follower count. This determines the reach of his tweet if people interact.
  + Remove bots
  + Cumulative follower count.
* What has the previous student done? Can I continue his work within the boundaries of my own project. I have until end of January for this. 🡨 Nothing computational.
* My tutor basically advised me to talk to you to be up-to-date regarding the state-of-the-art research, such that I can expand upon that.
* **Research questions:**
  + Twitter Users stance on Bitcoin
  + Combine meta-data, sentiment & keyword usage.
  + Sentiment is generalized and thus biased.
  + Focus on Tweets with $btc.
  + Research name: Twitter User Stance on Bitcoin prices (**can also do BTC meta-data**).
  + Q1: Should I use BTC price or meta-data?
  + Q2: What to use as a classification; price, risky/non-risky or something else? Go Long or Short for the coming 24h?
  + Research questions:
    - Can the sentiment of Bitcoin act as an indicator of fluctuation in prices?
    - Can keyword usage in Tweets act as an indicator of fluctuation in prices?
    - Can the meta-data of Twitter accounts act as a viable weighing factor for the aforementioned relations?
    - Can a viable trading strategy be developed by combining the aforementioned relations? *Price going up/down with a certain percentage for the coming 24h/7d/30d. Magnitude of move would be more interesting though.*

# Data gathering

1. Gather data of last 7 days, to gather active users using the $btc.
2. Preprocess data (see previous paper).
3. Scrape the unique users’s data.
4. Gather user-information of users that are left.
5. Preprocess Tweets (see previous paper).

## Questions class

1. In eerste instantie wilde ik alle retweets en reacties verwijderen. Echter zijn reacties vaak vol emotie en kunnen deze juist een goede indicatie geven van wat er in de markt aan de hand is. Nu zijn er natuurlijk ook veel reacties welke niks met het onderwerp zelf te maken hebben (e.g. zelfpromotie). Denk je dat het het waard is om een soort filter te verzinnen, of is het handiger om alleen tweets te verzamelen en dus reacties/retweets volledig te verwijderen? Ik denk zelf dat het tweede de meest sensible approach is, omdat ik zelf bang ben dat het verzinnen van zo'n filter ook averechts kan werken.

RT's met extra informatie houden, rest verwijderen (dus exact dezelfde verwijderen).

1. Op dit moment is mijn dataset nog niet gefilterd op locatie en hoewel het logisch is om op Amerikaanse tweets te focussen als je focust op de Amerikaanse markt, weet ik niet zeker of dit voor BTC ook het geval is. Het is een globale markt en dus hebben alle sentimenten/keywords mogelijk invloed. Ik zit er dus zelf aan te denken om te filteren op tweets welke in het Engels geschreven zijn (de taal vd tweet wordt namelijk meegegeven door de API van Twitter).

Als de globale assessment van Bitcoin, focus op Engels. **Wel beargumenteren.**

# Final

Questions:

* Can the stance of Twitter users towards Bitcoin act as an indicator of fluctuation of BTC prices?
* Can keywords, which originate from tweets, act as an indicator of fluctuation in prices?
* Can the meta-data of Twitter accounts act as a viable weighing factor for the aforementioned relations?
* Can a viable trading strategy be developed by combining the aforementioned relations?
  + Focus on going long/short for the coming 24h/3d. If possible, expand such that a reliability index can be established, such that a user knows what the strength of the prediction is (**Could be the prob the algorithm assigns to the class**).

I removed all tweets if:

* RT’s
* Non-english
* More than one type of ticker symbol (e.g. $btc 🡨 ticker symbol for btc).

Still need to:

* Preprocess tweets further:
  + Add neg etc (see prev tweet). 🡨 **Done**
  + Process all the Unicode signs \n etc **🡨 Done**
* Create own rows by: looping over 30 days of data, then checking if the day after the price of btc went up or down wrt the previous day so: (t – t-1) – margin\_trading 🡪 if pos, then trade is viable, else trade is not viable. 🡨 can also do 60 days.
* **SUPER IMPORTANT:** Make sure that the test data is data that we **haven’t** used yet, which is also in the **future**.

Models:

