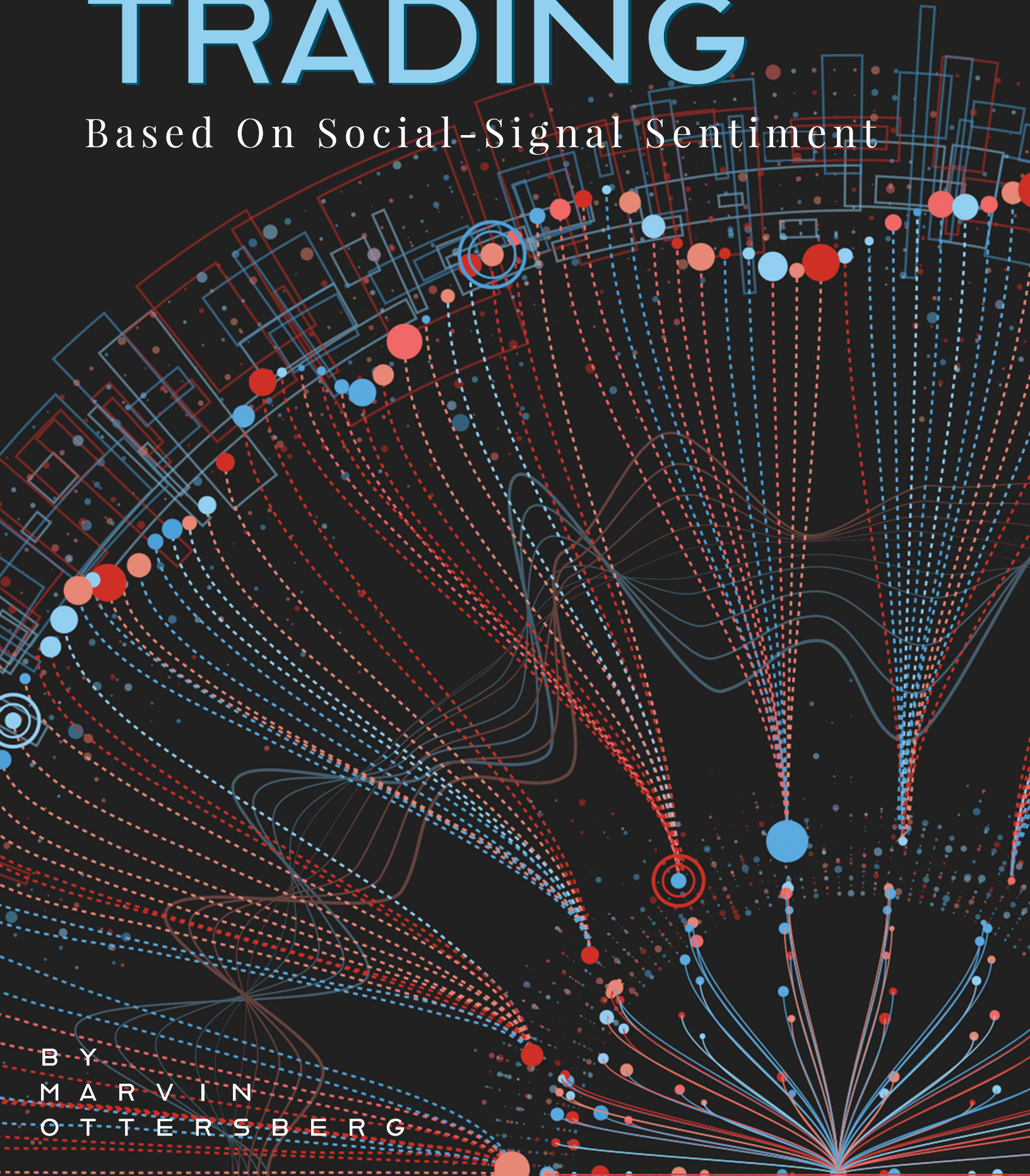
CRYPTOCURRENCY TRADING

Based On Social-Signal Sentiment



BY
MARVIN
OTTERSBERG

Social Signal Sentiment-Based Prediction for Cryptocurrency Trading

Table of Contents

Start with the Documentation

Development Environment

Code Editor

The code editor used was Visual Studio Code with their tremendous amount of extensions. Particular helpful extensions were the Jupyter Notebook, the TabNine AI supported Autocomplete and the LTeX LanguageTool that checked grammar and spell Markdown files. This way it was possible to directly write the documentation inside VSCode.

Environment Setup

To ensure version control for used python libraries the package and environment management system Conda was used.

With conda environments it is possible to work with defined python and package versions.

Setting up an environment is easy:

1. Install conda with `pip install conda`
2. Create the environment with `conda create --name myenv`
3. See if environment was created `conda env list`
4. Activate environment with `conda activate myenv`

To create an environment from an existing `requirements.txt` file add this to the second step:

```
conda create --name myenv --file requirements.txt
```

The following packages were used:

- pandas
- matplotlib
- sentiment
- streamlit
- wordcloud
- numpy
- openpyxl
- regex==2022.3.2
- pyOpenSSL
- demoji
- python-dateutil
- python-dotenv
- tweepy
- SQLAlchemy
- vaderSentiment
- psycopg2-binary
- streamlit_autorefresh
- boto
- schedule

- postgres
- python-binance
- python-kucoin

Export to requirements.txt or requirements.yml with

- `conda env export > environment.yml`
- `pip freeze > requirements.txt`

Local Development

Tweepy Stream and local Export

1. Activate the conda environment with required packages (see above)
2. Get your own API-Keys from Twitter API and Kucoin Sandbox and add them to .env-file
3. Set up a Postgres Database and add DB_URL to .env
4. Edit runner.py :
    - Uncomment line 48 for local export
    - Either add the keywords for the coins in the last line:
      Runner(['btc','ada','eth']))

      or Uncomment lines 66 - 72
5. run script via terminal:
   `python3 runner.py -k "btc,eth,ada" -i 5`

Streamlit

```
1. cd streamlit
2. streamlit run 01_💬_Tweet-Sentiment.py
3. open http://localhost:8501
```

Folder-Structure

main - Folder

main - Folder	
config.py	File to get the Environment-Variabes
Procfile	A Heroku file for starting the processes
docs	Contains the ordered Documentation

sentiment - Folder	
filter.py	Functions to filter the tweets by checking for blacklisted words, duplicates and unnecessary symbols
keywords.py	Class to build a keyword list for coins
listener.py	Class to listen and filter tweets
runner.py	Main Class. Called in Procfile and starts Listener with Keywords
trade.py	Functions for Trading. Called every hour in Heroku Scheduler.
Logs-Folder	All Logs (Heroku, Tweepy, Excel, Json)

sentiment/database - Folder	
database.py	SQL-Alchemy Connection with Heroku database
exporter.py	Export Local Tweets to Json/Excel
Trade.py	Trade Class to declare the Format and Type of each Column in the Database
Tweet.py	Tweet Class to declare Format and Type of each Column in the Database

streamlit - Folder	
01_💬 _Tweet-Sentiment.py	Main File to visualise all the data from tweets, sentiment and trades with Streamlit
financial_data.py	Functions to get the data from Heroku Database, get prices from Binance and for building Signals
streamlit_data.py	Functions to edit the data from the databases: Splitting the Dataframe, calculate average and convert to signals.
visualise.py	Functions to visualise the price chart and words.

Introduction

Social Signal Sentiment-Based Prediction for Cryptocurrency Trading.

This Project is supposed to cover two topics: It will analyse specific signals in social networks and their impact on trading digital assets, specifically cryptocurrency. This research will be used in a real-world test with the development of a bot that automatically trades cryptocurrency while considering the analysis of social signals.

Motivation

As being said by Schoen et al. [1] about social media:

> "... little is known about their overall potential, limitations and general applicability to different domains."

Chainsulting, a blockchain-company from Flensburg is keen on finding out if the idea behind social signals is working and interested in trying out new trading methods. Also, they are open for publishing this project, as it will greatly contribute to a scientific area where not much has been done before. It is obvious, that if an automatic trading bot can increase the revenue this will not likely be published. Natural Language Processing is a big part in today's world of big data and their impact is yet to be studied.

Context

This project will find context in the R&D Department of Chainsulting, a blockchain-company based in Flensburg. Contact person will be Yannik Heinze, CEO of Chainsulting (Email).

> "Chainsulting is a consulting and development company, on the subject of Distributed Ledger Technology (DLT) & Digital Assets. We show ways, opportunities, risks and offer comprehensive solutions. " [4]

More about Chainsulting on their Website or GitHub Profile.

Approach

First Phase: Research

The amount of data social media can offer is just too big to grasp, but it can be used to an advantage in trading. A study from 2013 has already shown that search trends have an impact on the price of Bitcoin:

> "Speculation and trend chasing evidently dominate the BitCoin price dynamics." [2]

There exists a lot of different social networks (Instagram, Twitter, Facebook, etc.) and after comparing them with respect to their user base, availability of API's and other factors, it will be clear which one is best for

this project.

To find the right cryptocurrency for this project, the Top 5 Coins based on market capitalisation are compared in relation to their trading volume, user base, API Access, amount of tweets about the cashtag($btc, $eth, etc.), etc.

If it finds a real-world use-case the costs of trading need to be considered as well. As Garcia and Schweitzer [3] are saying in their research: “Trading costs can potentially erode the profitability of trading strategies, especially if they require many movements.”

The last part of research is to find a suitable programming language.

Second Phase: Development

The development phase starts with the acquisition of data followed by the calculation of sentiment about the chosen cryptocurrency. For the backend, a service is needed where the script for data acquisition can run in the background and store all the data in a database. At last, a strategy for trading is built and executed.

Third Phase: Testing, Documentation and Evaluation

While testing the strategy, the documentation is written. The last step will be evaluating the results.

Research

Coin Comparison

In March 2022, there were a total of 18000 cryptocurrencies. To minimize the number of coins for comparison, it is wise to look at the biggest coins by market capitalisation (see Figure 1). They are the most known and impactful.

coinmarketcap.com

#	Name	Price	Market Cap	Volume(24h)	Volume(30d)	Volume / Mcap	Dominance	Total Supply
1	Bitcoin BTC Buy	$39,022.52	$741,198,735,930	$26,165,152,862 669,841 BTC	$730,465,273,949 18,700,268 BTC	0.0353	42.2349%	18,975,050 BTC
2	Ethereum ETH Buy	$2,639.04	$316,615,936,335	$12,411,109,203 4,697,404 ETH	$407,080,394,217 154,073,348 ETH	0.0392	18.0302%	119,834,013.56 ETH
3	Tether USDT	$1.00	$79,726,620,750	$55,037,406,214 55,028,042,416 USDT	$6,780,571,320,951,853,056 6,779,417,707,287,214,080 USDT	0.6903	4.5402%	82,164,697,048.78 USDT
4	BNB BNB Buy	$379.61	$62,686,625,307	$1,672,738,726 4,405,999 BNB	$53,682,893,892 141,400,905 BNB	0.02668	3.572%	165,116,760.89 BNB
5	USD Coin USDC	$0.9994	$52,870,766,294	$4,004,678,076 4,006,708,689 USDC	$129,218,330,793 129,283,852,271 USDC	0.07574	3.0127%	52,897,574,957.19 USDC
6	XRP XRP	$0.7514	$36,093,181,672	$2,276,041,949 3,023,689,523 XRP	$109,308,906,281 145,215,335,289 XRP	0.06306	2.0554%	99,989,739,391 XRP
7	Terra LUNA Buy	$83.93	$31,082,599,078	$3,663,975,767 43,617,499 LUNA	$96,714,488,435 1,151,329,697 LUNA	0.1179	1.7711%	783,141,864.75 LUNA
8	Cardano ADA	$0.86	$29,020,594,167	$1,043,610,677 1,210,971,427 ADA	$29,527,798,743 34,263,084,269 ADA	0.03596	1.6526%	34,162,713,384.26 ADA

Figure 1: Snapshot from the 5th March 2022 from Coinmarketcap

Stablecoins like Tether, USD Coin or Terra are not included, since they are pegged to a FIAT-Currency like the US-Dollar. It is totally normal to trade with Currencies or Stablecoins, but you most likely loose to professionals, and you need to use leverage to even get a good arbitrage. In Figure 2 it can be seen, that the Trading-Volume (30days) of Tether is enormously more than the total circulating supply. While this is a good sign for liquidity and trading itself it is not very suitable for our Use-Case. Therefore, we will focus on the Top 5 Cryptocurrencies based on market capitalisation: Bitcoin, Ethereum's Ether, Binance Chain BNB, Ripples XRP and Cardanos ADA.

Coin	Bitcoin BTC	Ethereum ETH	Binance Chain BNB	Ripple XRP	Cardano ADA
Price ($)	39022.52	2639.04	379.61	0.7514	0.86
Marketcap (in billion $)	741	316	62	36	29
Volume (in billion $ / 24h)	26.16	12.4	1.67	2.27	1.04
Volume (in billion $ / 30days)	730	407	53	109	29
Total Circulating Supply (in million)	18,975 BTC	119,83 ETH	165 BNB	99980 XRP	34160 ADA
Volume (30days) / Total Supply (in %)	98.6	128.5	85.5	154	100
Volume / Mcap	0.0353	0.0392	0.0266	0.063	0.035
Dominance	42.2	18	3.57	2	1.65
Amount of Github Commits	32.998 (as of 18.03.22)	42.457 (as of 2021)	12.808 (as of 18.03.2022)	12.613 (as of 18.03.2022)	37.327 (as of 2021)
Launched	2008	2015	2017	2012	2017

Figure 2: Detailed Coin Comparison

Marketcap & Volume

Since Bitcoin is the pioneer and exists for more than 14 years now it is the most dominant Coin with the highest market capitalisation. It stands now at about 740 billion dollars but has already breached 1 trillion dollars in February 2021. Bitcoin reached a trillion marketcap. in only 13 years. The first company to do that, Apple, took **38 years** for that.

Ethereum is chasing Bitcoin and has already reached a dominance of 18% of the market. The Others seem quite small in comparison but when you see that they just exist for 5 years and already accumulated billions it quite impressive.

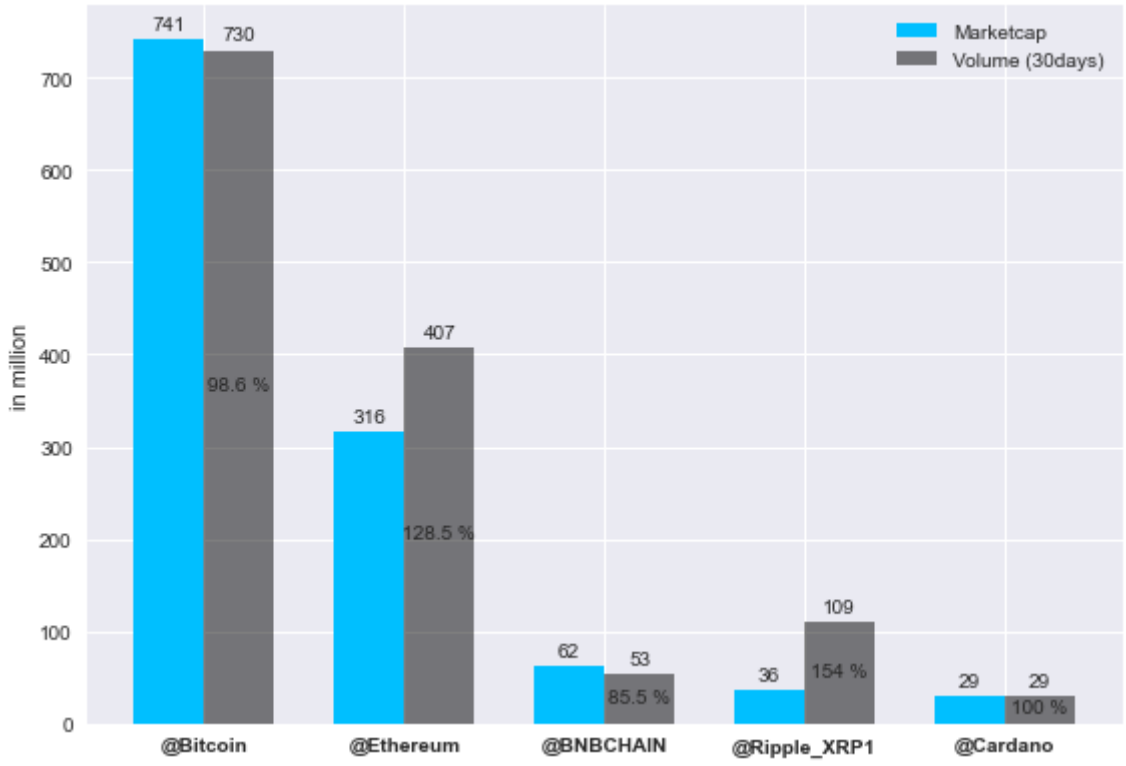


Figure 3: Market capitalisation and Volume for Top 5 coins

Adding the trading volume for the last 30 days into the comparison we can see that Ethereum and Ripple are being traded the most. At a Volume/Total Supply of 128.5% for Ethereum and 154% for Ripple.

With Bitcoin and Cardano, nearly 100% of their total supply is being traded in a month. The least monthly trading happens with BNB at only 85% of their total supply.

Crypto in that regard is more similar to FIAT-Currency than to other investments like stocks.

There is also a very big difference in the maximum Supply of Coins. For Bitcoin, there will never be more than 21 M Bitcoin. Ripple and Cardano have a hard cap as well (100 bil. and 45 bil.) and Binance initially offered 200 billion BNB but conducts a quarterly burning with the goal to have 100 billion left at the end. Ethereum, to the contrary, does not have a maximum Supply. The hard cap and the halving of Mining rewards every four years ensures that Bitcoin is deflationary by design. Ethereum handles this with burning ETH, but can still issue new ETH.

What does this mean for our trading purpose?

When there is more trading there is more liquidity in the market and therefore orders are fulfilled faster and cheaper. This would mean for Ethereum and Ripple to be a good choice for trading, but since we are not keen on trading very often very fast it really does not really matter which of these big coins we are choosing.

Social Media Relevance

The data will be collected from Social Media, and therefore we need to compare the different Social Media Platforms and then comparing the coins on the most suitable platform.

Social Media Platforms

In July 2022, there are more than 17 social media platforms with at least 300M monthly users:

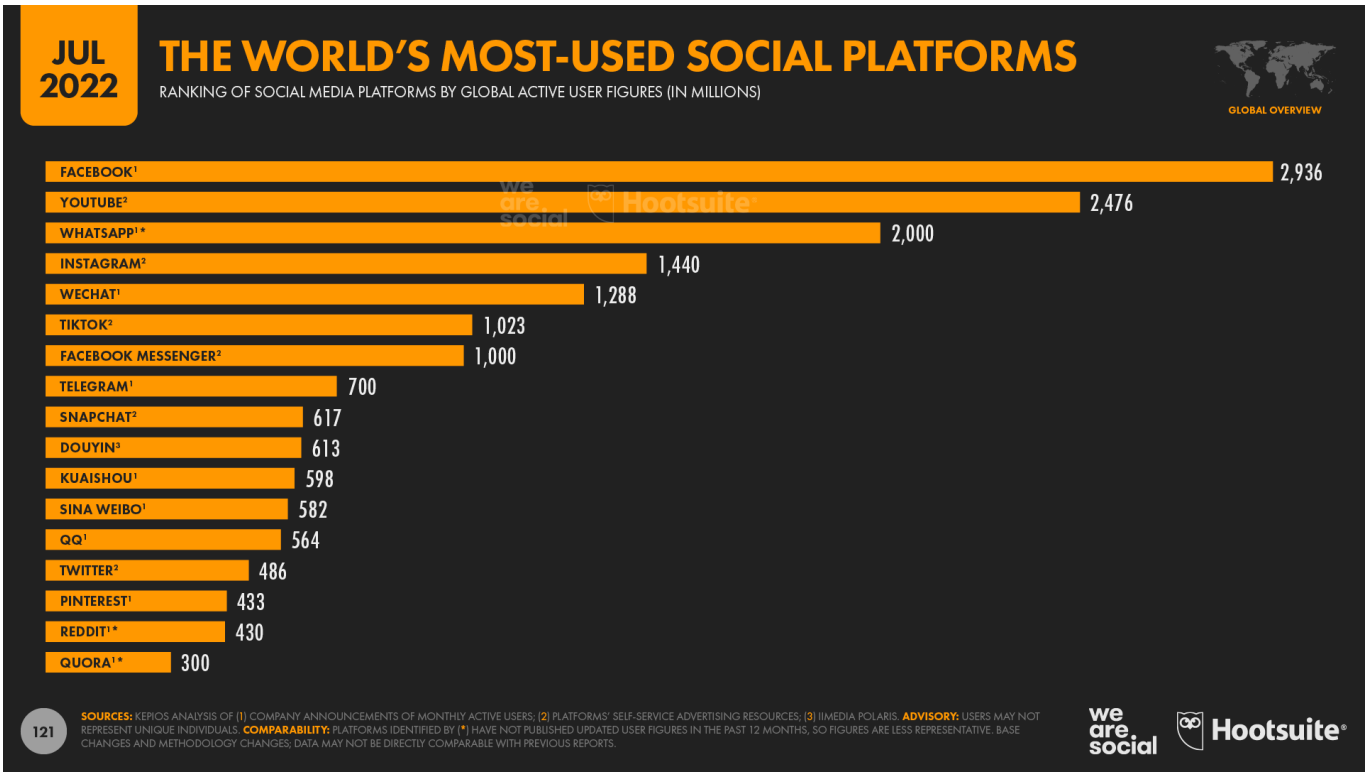


Figure 4:The worlds' most used social platforms, taken from datareportal.com

They are all very different in regard of how they work. Some are focused on video (Youtube, Tik Tok, Instagram), some for communication (Telegram, WeChat, Whatsapp), photos (Instagram, Pinterest) and some for writing (Twitter, Reddit, Quora). We are only interested in analysing and processing text, leaving us with Facebook, Twitter, Reddit and Quora.

Facebook disallows nearly every scraping or collection of data (Read this) and has shown to be a platform where a lot of complaints and hate exists. Moreover, well-known personalities, like Presidents or CEO's, are more likely to be found on Twitter.

Quora does not have a public API and is primarily used as a Q&A-Forum instead of spreading quick and fast opinions.

Reddit is a good choice for gathering sentiment and has been quite popular amongst traders. Especially since it made the headlines when the reddit members of the room r/wallstreetbets with their 4.8M members (2021, 2022: 11.8M members) came together and bought stocks like crazy.

> "Prompted by the information posted on social media retail investors began buying these so called “meme-stocks” including GameStop, AMC Entertainment, Blackberry, and Nokia. The activity sent their prices soaring, with the GameStop share prices climbing over 1000% in just two weeks." (Source)

But Reddit is also not very good for fast live data and their API and libraries are not very well documented.

Leaving us with our final choice: Twitter

A lot of Users, including well-known Personalities, who are talking about different topics in a somewhat appropriate manner in real-time. Twitter offers a well written Documentation for their API and easy-to-use

libraries for Python.