* **Keywords**
  + Gather keywords per date (text mining).
  + Create CNN with keywords on y axis and datetime on x axis.
* **Sentiment/stance**
  + Classify words as pos, neut, neg. 🡨Should I build a separate classification algorithm for this? Hard to differentiate between ‘not that bad’ and ‘bad’. 🡨 as we classify per tweet, we can use the user meta-data **and** meta-data of the tweet (n RT’s) to influence the total count.
  + Create CNN with y axis sentiment and x axis time. Not sure about the y axis yet as tweets per day will vary.
  + Would be cooler if I can do users on y axis and and time on x axis, but then I’d have to follow a specific user set which is kinda hard. 🡨 Can do weighted sampling of users for a specific day. This way I can introduce the importance of twitter followers etc. Then an additional ‘user’ could be the weighted average as well so setup: user\_1 .. user\_n, user\_avg would be the y-axis. 🡨 **It’s important that there are similarities, because users are constantly replaced, meaning that the replacement should somehow have some form of the same Sentiment as other users. 🡨 Cannot simply use all users, cause there will be a lot of zeroes for users who aren’t tweeting on a particular day.**
  + Can sample with prob based on user meta-data **and** meta-data of the tweet (n RT’s) 🡪 add their sentiment over t days
    - Repeat procedure to generate N rows
    - Each particular row will new new sampled users based on those weights.
    - Input size cnn should be x = 30/60 and y = 320 (meaning 320 sampled users) 🡨 Check minimum per day first in dataset to determine y.
  + **SHOULD** actually try the above as well for keywords. SO a 3d conv net with: users, time and set of keywords (keywords = 3rd dimension).
  + Another possibility is to use the min amount of tweets we have for a day as the max input size. Then for each day that we use, apply maxpooling or an additional CNN up until we’re at the right size. 🡨 <https://arxiv.org/pdf/1406.4729.pdf> CNN implementation with varying input size. 🡨 Don’t think this is a good idea, because the input size will then be huge.

# Milestone 3 Mick van Hulst (s1013954)

My project is Product-oriented as I’m going to develop a predictive model that will predict whether or not the Bitcoin price will go up or down in the coming 24h. Before even using these methods, I’ll have to preprocess the data which will consist of:

1. Focusing solely on users who have at least been active over a set amount of time (have to decide on the boundaries).
2. Removal of stop words.
3. Focusing solely on English tweets and tweets who only have a single type of ticker symbol in their tweet (either ‘$btc’ or ‘$BTC’). Approach based on [1].
4. Replacing certain words like negation with the keyword ‘NEGATION’ as prior research found that to be useful. Approach based on [0].
5. Removal of retweets. Although retweets have an impact on the reach of a certain tweet, they do not provide additional sentiment/textual information.
6. Other based on further research and/or advise tutors.

After preprocessing, I’d like to propose the following models, which can be divided in two categories and where each model predicts whether the price of Bitcoin will go up or down after 24h from making a prediction:

1. **Classification using keywords of Tweets:** This model will consist of a ConvNet which requires a matrix input. The input for this model will be a 2-dimensional matrix consisting of the usage of keywords over t timesteps for a certain user.
   1. To generate data, I’ll first get a certain price of Bitcoin at time t and t-1 and take the difference of those two (this will tell me whether the price went up/down). This difference will be weighted as traders pay a fee to make trades.
   2. Second, I’ll generate 2d matrices based on keyword usage of users. These keywords will be generated first based on a model described below. For each user, I can then generate a: k x t matrix, where k is a set of keywords that’s present or not in the set of tweets[[1]](#footnote-1) for a certain user at timestep t (t is the amount of timesteps we look in the past, e.g. 30 days). This matrix will be assigned a binary variable (price going up/down) based on the fluctuation of Bitcoin price at t+1 using the proposed method (see 1a).   
      *Extra explanation for classification: given an arbitrary matrix m, which consists of keywords from timestep t-30 to t, we predict whether or not the price will go up after 24h. We then evaluate this prediction by using the price of t+1 (i.e. price after 24h) to see whether or not the model was correct or not.*
   3. As the set of keywords will have to be fixed, I’ll need to do some preprocessing to find which keywords in a tweet are most important. This means that I’ll first have to do a correlation analysis to find words which relate to a fluctuation in price of Bitcoin. These words will be assigned a token such that I can use them in the model.
   4. Classification will consist of gathering keywords for certain users from timestep t-n to t (where t-n is the amount of time we look in the past and t is the current date), then classifying these matrices on a per user basis, to predict whether or not the price will go up or down after 24h. This will result in a list of predictions for each user, where we’ll perform a majority vote/weighted average to come to our final prediction for the price of Bitcoin going up/down after 24h.
2. **Classification using clusters of similar keywords of Tweets:**
   1. Basically same as 1., but instead of finding most useful words, we **cluster** similar words and then when new tweets arise, the corresponding words are assigned to clusters and these clusters are added as k x t matrix, where the clusters are k.
3. **Classification using the stance of Twitter users on Bitcoin:** For this model I’d like to use the sentiment of tweets. Prior research [1] used a dictionary[[2]](#footnote-2) that classifies words in a tweet as either NEG or POS and then uses certain formulas to determine negativity/positivity of an entire tweet. As they found that the used formulas give different results, my research will use all three formulas to generate three sentiment indexes. This sentiment will be used to propose three ConvNets, where I’ll use the same form of assigning a class to a data point as model 1:
   1. *Single user sentiment:* Single user 1d ConvNet that uses a 1 x t x s matrix[[3]](#footnote-3), where we classify the price as going up/down based on the development of the sentiment of a user over timesteps t by using three channels. I’d then train the classifier on all users at timestep t. Prediction will be based on classifying all users for a certain timestep and then taking a majority vote/weighted average which is comparable to method 1.
   2. *Ranking based model*: I’ll use a ranking model to rank users based on their Twitter meta-data (e.g. user activity) and perhaps content of their tweets. Based on that ranking model I’ll select the top n users and use them as my user input. Then each datapoint will consist of: n x t x s matrix, where t is the amount of timesteps (e.g. 30 days), n is the sentiment of the users that we’re following and s consists of three sentiment channels. For this model it’s highly important that we follow users who are active as inactive users means that there’s no sentiment to track for a given day.   
      **Problem[[4]](#footnote-4):** Assuming I have 365 days of data, then I’d be able to only generate 335 data points as I’m following a fixed amount of users, meaning that n is set and the only thing that varies is t. ***To solve this I propose model c.***
   3. *User cluster model*: I’ll use a clustering model to cluster users based on their tweet content. Based on this model, each datapoint will consist of: n x t x s matrix, where t is the amount of timesteps (e.g. 30 days), n is the set of clusters which were generated using a clustering algorithm and s consists of three sentiment channels. As we’re working with clusters, the cluster will consist of multiple users and I’ll be able to sample users from these clusters to generate multiple rows, thus creating multiple data points with these clusters per set of timesteps.
   4. *Keyword sentiment cluster:* Use method to cluster words based on sentiment, then create a t x s matrix, where t are the timesteps and s is the list of clusters. When analyzing a tweet we see which word in the tweet belongs to which cluster, then if it exists, the binary value becomes 1, else 0.
4. **Ensemble:** If all goes well, I’d like to make an ensemble of the models by either performing some form of an average or cutting the models off at a certain point and add another neural net to process the signals of both models.