The Top 5 Coins on Twitter and Reddit

This Snapshot from 14th of March shows how different the Social Media Presence can be. In spite of being one of the youngest coins in the list, BNB has the second place in terms of total tweets and is third in total followers on Twitter.

Twitter Handle	@Bitcoin	@ethereum	@BNBCHAIN	@Ripple_XRP1	@Cardano
Twitter Followers (in million)	4.8	2.2	2.100	0.311	1.2
Amount of Tweets from their Twitter Acc.	23600.0	3273.0	6238.000	5722.000	4853
Tweets in the last hour about their hashtag (18.03.2022)	2330.0	2390.0	3320.000	6350.000	1020
Reddit Members (in million)	4.0	1.2	0.814	0.310	0.694

Figure 5: Comparison of Coins on Twitter and Reddit

In Terms of Followers on Twitter and Reddit, for obvious reasons Bitcoin is in first place. It is very interesting to see that Ripple has a very little following despite being around for 10 years.

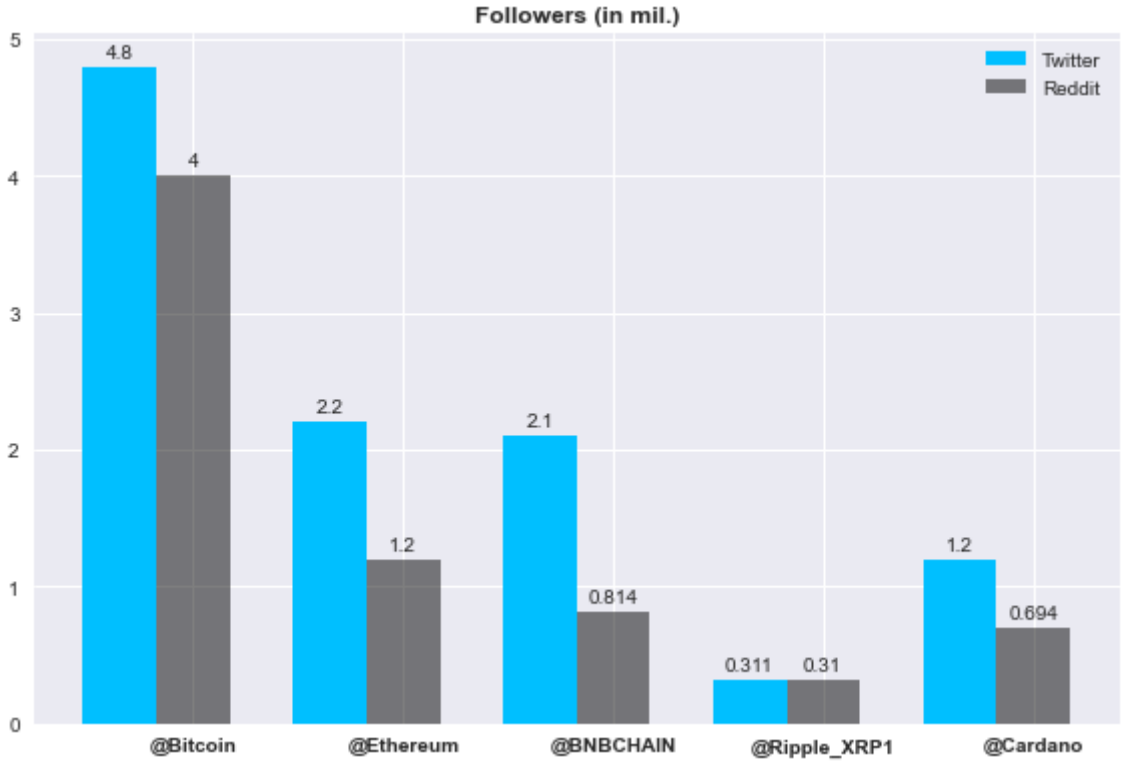


Figure 6: Visualisation of Followers

Since there is a lot of volatility and hype in the cryptocurrency world the amount of talk about one cryptocurrency can change in minutes. Twitter can show you the amount of tweets in the last hour when you search for a hashtag or cashtag. For these five Coins the range of tweets can be anything up to 8000 tweets in the last hour. This obviously depends on the timezone (CET /UTC+1). On a random day (18.03.2022) the hashtag #xrp was the strongest (6350 tweets in the last hour) and just 5 days later, on the 23rd of March, #btc trended with 7480 tweets in the last hour. At the same time #bnbchain had only 460 tweets in the last hour.

It is also quite random which hash- or cashtag will be trending and there are lots of different synonyms or variations for each Coins. #btc #bitcoin #btcusd is commonly being used to talk about Bitcoin, but this range can expand greatly. Especially when the name of the Blockchain, for example Cardano, is different than the name of their underlying Cryptocurrency, which is being called Ada. So, in case of Cardano, there are many more different hash- and cashtags: #cardano #ada $ada #cardanoada #cardanocommunity and since Cardano has one of the most Developers (see the second image -> most Github Commits), there exists a lot of talk about #buildingoncardano or #builtoncardano. All these different synonyms and variations need to be considered and evaluated to get the sentiment about one Cryptocurrency.

In case of Ethereum, the cashtag $eth is typically being more used than the #eth and is often misspelled but still trending (Etherum instead of Ethereum). As you can see in the following picture, Twitter only shows these hints about the amount of tweets when you search for this exact hashtag and even then you need to be lucky that twitter shows it. Thus, this can’t be really used as a qualified statistic to compare the sentiment about the Coins.

Focus on Bitcoin

Because there is high volatility in both the crypto market and Twitter, you can't really determine which coin is being talked about the most at any given time. Ripple has the least followers and in total not a lot of talk about it.

Building a modular model in a way that it can retrieve tweets about any coin was a likely goal at first, but since the database storage will be limited, it is wise to primarily concentrate on Bitcoin since it is the one that has started the whole cryptocurrency era and the most followers and talk about it.

Choosing a programming language

Python was the chosen the programming language for this project. Not only had the developer a good amount of experience with it, further more offers Python the needed libraries to analyse and visualise data in a way that allows for rapid prototyping.

Requirements Analysis

To access the requirements, one normally would use a stakeholder analysis.

Since this project is more like a research project, it is not really intended to be used as a product outside of Chainsulting. Thus, the primary and only stakeholder is one of the CEO's of Chainsulting, Yannik Heinze, who is keen on finding out about the effects of sentiment on trading performance.

In consultation with Mr. Heinze, the requirements for this project have been set beforehand.

The requirements are structured as functional and non-functional with three categories:

- *Must-Have*-Requirements are essential for the system to work at all.
- *Should-Have*-Requirements are improving the system, but are not necessary for the system to run.
- *Optional* Requirements can improve the final product, but not at all necessary.

Functional Requirements

Must-Have

Nr.	Description
[FR 10]	The system delivers real-time tweets from Twitter
[FR 20]	The system delivers a sentiment for each tweet
[FR 30]	The system builds trade-signals based on sentiment
[FR 40]	The system trades based on these signals

Should-Have

Nr.	Description
[FR 50]	The system filters tweets
[FR 60]	The system visualises metrics of tweets, sentiment and trades
[FR 70]	The system stores the data in a database
[FR 80]	The system can run in the background

Optional

Nr.	Description
[FR 90]	The system uses different trading strategies at the same time
[FR 100]	The system uses Machine-Learning to improve the strategy

---

Non-Functional Requirements

Must-Have

Nr.	Description
[NFR 10]	The project must contain research on used cryptocurrencies and social-media platform
[NFR 20]	The project will be published open-sourced on GitHub

---

Steps for Development Phase

From the above requirements analysis, a flow diagram was derived to simplify and visualise the order of steps that are needed to be taken in the development phase. This helped to split the available time into phases for each part in this diagram. It was basically the key to time-management of this project. The orange bigger box are the parent steps, in which the smaller steps take place.

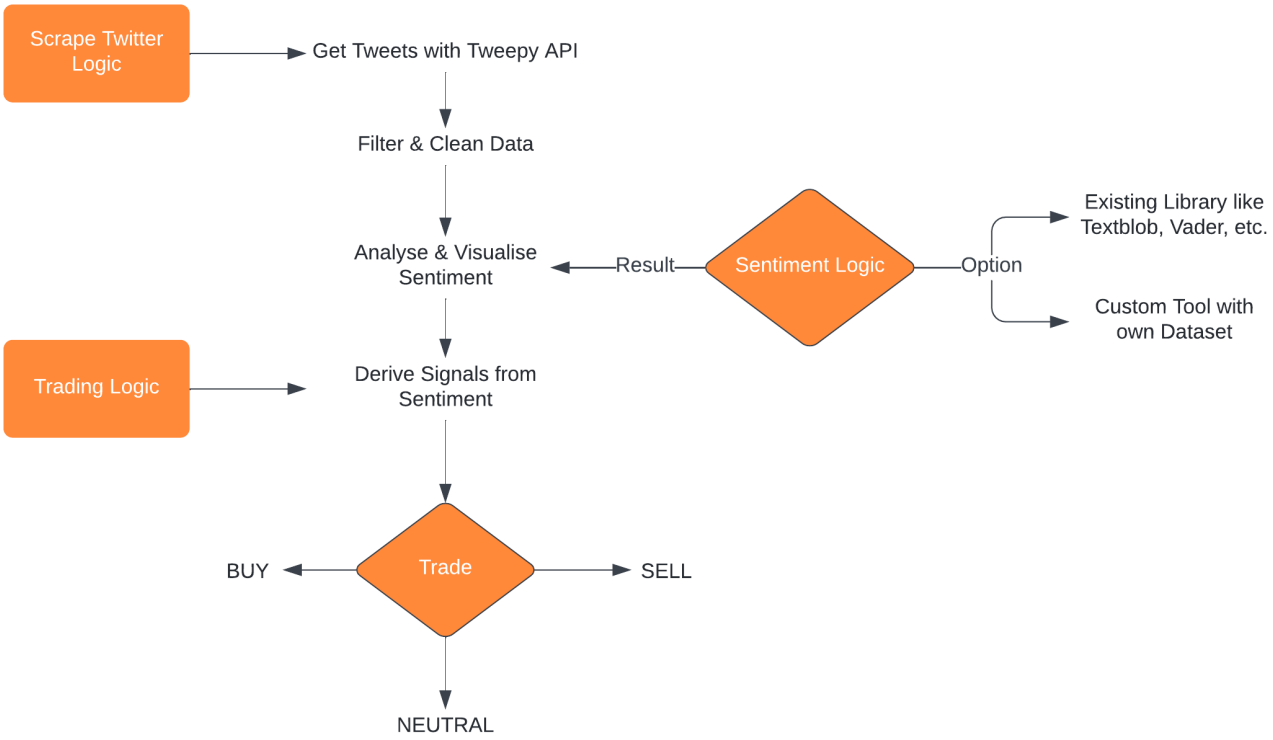


Figure 7: Flow Diagram for Development Phase

Data Acquisition

Retrieving and Processing of Tweets with the Twitter API

The development phase starts with the acquisition of data from Twitter, as being stated in the *Must-Have*-Requirements (compare [FR 10]). Instead of dealing with HTTP Requests, Data Serialisation and Rate Limits it is easier to use pre-built libraries to access the Twitter API and be able to focus more on building functionality. There are hundreds of different libraries, but I've found two to be standing out and compared them: Twint and Tweepy.