In terms of techniques, I’ll use Python/R for data-preprocessing. Furthermore, I’ll use Python and the packages PyTorch/Keras (most likely PyTorch as I’d like to develop my skills using this package, but Keras if I don’t have sufficient time) for the Deep Learning part.

Resources wise, I’ll gather data using the Twitter API. At the moment I’ve been scraping for about a week and my strategy consists of gathering t amount of days of unique tweets which contain the keyword ‘$BTC’. These unique tweets will then give me a list of users who tweet about Bitcoin. As the Twitter API doesn’t allow me to go back in time more than 14 days, I had to find a workaround, which consists of scraping Tweets of individual users. This method does allow me to go back further in time than 14 days.

In terms of priorities, I’d like to focus on model 1 and model 2a first. Model 1c is a bit of a wild-card as I’m not sure how effective clustering will be. After speaking to Florian Kunneman, he mentioned that there hasn’t been that much research regarding computational methods and predicting the Bitcoin price, so in that regard, I’d like to see this as a form of a basis which I or others can use for further research.

**Sources (sorry for the ugly references, I promise I’ll do better when I write my report :P):**

[0] Smailovic, J., Grcar, M., Lavrac, N., & Znidarsic, M. (2014). Stream-based active learning for

sentiment analysis in the financial domain. *Information Sciences, 285(32),* 181–203

[1] Sul, H.K., Dennis, A.R. & Yuan, L. I. (2017). Trading on Twitter: Using Social Media Sentiment to

Predict Stock Returns. *Decision Sciences, Vol. 48, No3.*

# Additional ideas

* Adding user meta-data as an extra column to the matrix.
* A k x t matrix, where k are the words of a tweet in sequential order and t are the timesteps. A matrix is unique for a certain user over t timesteps, where the cells are filled with tokens of words that are used (not binary, but integers), meaning (ex is a 3 x 3 matrix):

|  |  |  |
| --- | --- | --- |
| * **Tweet for day 1** | * **Tweet for day 2** | * **Tweet for day 3** |
| * 4 (I) | * 8 (my) | * 8 (my) |
| * 5 (am) | * 9 (second) | * 11 (third) |
| * 6 (Mick) | * 10 (tweet) | * 10 (tweet) |

­

1. If a certain user tweets several messages per day, these tweets will be aggregated into one tweet. [↑](#footnote-ref-1)
2. This dictionary was used for the same use-case, but instead of focusing on Bitcoin they focused on the stock market. The authors who wrote this paper mention that: *“We used the Harvard-IV dictionary (Jorgenson & Vu, 2005), which is a commonly used source for word classiﬁcation in the ﬁnancial content analysis of popular press articles and Web news sites, used, for example, by Tetlock(2007), Tetlock etal. (2008), and Da, Engelberg, and Gao(2011).”*. There’s a chance that some words that are tweeted will not be known by the dictionary, these words will be hand-labeled (by either myself or some people I know from the domain) as positive or negative and mentioned in the research as a possible limitation. [↑](#footnote-ref-2)
3. This model assumes that there’s no benefit in moving a filter over the several different sentiment indexes at the same time. If there is a benefit there, then I’d have to change it to a s x t matrix, such that I can apply a filter that moves over the three different sentiment indexes at the same time. [↑](#footnote-ref-3)
4. I’ve added this model as the tutors might know some form of a solution to this problem and/or it might spark new ideas which can further aid me during my research. If not, then I’ll abandon this idea. [↑](#footnote-ref-4)