Library	Twint	Tweepy
Twitter API	No authentification with official Twitter API necessary.	Auth. with Twitter API needed
Limits	No limits on search for last tweets	Search tweets as early as 2006
Development	12,7k Stars on Github with 845 commits (last Commit in March 2021)	8,6k Stars on Github with 2914 commits (last Commit: 25.03.2022)
Documentation	Minimal, Not many Tutorials	Big, updated and easy to understand documentation and a lot of tutorials
Purpose	Good for grabbing Tweets from the past, not for posting or sending DM's	Everything what the Twitter API allows: From Scraping to Tweeting to Sending DM's

Twint seems to have the upper hand because it does not need an authentication and has good filters for tweets (even Cashtag filter). Acquiring some tweets with Twint was done in a few lines of code, but a few days later this didn't work any longer. It appears that Twint wasn't retrieving any data any longer. This could've been a temporary problem and may work later on, but since their GitHub hasn't been updated in more than a year and others had this issue as well, Twint no longer seemed as a good option.

Getting authorized from Twitter was easy. Login to the Twitter Developer Portal and create a new Project and App. There you can generate and copy the necessary token: API Key, API Secret, Access Token and Access Secret. The Twitter API also released a Version 2 which is able to authorize using only a bearer token.

Authorization and searching for Tweets with Tweepy is simple:

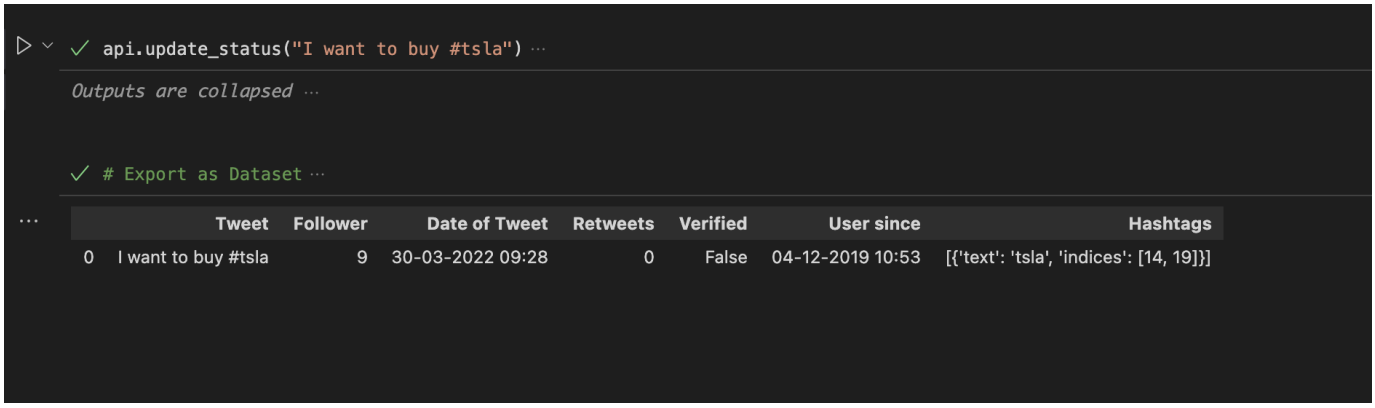
```
import tweepy
auth = tweepy.OAuth1UserHandler(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True)
posts_for_elon_musk = api.user_timeline(screen_name="elonmusk",
count=5,tweet_mode = "extended")
print(posts_for_elon_musk)
```

Output:

> **Elon Musk's latest Tweets:**
>
> 1. The ratio of digital to biological compute is growing fast. Worth tracking.
> 2. Just came across this pretty good CNBC piece on SpaceX & Starship https://t.co/RELYzC40M9

> 3. Tesla + Twitter = Twizzler
> 4. A new philosophy of the future is needed. I believe it should be curiosity about the Universe – expand humanity to become a multiplanet, then interstellar, species to see what's out there.
> 5. The media is a click-seeking machine dressed up as a truth-seeking machine

A first test with Tweepy was a simple tweet with the authors account was done with only one line of code:



	Tweet	Follower	Date of Tweet	Retweets	Verified	User since	Hashtags
0	I want to buy #tsla	9	30-03-2022 09:28	0	False	04-12-2019 10:53	[{'text': 'tsla', 'indices': [14, 19]}]

Figure 8: Tweet Status with Tweepy

Figure 9: Tweet on Twitter

What does the Law say?

The EU General Data Protection Regulation, or GDPR only applies to personal data.

Here, "personal data" refers to the data that could be used to directly or indirectly identify a specific individual. This kind of information is known as Personally Identifiable Information (PII), which includes a person's name, physical

address, email address, phone number, IP address, date of birth, employment info and even video/audio recording. If you aren't scraping personal data, then GDPR does not apply.

If you are not scraping in the EU you are good to go. (Source)

Retrieving Tweets with Tweepy

Using the Stream Class from Tweepy allows filtering and sampling of realtime Tweets with the Twitter API.

A Listener for certain keywords is initiated and every time a tweet contains these keywords, it is collected.

This default dictionary contains the keywords that filter the tweets for each coin:

```
default_keyword_dict = {
            "btc":["$btc","#btc","bitcoin","#bitcoin"],
            "ada":["#ada","$ada","cardano"],
            "eth":["#eth","$eth","ether","ethereum","etherum"],
            "bnb":["#bnb","$bnb","binance coin"],
            "xrp":["#xrp","$xrp","ripple"]}
```

The keyword class builds a list of keywords from above dictionary and combines the single lists, so it is possible to filter for multiple coins at the same time.

It also contains the method to search through each word in the tweet for the keywords.

Tweet Metrics

I am not only collecting the tweet but some other information/metrics about the tweet or the user:

Timestamp	id	Tweet	Keyword	Location	User verified	Followers	User created	Sentiment Score
2022-08-04 11:54:37	25708	$btc support holding for now	$btc	Egypt Lake-Leto, FL	False	972	2021-11-27 14:20:42	0.4019
2022-08-04 11:54:34	25707	scalpers its very simple #bitcoinwhat are you betting on	bitcoin	Everywhere	False	36598	2016-08-02 14:31:10	0.0
2022-08-04 11:54:34	25706	he need to focus on his #bitcoin investment	bitcoin	Ivory Coast	False	786	2020-06-04 10:55:58	0.0

Timestamp	id	Tweet	Keyword	Location	User verified	Followers	User created	Sentiment Score
2022-08-04 11:54:25	25704	who wants a #bitcoin update video hit up that like button	bitcoin	Magic Internet Bank	False	3810	2021-02-11 15:29:52	0.3612
2022-08-04 11:54:15	25703	picture perfectthe $4250 resistance will be hard to break thoseeing a small retracement from hereif $btc $eth behave and hopefully pump we may see another ~50% pump from here for $metis as well to around $65	$btc		False	505	2022-03-14 15:14:53	0.7964

Using the Location to filter for different timezones, so it is possible to enable the bot at different places at different times. It would be more efficient to collect tweets p.e. from America when most people are awake and able to tweet and not asleep. The problem is that the Location is not automatically read and set by Twitter, but rather manually set by the user. This means it is nearly useless to filter it since some tweets are from *"Everywhere"* or from the *"Magic Internet Bank"*, as can be seen in the table above.

For the tweet the timestamp, the found keyword and the calculated sentiment are included. More about sentiment in this section.

The information about the user includes their follower base, when the user was created and if they are verified. No personal information with which one can conclude the identity of the person is acquired.

These metrics are collected to filter the tweets, so we have more high quality data at hand.

Filter Tweets

The data needs to be as significant as it can be, which means Tweets from Bots should not be included since they are deflecting the overall opinion by spreading the **exact** same opinion/tweet repeatedly. Applying filters is increasing the quality of the final product [FR 50].

But what differentiates a tweet by a bot from a tweet by a human?

1. Bots often retweet
2. Bot Accounts are created quite often thus fairly new to the platform
3. Bots (especially when new) have little following
4. They often offer giveaways or free coins
5. They repeat themselves... a lot

Accounts with less than 500 followers or when they have just been created in the last two months are filtered out. Also Retweets and Tweets that contain blacklisted Words like *Giveaway, Free or Gift*.

The next step is cleaning the tweet itself. Most often a tweet can contain a lot of characters like *underscores, hashtags or emojis* and special characters like *"&amp"* (which is used in Html entities for a normal "&"-Symbol. There was one in one of Elon Musks Tweet above).

With the help of regex functions it can be done to filter and substitute words or symbols that are not useful. This is an excerpt from the `cleanTweets`-Function in the filter.py file:

```
text = re.sub(r'@[A-Za-z0-9]+',"",text,flags=re.IGNORECASE)
text = re.sub(r'_*|\+*',"",text) # removes _ and +
text = re.sub(r'&amp;*|&amp|amp',"",text)
text = demoji.replace(text, "")
cleaned_words = [x for x in words if not bool(re.search('^[0-9]+$|^$', x))]
```

With `re.sub` we can substitute the text we don't want. The first three lines remove the "@", "_", "+" and different variations of "amp" (You can find it in the second Tweet from Elon Musk in the output above) All these are substituted with empty strings which are being removed in the last line. The last line also cleans alone standing numbers like "734982" but leaves numbers with characters in the string like "$20k".

As an example, this is a tweet before cleaning:

> "_boii &amp people think is high now wait till it’s $20k + by the end of this week $50k+ by end of 734982 this month y’all should follow he's a super underrated !bitcoiner i’ve been :following her tweets and tips seriously i’ve been doing really great! #btc #ada"

After cleaning, it looks like this:

> "boii people think is high now wait till it’s $20k by the end of this week $50k by end of this month y’all should follow hes a super underrated bitcoiner i’ve been following her tweets and tips seriously i’ve been doing really great"

Duplicates

The next thing is to find and delete all the duplicates. As can be seen in Figure 10, the same tweet was tweeted 10 times in one minute. It is the exact same tweet, and it kept on tweeting for quite a bit. Since this is obviously a bot, a way to filter out these duplicates was needed.

id	body	keyword	tweet_date	location	verified_user	followers	user_since	sentiment
510785	yesterday at commission i passed a unanimous resolution directing the urgent repair and opening of the final piec	#ada	2022-07-29 11:53:02	Miami, FL	FALSE	16765	2015-02-01 05:57:29	0.2023
510784	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:53:02	Bogalusa, LA	FALSE	1467	2022-04-17 12:11:36	0.8221
510783	$wampl should 3x like $forth $amc $gme $btc $eth $xrp $doge $icp $ect $storj $fil $mana $bond $time $muse $	$btc	2022-07-29 11:52:59	Penn Hills, PA	FALSE	2738	2009-03-06 03:40:00	0.3612
510782	_israar __ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all sh	#btc	2022-07-29 11:52:56	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510781	black ppl learned absolutely nothing from 08 housing crisis bitcoin bout to leave alot of us unhoused and pennies	bitcoin	2022-07-29 11:52:55	The holiest ground in NYC	FALSE	1597	2017-01-18 17:43:50	-0.7964
510780	_binance hey if you help us burn we will help you build community is better than	#btc	2022-07-29 11:52:50	المملكة العربية السعودية	FALSE	2333	2020-07-26 16:26:11	0.8074
510779	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:52:48	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510778	don't open coinbase check the price of on	bitcoin	2022-07-29 11:52:44	Washington, DC	FALSE	1166	2014-03-08 17:02:23	0.0
510777	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:52:43	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510776	coins strength pt vs $btc	$btc	2022-07-29 11:52:37	Not financial advice #DYOR	FALSE	2445	2020-12-12 17:21:57	0.4939
510775	_lordness __ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all	#btc	2022-07-29 11:52:37	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510774	and 70 others 43 28 10781억원 $9225017 ¥1013260683 €7807860 59554182元upflow volume now upbitkorea	#btc	2022-07-29 11:52:35	SEOUL	FALSE	1612	2021-08-18 18:22:20	-128
510773	bitcoin rising back to the above price gap to $25000 all trading desks to exit sell positions and wait higher $btc	$btc	2022-07-29 11:52:32	None	FALSE	1812	2019-08-25 07:16:26	0.0
510772	cardano vasil hard fork hit with another delay for several weeks / / follow _nagar for more nft news	cardano	2022-07-29 11:52:25	NJ	FALSE	1481	2021-01-23 00:23:37	-0.4019
510771	_u_f_s __ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all sh	#btc	2022-07-29 11:52:24	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510770	bitcoin holds k as usd taps 3week lows on eurozone inflation report / / follow _nagar for more nft news	bitcoin	2022-07-29 11:52:20	NJ	FALSE	1481	2021-01-23 00:23:37	-0.2023
510769	_israar __ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all sh	#btc	2022-07-29 11:52:18	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510768	cardanos vasil hard fork erneut verzögert ögert _news	cardano	2022-07-29 11:52:17	Stuttgart	FALSE	1168	2021-07-04 14:20:03	-0.2023
510767	just in cardano's vasil hard fork delayed again	#btc	2022-07-29 11:52:16	Cheers in the Moon	FALSE	7099	2021-05-16 14:37:55	-0.3182
510766	coins strength lp vs $btc	$btc	2022-07-29 11:52:16	Not financial advice #DYOR	FALSE	2445	2020-12-12 17:21:57	0.4939
510765	crypto exchange zipmex goes bankrupt – who's next to fall $btc $eth	$btc	2022-07-29 11:52:16	NYC	FALSE	561	2019-09-03 13:13:20	-0.5574
510764	just in price analysis july29 coin and	#btc	2022-07-29 11:52:15	Cheers in the Moon	FALSE	7099	2021-05-16 14:37:55	0.0
510763	precio actual $btc = u$s 2365668	$btc	2022-07-29 11:52:10	Worldwide	FALSE	9123	2018-03-28 18:10:44	0.0
510762	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:52:10	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510761	and 70 others 48 23 10781억원 $9225488 ¥1013312446 €7808259 59557224元upflow volume now upbitkorea	#btc	2022-07-29 11:52:07	SEOUL	FALSE	1613	2021-08-18 18:22:20	-128
510760	a great super amazing project looking very promisingthe project has a good development long term opportunity fo	#btc	2022-07-29 11:52:07	None	FALSE	2257	2021-04-08 07:11:21	0.9552
510759	if we do keep rallying my next targets above my 3rd seen here is 4200 if 4200 is broken i think we will test the 200	#btc	2022-07-29 11:52:06	None	FALSE	514	2021-01-31 22:37:26	0.1857
510758	bitcoin price predictionprice fall in 60 mintarget price 2324261 usdtsell 0126 btc for 2371695 usdt apiece on bina	#btc	2022-07-29 11:52:06	None	FALSE	2723	2020-05-24 06:50:59	0.0
510757	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:52:06	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221
510756	as the corporation works to remedy its financial concerns it has requested that the moratorium be lifted for five of i	#btc	2022-07-29 11:52:05	UK	FALSE	47210	2014-07-18 12:05:03	0.0
510755	[$btc] bitcoin crypto exchange zipmex goes bankrupt – who's next to fall	$btc	2022-07-29 11:52:02	None	FALSE	2865	2021-11-26 23:57:13	-0.5574
510754	__ people think is high now wait till it's $20k + by the end of this week $50k+ by end of this month y'all should fo	#btc	2022-07-29 11:52:02	Bogalusa, LA	FALSE	1468	2022-04-17 12:11:36	0.8221

Figure 10: Duplicate Tweets in a 1 Min. time period

The first approach was as follows: Adding the collected tweets to a list, and after the list had 40 items, run a function that checks for duplicates and deletes them. There were about 40 collected items a minute, so it seemed like a good choice.

The following function converts the list to a Pandas Dataframe and checks for duplicate Tweets with their innate `duplicated()`-method. The `keep=False`-parameter deletes the found duplicates.

```
def check_duplicates(tweet_list):
    cols = ["Tweet", "Keyword", "Time",
            "Location","Verified","Followers",
            "User created", "Sentiment Score"]
    df = pd.DataFrame(tweet_list,columns=cols)
    duplicates = list(df.index[df.duplicated(subset=["Tweet"],keep=False)])
    df.drop(labels=duplicates,inplace=True)
    return df.values.tolist()
```

Unfortunately this lead to the following problem:

Firstly, even though this function found and deleted some duplicates nearly every time it was called, checking only 40 tweets at a time is not sufficient to delete **all** duplicates.

When applying this function to the whole database, it deleted 17318 duplicates from a total of 32698. More than 50% duplicate Tweets is massive! This would've deflected the calculation of the sentiment significantly. This lead to the decision to apply the deletion of duplicates for the entire database before calculation of the sentiment.

Backend

Reason for Heroku

Tweepy uses two ways of searching for tweets: Searching through the history or listening to real-time tweets. One approach would be to schedule a task, p.e. for every hour, that searches the history of tweets and does the trading afterwards. However, after looking closer into the Documentation, it was discovered that the search service with the Twitter API doesn't show all available tweets. This would lead to incomplete data, which would deflect the strategy and therefore wasn't an option.

When listening to real-time tweets, the python script needs to run the whole time in the background ([FR 80]). It is no option to only run it on the local machine when it is intended to run for days or weeks. When the laptop is closed, the script stops running.

That is where cloud platforms as services come in handy. Today, most of the SaaS-products run in some sort of cloud since it is not very desirable for smaller companies to run and maintain servers for their clients when other bigger corporations such as Google, Amazon or Microsoft offer better, more scalable and cheaper services.

For this project, Heroku was used as the cloud provider, because it works well with Python, has a free tier available and an easy-to-use Postgres Database Add-On ([FR 70]) and GitHub Continuous Integration.

Heroku works with containerization: It automatically detects a change in the GitHub Repository and starts a new Deployment. Because this did not always work, it is still possible to manually deploy in the Heroku Dashboard or with the Heroku CLI.

Heroku will then package the code and dependencies and runs it in lightweight, scalable and isolated containers, which are called *dynos*. In the free tier you have free 550 dynos each month and every time a web service is called, or a script (worker) is run a dyno is consumed.

Using Heroku allows to run the script in the background, store data in a database and schedule trades.

Setting up Heroku

Creating an App in the Heroku Dashboard and initializing it with our codebase in VSCode was the first thing to do. The easiest way to do this is by connecting the Heroku App with the GitHub Repository. But since we were using the Heroku Account from Chainsulting and are not able to access their GitHub account, it was not possible to use GitHub Continuous Integration. Instead, manually deploy every change of code from the terminal with `git push heroku main`.

Secondly, the Heroku Postgres Add-On was added and credentials were stored inside the Config Variables of the Heroku Dashboard.

Our connection with the Database from our python code is established with psycopg2 and SQLAlchemy. Psycopg2 reads the data and SQLAlchemy is responsible for creating a session and then commits the data to the database.

The following example shows how to connect to the database and read the data into a Pandas DataFrame by using a simple SQL Statement.

```
import os
import psycopg2

DATABASE_URL = os.environ['DATABASE_URL']

conn = psycopg2.connect(DATABASE_URL, sslmode='require')

query = f"select * from tweet_data order by id desc limit 10;"

df = pd.read_sql(query, conn)
```

Heroku uses a so called Procfile to start the processes. This meant the worker script for listening to tweets and adding to the database is added to this file:

```
worker: python3 sentiment/runner.py
```

The Heroku Scheduler Add-On is used to execute the trading script every hour. It checks if there are trades for a past timeframe and either buys, sells or does nothing if there is a trade already. More on the trading part here.

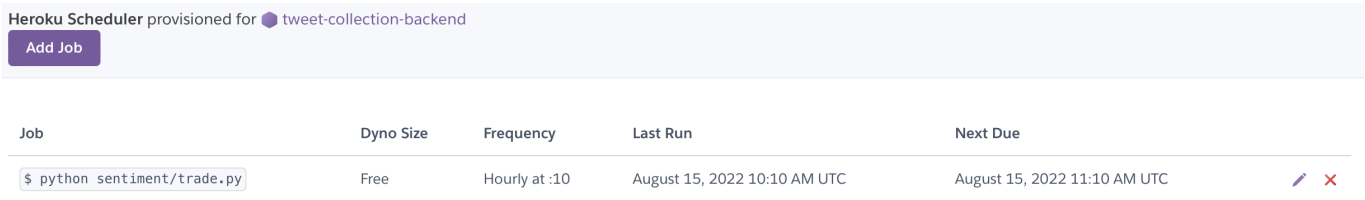


Figure 11: Scheduler Add-On on Heroku

Checking the Databases

Heroku Dataclips offers to run SQL queries and view the output directly in the Browser. It was a very valuable tool to check quickly if a SQL Statement is executed successfully on the real database and if

everything works smoothly in the production phase.

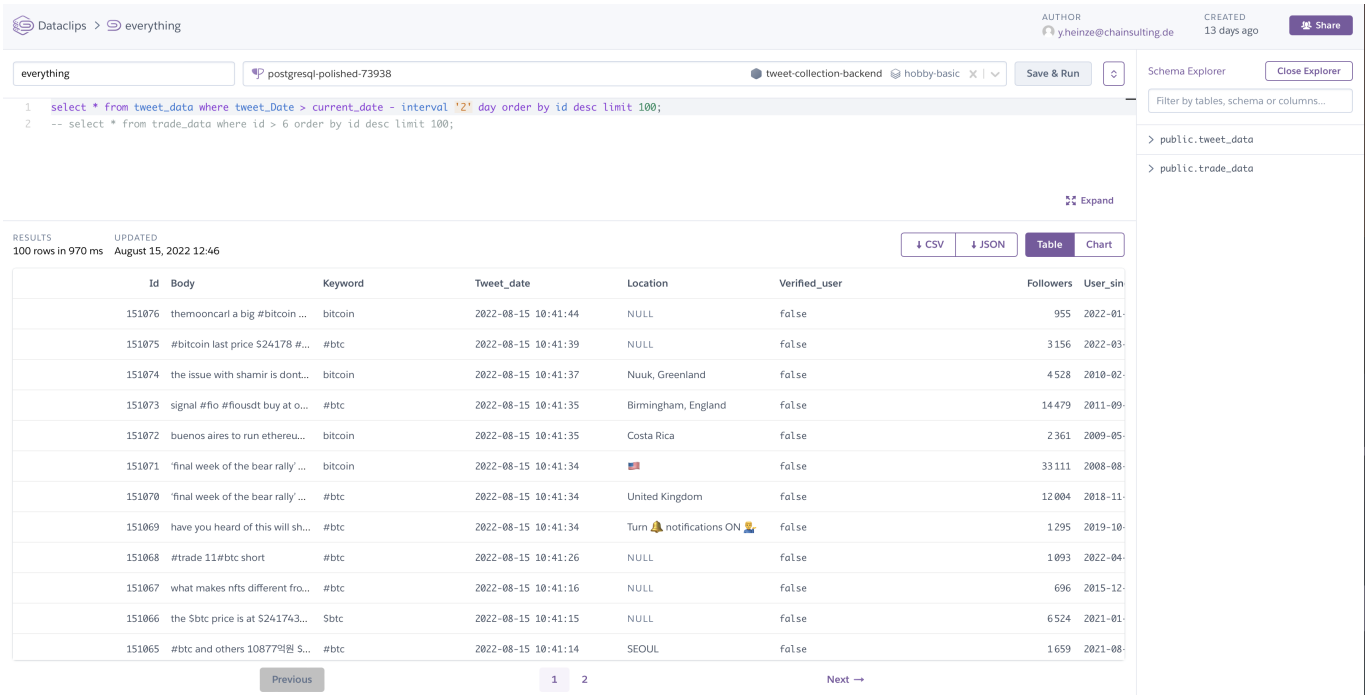


Figure 12: Dataclip to view the Tweet database with SQL queries

---

Challenges and Solutions

Heroku was an essential tool for this project, but revealed a lot of challenges during configuration. Below are a few challenges and how there were overcome.

No such Procfile

The first problem appeared directly after initializing it with Git and deploying the first time with GitHub Continuous Integration in my own GitHub repository. Although, the Procfile was in the main directory, Heroku could not find it. Weirdly enough, it was fixed after deploying it manually and setting the correct python buildpacks via the Heroku CLI with the following commands:

```
heroku buildpacks:set heroku/python`
git push heroku main
```

Connection with Postgres Database

The Postgres Database revealed many problems. You can connect to the Database either via Database URL or via Username, Password and Hostname. Since the latter worked perfectly when connecting to other

databases before, this was the first approach, but the password authentication always failed. After a long while of debugging and going through StackOverflow, nothing worked. At random, the connection way was switched to the Database URL and after fixing two further problems the connection was finally established. The URL just needed to be changed from `"postgres://..."` to `"postgresql://.."` and the connection required SSL, which was easily set by `heroku config:set PGSSLMODE=require``

Postgres Row Limit of 10000

The Free Hobby Plan for Heroku Postgres Database has a limit of 10000 rows, and it got filled very quickly. The database was filled in a few hours. But nothing happened. A week passed, and the database was filling up until I got an email from Heroku:

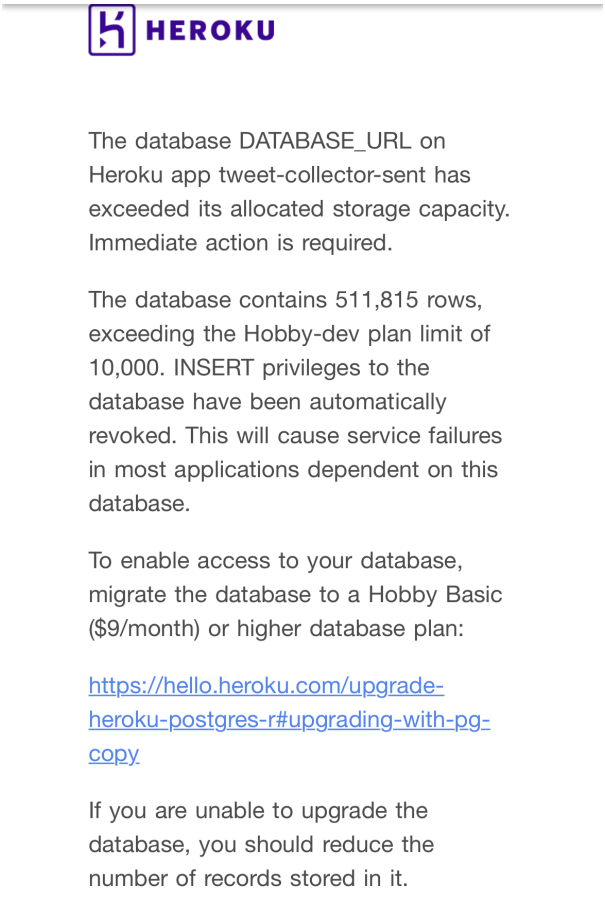


Figure 13: Email from Heroku: Database Row Limit reached

The Heroku Database got filled such quickly because the system was collecting Tweets about Bitcoin **and** Cardano. It would be very interesting to compare different coins side by side, but it made sense to only look at Bitcoin after this point to gather more Tweets. In the end this was a waste of time because Chainsulting was so kind and upgraded the Heroku Database Plan to a row limit of 10M, which should last for about a month of Tweets. Of course, this can be improved by better filters and periodically deleting tweets from the database. Unfortunately, when the Database is full, Heroku revokes *Writing-Privileges*, which makes it even impossible to delete single rows via a dataclip. Leaving the only solution to clear the whole database, which has been done a few times before the upgrade.

Sentiment

The goal is to find out if trading cryptocurrency based on the overall opinion about a cryptocurrency on Twitter is able to make profits.

Therefore, we need to find the opinions and metrics to objectively compare these opinions from People about a particular topic, in this case Bitcoin. This is where *sentiment analysis* comes in handy.

Sentiment analysis is a part of *Natural Language Processing (NLP)* to systematically identify and classify opinions from text. For every sentence we say or word we use, we kind of know if it has a positive or negative meaning. A program can't do that. It needs validated values for each word, saying if it's positive or negative, to calculate the overall opinion in a sentence. This is why a lot of teams around the world have built lexicons with thousands of words in it and rated each of them with a value to either be positive or negative.

There are a lot of different tools that apply these lexicons for NLP for different use cases, such as calculating and classifying sentiment polarity (positive or negative) from text-input.

Two of the primarily used tools for NLP for python are compared in the next section.

Tools for Natural Language Processing

TextBlob

TextBlob returns polarity and subjectivity of a sentence. Polarity lies between -1 and 1, defining negative and positive sentiment, respectively.

Subjectivity lies between [0,1]. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information. TextBlob has one more parameter — intensity. TextBlob calculates subjectivity by looking at the ‘intensity’. Intensity determines if a word modifies the next word. For English, adverbs are used as modifiers (‘very good’).

Vader

Vader (Valence Aware Dictionary and sEntiment Reasoner) does the same thing as TextBlob, but it is especially good for social media sentiment analysis, because it uses some heuristics to improve their calculations. They not only calculate the polarity (positive or negative), but also the intensity/valence of each sentiment on a scale from -4 to +4. Meaning, that the word "Good" has an intensity of 1.9, the word "Great" of 3.1. This value, coming from observation and experience, will be taken into account for the calculation. They also use intensifiers or degree modifiers to either increase or decrease the intensity. This

is more accurate than models who only use a lexicon of words. They also understand emoticons like "😄" or emojis and many slang words like "kinda" (instead of "kind of").

Punctuation ("Good!!!!"), Word Shape ("ALL CAPS WORDS"), Intensifiers ("It is **extremely** hot"), Negation("The weather isn't really that hot") all shift the sentiment intensity.

Comparison

Getting the polarity from a sentence with these tools only requires a few lines of code.

```
from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

text = TextBlob("Just came across this pretty good CNBC piece on SpaceX &
Starship")
print(f"TextBlob: {text.sentiment.polarity}")

vader = SentimentIntensityAnalyzer().polarity_scores(text).get("compound")
print(f"Vader: {vader}")
```

Output:

> TextBlob: 0.475
> Vader: 0.7684

As can be seen the difference is quite huge between those two values. The sentiment scores for the dataset of tweets were quite similar from both TextBlob and Vader. Considering, that Vader was built for social media sentiment analysis and uses special heuristics to improve the overall performance, compared to other NLP-Tools, Vader will be used to get the sentiment for each tweet in this project ([FR 20]).

Vader gives sentiment score between -1 and 1 for each tweet, normally meaning, that positive values mean a positive sentiment and negative values mean a negative sentiment. To differentiate even further, the decision has been made to classify values above 0.6 to be *"Very Positive"* and values below -0.6 to be '"Very Negative"*.

The final classification:

Sentiment Score	Meaning
> -1 & < - 0.5	Very Negative
>= - 0.5	Negative
0.0	Neutral
<= 0.5	Positive
> 0.5 & < 1	Very Positive

More Filters

To make better trading decisions some more filters and data cleaning after reading the tweet data from the database is needed.

Since sentiment analysis libraries are obviously not able to declare a sentiment polarity to all the words in the world, it tends to classify a lot of the words and thus their tweets as neutral. When going through the acquired database, there were a lot of *neutral* tweets. This would decrease the impact of positive and negative tweets in the calcuation and since most of the tweets were not really useful tweets anyway, they were excluded. Many neutral tweets contained bad grammar, random words, only hashtags or a lot of gibberish. Filtering out neutral tweets was a necessary step and is achieved with just one line of code See here.

With these sentiment values and their meaning at hand some more calculations will be made to build a strategy for signals, when a buy or sell order is made.

Trading

Choosing an API

There are a lot of exchanges for trading cryptocurrencies. Which all have pros and so leads to a comparison of APIs with respect to this project's use case. It is important that the API has a good and easy documentation and an extensive python library/wrapper.

The following table shows the full comparison of five cryptocurrency API's, which was made using this Article:

Platform	Binance	CoinBase	Kucoin	Coin
Pros	Directly sell and buy with the API, Connector to diff. Languages (python, Ruby), Commission when using BNB coin, 1200 free Requests per Minute, Binance has one of the biggest markets	API can be used as an Exchange or Wallet, Real-Time Notifications for Account Events	Many currencies(>200), Telegram Group for Support, 1800 Free Requests per MinuteCrypto Lending available	Gooc langu 11 er
Cons	There have been outages in the past. When exceeding the transaction limits, the IP will be temporarily banned	Only etc, eth, btc cash, litecoin, Only 180 Requests per Minute	Performance issues (payments & trades hold back)	No h Pay histc 60 R Not
Historical Data	Yes	Yes	Yes	No
Fees	0.1%, but when using BNB can be lowered to 0.075%	1%	3-5%, when buying with FIAT 0.1% for crypto, 0.08% when buying with Kucoin Token	not f
Rating of Documentation	Very good	Good	Very good	Okay
Availability of Python Libraries/Wrappers	Very good and updated. https://github.com/sammchardy/python-binancehttps://github.com/binance/binance-connector-python	Last updated 8 years ago. https://github.com/resy/coinbase_python33	Very good and updated. https://github.com/sammchardy/python-kucoinhttps://github.com/Kucoin/kucoin-python-sdk	Not https coin

When it came to the decision which API to use, it was quite easy to filter out the ones that weren't suitable. CoinBase was the first to drop because it has only five coins and no currently active python libraries available.

Coinmarketcap was also not very intriguing because it has limited Endpoints and no historical Data. Paying 29$ a month for 1 month of historical data didn't seem worth it when others offer it for free. Their python wrapper was also not very detailed.

I personally use Kraken, so the ID Verification wouldn't even be a problem, but they do not offer a test/sandbox environment. Their way of working with request limits is also very different to the others and seemed to be a limiting factor. But the biggest problem with Kraken was that there is no recently updated python wrapper available.

Kucoin offers many currencies and the most free requests per minute, but reported some performance issues, which lead to the decision to Binance. At least at first.

Binance has one of the biggest markets with the highest liquidity and small fees. They have a very good documentation and an updated python wrapper with lots of tutorials. The only thing to keep in mind is the potential ban of my IP when the request limit is reached.

The Problem with Binance

The verification process was long and even required to send some Bitcoin to the Binance Wallet. Afterwards, it was possible to create the API-Keys, which are needed to connect the Client with Binance. Getting live price data from the Binance API was working perfectly, so everything seemed fine. Visualisations of Charts were made to have a better overview and make a small simulation to test the strategy (see below).

Then it was time to implement the trading strategy into the Binance sandbox and the problem appeared. The Sandbox Testnet didn't accept the API-Keys and even after a while of debugging the error, it wouldn't work, leaving us with no choice but to drop Binance for Kucoin. Setting up Kucoin was faster and their sandbox worked perfectly.

Kucoin API

After registration you need to go through a verification process. Typically, this is a call with a person that verifies your identity or uploading a photo of a personal ID-Card, but Kucoin only needs verification via Google Authenticator.

After that the API-Keys could be generated. These Keys are needed to connect and verify the Client with the API.

The *python-kucoin* wrapper helps with the Kucoin API, so there is no need to manually connect with the API via HTTP-Requests.

Acquiring data from the Kucoin API is fairly easy. Getting the latest price for BTCUSDT, the account balance, and making a market order are all done with a few lines of code. As can be seen below in the output, the current USDT Balance in the account is over 48k USDT. Kucoin offers every account a virtual amount of 250k USDT (~11 BTC) to trade with in their sandbox environment. They also offer Sub-Accounts, which is great to test a strategy for a specific amount of money, without risking to lose it all, when something goes wrong. This meant, papertrading could be performed and evaluated.

```
from kucoin.client import Client as kucoinClient

kClient = kucoinClient(kconf.KUCOIN_KEY, kconf.KUCOIN_SECRET,kconf.KUCOIN_PASS,sandbox=True)

tickers = kClient.get_ticker(symbol="BTC-USDT")
btc_price = tickers["bestAsk"]
print(f"Current BTC Price: {btc_price}")

accounts = kClient.get_accounts(account_type="trade")
usdt_balance = accounts[0]["balance"] #for Main Client
btc_balance = accounts[1]["balance"] #for Main client
btc_in_usdt = float(btc_balance) * float(btc_price)

print(f"USDT Balance: {usdt_balance} $")
print(f"BTC Balance: {btc_balance} ({btc_in_usdt} $)")

# Make a Market Order for 5% of the current USDT Balance
funds = re.match(r'\d+.\d{3}', str(usdt_balance*0.05)).group(0)
order = kClient.create_market_order('BTC-USDT', kClient.SIDE_BUY, funds = funds)
```

Output:

> Current BTC Price: 22420.7
>
> USDT Balance: 48942.07811348 $
>
> BTC Balance: 0.00194057 (43.508937799 $)

Strategy

Building the strategy to make signals when a trade (buy or sell) should be executed, was a real challenge. There are hundreds of different strategies on different technical indicators, but now the buy or sell signal should be based on the sentiment of tweets on Twitter ([FR 30]).

At first, the idea was to calculate the average of sentiment for a particular time period and if this average reaches a certain threshold, a trade should be executed.

In the left table in Figure 14 you see the Timestamp, Sentiment Score and the corresponding meaning for *each tweet*.

Sentiment Score for Tweets in a Timeframe of 72 hours

	Sentiment Score	Sent is
2022-08-06T04:38:00	0.4019	Positive
2022-08-06T04:38:00	-0.3804	Negative
2022-08-06T04:38:00	0.4019	Positive
2022-08-06T04:38:00	-0.6361	Very Negative
2022-08-06T04:11:00	0.6605	Very Positive
2022-08-06T04:11:00	0.4404	Positive
2022-08-06T04:11:00	-0.8934	Very Negative
2022-08-06T04:11:00	0.8316	Very Positive
2022-08-06T04:11:00	0.6369	Very Positive
2022-08-06T04:11:00	0.1779	Positive

Average Sentiment and Tweet Count for 60 Min. Periods

	Avg	Sent is	Total	Signal
2022-08-06T05:00:00	0.2093	Positive	16	BUY
2022-08-06T04:00:00	0.2217	Positive	46	BUY
2022-08-06T03:00:00	0.2062	Positive	18	BUY
2022-08-06T02:00:00	-0.0078	Negative	2	SELL
2022-08-06T00:00:00	0.3530	Positive	12	BUY
2022-08-05T19:00:00	0.3433	Positive	39	BUY
2022-08-05T18:00:00	0.3263	Positive	13	BUY
2022-08-05T14:00:00	0.2736	Positive	42	BUY
2022-08-05T13:00:00	0.1280	Positive	1	BUY
2022-08-05T06:00:00	0.0888	Positive	39	BUY

Figure 14: Sentiment for a single tweet on the left, Average Sentiment and counted Tweets for a timeperiod of 1h

The right table shows the grouped and counted tweets for intervals of 60 min. Afterwards, the average sentiment is calculated and converted into a *Buy* or *Sell* Signal. It can be seen that there isn't a lot of tweets and some timestamps are missing. This was an error from SQL-Alchemy, which lead to the state of not

committing the current Database-Session and thus leading to missing timestamps. This is fixed in the final version, but was kept here to correspond to the below chart.

The sentiment score for single tweets are all very different, but the average will naturally move towards neutral sentiment (0.0). This shift would have been even more if all the neutral tweets had not been removed beforehand.

It would have been easy to just define the buy signal when the average sentiment is positive and a sell signal when the average sentiment is negative. But this would mean a lot of buy signals for the above example and since the database is not quite big enough to see if this average is only positive at the moment or if the calculation just tends to be a little above neutral. The bigger the time intervals, the more tweets and the bigger the shift towards neutral. This is why the threshold is set to 0.2.

> An average sentiment above 0.2 means *Buy*, below 0.2 means *Sell*.

Papertrading

The image below shows the sentiment and bitcoin price from August 5th till August 6th. If the sentiment is above 0.2 (positive or very positive) it is marked as a buy signal in the chart (green triangle). Since this looked promising it was implemented to real-time Papertrading in the Kucoin Sandbox ([FR 40]).

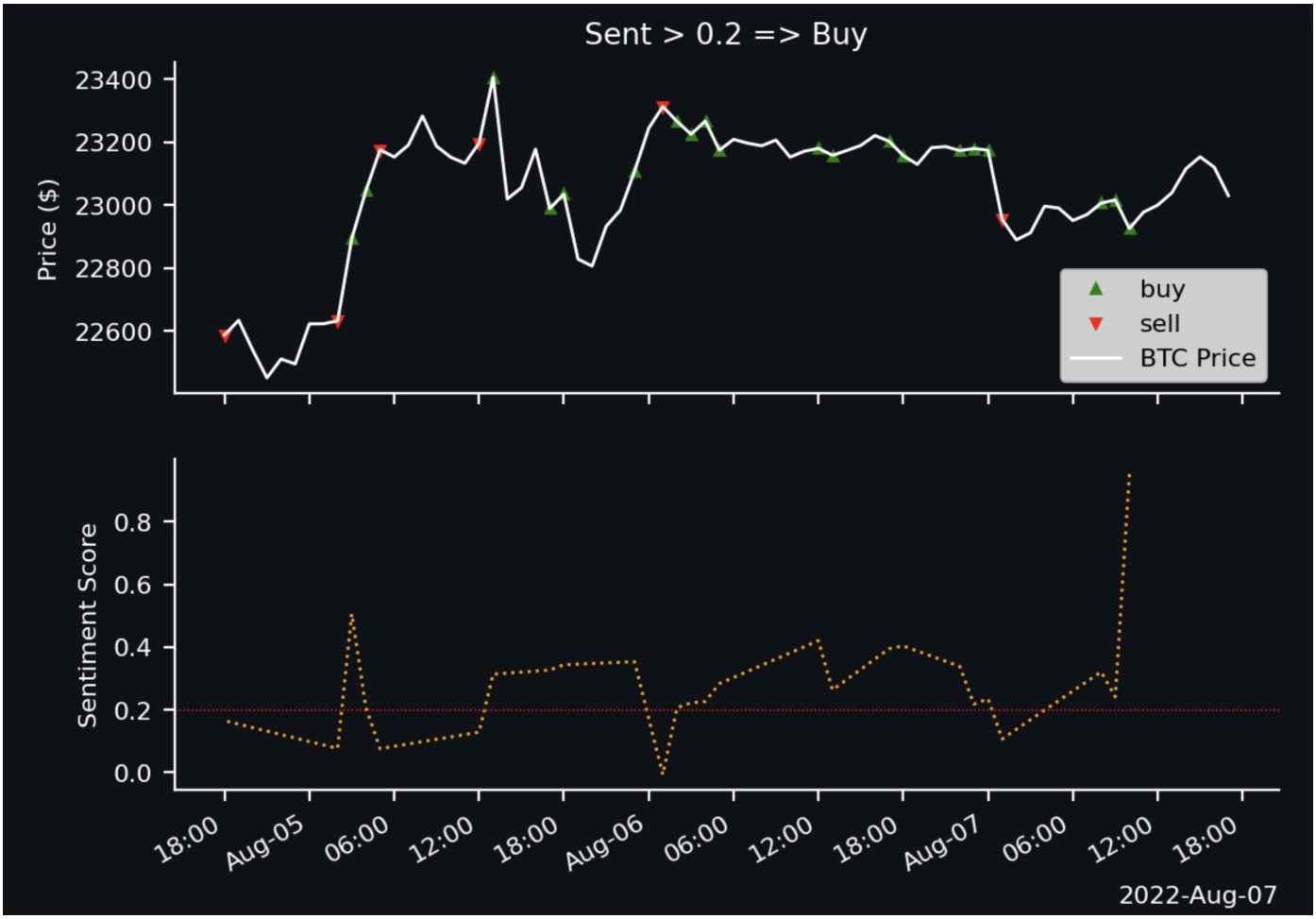


Figure 15: Charts for Bitcoin Price and Average Sentiment with corresponding signals

The trade-file starts with the essential function to collect all the single tweets from the database and stores them in a pandas DataFrame. This is needed to delete the duplicates with the following methods:

```
duplicates = list(
    df.index[
        df.duplicated(
```

```
            subset=["Tweet"], keep=False
        )
    ]
)

df.drop_duplicates(
    subset=["Tweet"], keep=False, inplace=True
    )
```

Afterwards, the neutral tweets are filtered and the rest are counted and the mean/avg is calculated for a resampled interval of 1h. This means all the timestamps from 1h (p.e. from 16:00 - 17:00) are grouped and a function is applied to all of them. This function could be to summarize, count or calculate the mean/avg.

```
df = df[df["Sentiment Score"] != 0.0]

count_tweets = df.resample(f"1H").count()

mean_df = df.resample(f"1H").mean().sort_index(ascending=False)
```

These two series get concatenated together to have one table:

	Avg	Sent is	Total	Signal
2022-08-06T05:00:00	0.2093	Positive	16	BUY
2022-08-06T04:00:00	0.2217	Positive	46	BUY
2022-08-06T03:00:00	0.2062	Positive	18	BUY

Figure 16: Last three time periods with average, total tweets and signal

And these last three timestamps are important for the trading. The first row is showing the actual timestamp. So at the time this picture was taken it was somewhere between 05:00 and 06:00. Since the tweets are collected live and every few seconds this row will be updated frequently and the avg and total tweets will change.

This is why the second row will be selected as the trading decision. Firstly, it will be checked if a trade was already made for this timestamp. Then, if the signal indicates buy, a market order will be created with 5% of the available USDT Balance (after a check, if there are any funds available).

A sell order will sell a quarter of the current bitcoin holdings.

At last, some metrics from the trade are uploaded into a separate table on Heroku for a better overview of the trades and later calculation of current PNL (Profit and Loss). An excerpt from this table is shown below in figure 17. It contains the sentiment average and the time period, as well as information about the trade itself. The last balances are actually the balances after the trade was made. So, it is possible, to look back at all the balances without gaps.

Last Trades for Kucoin Sub-Account

	↓ id	avgTime	avg	tradeAt	symbol	side	funds	fee	tradeId	usdt_balance	btc_balance
2	9	2022-08-12T17:00:00	0.2780	2022-08-12T18:17:30	BTC-USDT	buy	41	0.0407	62f67d1a36d21f00011d68e8	773.5778	0.0101
1	8	2022-08-12T16:00:00	0.2228	2022-08-12T17:05:59	BTC-USDT	buy	43	0.0429	62f66c570091a6000100d8c4	814.3352	0.0083
0	7	2022-08-12T15:00:00	0.2829	2022-08-12T16:28:39	BTC-USDT	buy	45	0.0451	62f6639736d21f00011d4ad1	857.2399	0.0064

Figure 17: Last Trades with Kucoin Sub-Account

Looking at the trades

Figure 18 shows the Heroku Logs. At first, the runner is started, that listens to the tweets and adds them to the database. When you see the line `Listening to tweets now...`, you know, everything is working properly.

```
2022-08-14T09:47:00.202248+00:00 heroku[worker.1]: Starting process with command `python3 sentiment/runner.py`
2022-08-14T09:47:00.913455+00:00 heroku[worker.1]: State changed from starting to up
2022-08-14T09:47:05.435285+00:00 app[worker.1]: Listening to tweets now...
2022-08-14T09:47:13.000000+00:00 app[api]: Build succeeded
2022-08-14T09:52:23.497676+00:00 app[api]: Scaled to clock@1:Free worker@1:Free by user y.heinze@chainsulting.de
2022-08-14T09:52:27.252173+00:00 app[api]: Scaled to clock@0:Free worker@1:Free by user y.heinze@chainsulting.de
2022-08-14T09:52:27.581412+00:00 heroku[clock.1]: State changed from starting to down
2022-08-14T09:52:35.527765+00:00 heroku[clock.1]: Starting process with command `python sentiment/clock.py`
2022-08-14T09:52:36.987869+00:00 heroku[clock.1]: Stopping all processes with SIGTERM
2022-08-14T09:52:37.152912+00:00 heroku[clock.1]: Process exited with status 143
2022-08-14T10:10:12.941988+00:00 app[api]: Starting process with command `python sentiment/trade.py` by user scheduler@addons.heroku.com
2022-08-14T10:10:23.899460+00:00 heroku[scheduler.3347]: Starting process with command `python sentiment/trade.py`
2022-08-14T10:10:24.714393+00:00 heroku[scheduler.3347]: State changed from starting to up
2022-08-14T10:10:28.311348+00:00 app[scheduler.3347]: Sent Avg: 0.2359751700680272
2022-08-14T10:10:28.311375+00:00 app[scheduler.3347]: USDT Balance: 512.99176508 $
2022-08-14T10:10:28.311375+00:00 app[scheduler.3347]: BTC Balance: 0.02153123
2022-08-14T10:10:27.801463+00:00 app[scheduler.3347]: Getting data from Today
2022-08-14T10:10:27.807566+00:00 app[scheduler.3347]: /app/.heroku/python/lib/python3.10/site-packages/pandas/io/sql.py:761: UserWarning: pandas only support
SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLAlchemy
2022-08-14T10:10:27.807568+00:00 app[scheduler.3347]:   warnings.warn(
2022-08-14T10:10:27.919404+00:00 app[scheduler.3347]: Deleted 249 duplicates from a total of 1299
2022-08-14T10:10:27.939517+00:00 app[scheduler.3347]: /app/.heroku/python/lib/python3.10/site-packages/pandas/io/sql.py:761: UserWarning: pandas only support
SQLAlchemy connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLAlchemy
2022-08-14T10:10:27.939520+00:00 app[scheduler.3347]:   warnings.warn(
2022-08-14T10:10:27.941908+00:00 app[scheduler.3347]: Check if a trade exists for 2022-08-14 10:00:00
2022-08-14T10:10:27.941911+00:00 app[scheduler.3347]: Last Trade at: 2022-08-13 11:00:00
2022-08-14T10:10:27.941929+00:00 app[scheduler.3347]: No Trade. Starting Trade for 2022-08-14 10:00:00
2022-08-14T10:10:28.797925+00:00 app[scheduler.3347]: BUY ORDER executed with 25.64959 at 2022-08-14 10:10:28.797858
2022-08-14T10:10:39.276876+00:00 app[scheduler.3347]: New Balances: 487.3165263 $ and 0.08245662 btc.
2022-08-14T10:10:39.278048+00:00 app[scheduler.3347]: Added Trade to Heroku DB
2022-08-14T10:10:39.668776+00:00 heroku[scheduler.3347]: Process exited with status 0
2022-08-14T10:10:39.878332+00:00 heroku[scheduler.3347]: State changed from up to complete
```

Figure 18: Logs from Heroku on scheduled trades

As explained in the Backend-Section, the scheduler is executing the trading script every hour.

The sentiment average is shown and the Balances before and after the trade. It also says if a trade already exists and then either starts a new trade or does nothing.

The Heroku logs were an essential part of the development-phase, since they allowed for a good way of debugging. Usually, no `print`-statements are left in the final code, when it's entering the production phase. However, Heroku seems to only print the `print`-statements rather than from the logging-library, which is otherwise used. Therefore, a few print statements are left in the final code.

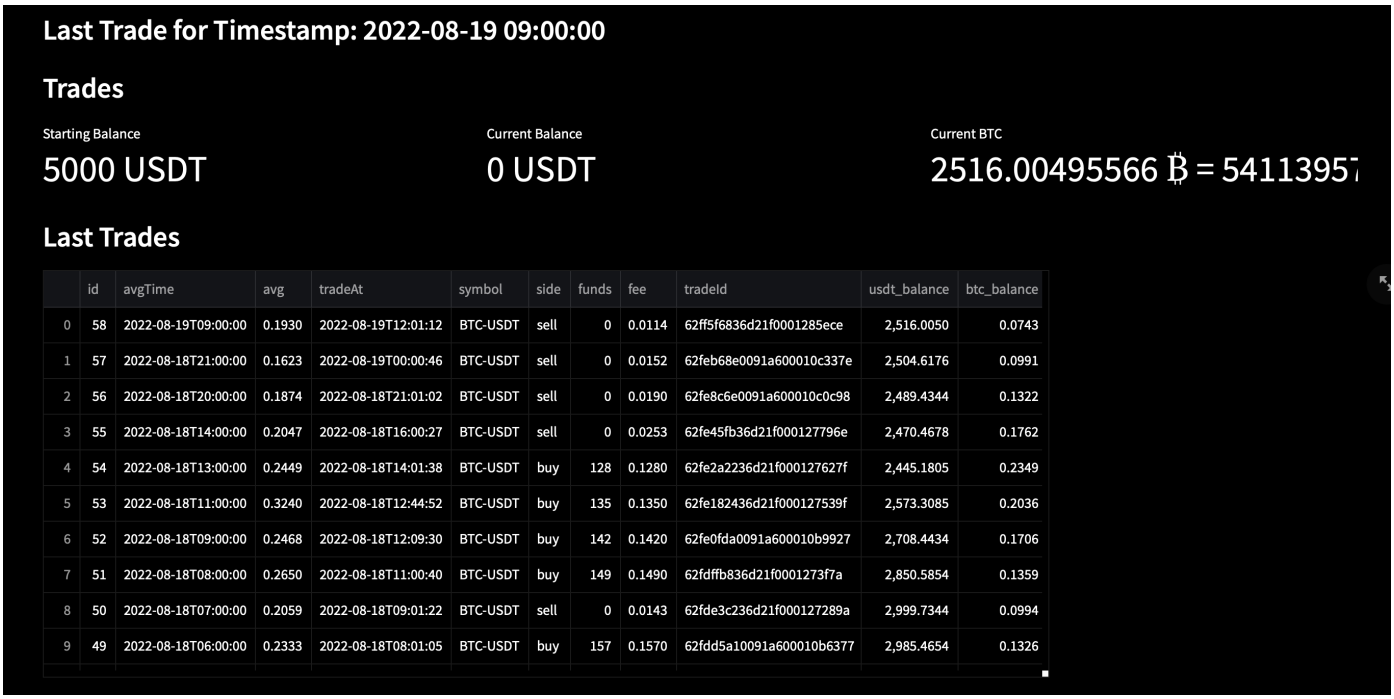
Challenges

Getting false Account Balance

The System was trading fine for a couple of days, but then, all of a sudden, it did not execute any trades for a couple timestamps. As can be seen in figure 19 below, the USDT and BTC Balance are switched, and the system tried to buy for an amount of *0* USDT, which did not work. Interestingly, in figure 20, you can see that this was not a consistent state since there are some trades at random timestamps. The problem lay with Kucoin. When acquiring the current account balances, USDT normally came before BTC. However, this order seems to be switched at random. This unforeseen coincidence needed to be checked by the system and be acted upon.

```
2022-08-19T09:01:26.243265+00:00 app[scheduler.9790]: Check if a trade exists for 2022-08-19 08:00:00
2022-08-19T09:01:26.243268+00:00 app[scheduler.9790]: Last Trade at: 2022-08-18 21:00:00
2022-08-19T09:01:26.243310+00:00 app[scheduler.9790]: No Trade. Starting Trade for 2022-08-19 08:00:00
2022-08-19T09:01:27.537029+00:00 app[scheduler.9790]: Starting trade process at 2022-08-19 09:01:27.536970
2022-08-19T09:01:27.537178+00:00 app[scheduler.9790]: Sent Avg: 0.2178597464342314
2022-08-19T09:01:27.537194+00:00 app[scheduler.9790]: USDT Balance: 0.09911778 $
2022-08-19T09:01:27.537208+00:00 app[scheduler.9790]: BTC Balance: 2504.61755446
2022-08-19T09:01:27.537221+00:00 app[scheduler.9790]: Buying for 0
2022-08-19T09:01:27.537247+00:00 app[scheduler.9790]: Exception: MarketOrderException: Need size or fund parameter
```

Figure 19: Logs from Heroku: No Trade was executed because the size or fund parameter was false



Last Trade for Timestamp: 2022-08-19 09:00:00

Trades

Starting Balance
5000 USDT

Current Balance
0 USDT

Current BTC
2516.00495566 ₿ = 5411395

Last Trades

	id	avgTime	avg	tradeAt	symbol	side	funds	fee	tradeId	usdt_balance	btc_balance
0	58	2022-08-19T09:00:00	0.1930	2022-08-19T12:01:12	BTC-USDT	sell	0	0.0114	62ff5f6836d21f0001285ece	2,516.0050	0.0743
1	57	2022-08-18T21:00:00	0.1623	2022-08-19T00:00:46	BTC-USDT	sell	0	0.0152	62feb68e0091a600010c337e	2,504.6176	0.0991
2	56	2022-08-18T20:00:00	0.1874	2022-08-18T21:01:02	BTC-USDT	sell	0	0.0190	62fe8c6e0091a600010c0c98	2,489.4344	0.1322
3	55	2022-08-18T14:00:00	0.2047	2022-08-18T16:00:27	BTC-USDT	sell	0	0.0253	62fe45fb36d21f000127796e	2,470.4678	0.1762
4	54	2022-08-18T13:00:00	0.2449	2022-08-18T14:01:38	BTC-USDT	buy	128	0.1280	62fe2a2236d21f000127627f	2,445.1805	0.2349
5	53	2022-08-18T11:00:00	0.3240	2022-08-18T12:44:52	BTC-USDT	buy	135	0.1350	62fe182436d21f000127539f	2,573.3085	0.2036
6	52	2022-08-18T09:00:00	0.2468	2022-08-18T12:09:30	BTC-USDT	buy	142	0.1420	62fe0fda0091a600010b9927	2,708.4434	0.1706
7	51	2022-08-18T08:00:00	0.2650	2022-08-18T11:00:40	BTC-USDT	buy	149	0.1490	62fdffb836d21f0001273f7a	2,850.5854	0.1359
8	50	2022-08-18T07:00:00	0.2059	2022-08-18T09:01:22	BTC-USDT	sell	0	0.0143	62fde3c236d21f000127289a	2,999.7344	0.0994
9	49	2022-08-18T06:00:00	0.2333	2022-08-18T08:01:05	BTC-USDT	buy	157	0.1570	62fdd5a10091a600010b6377	2,985.4654	0.1326

Figure 20: Wrong Account Balance on Kucoin

Different Sandbox Prices

The biggest challenge came with the sandbox. All the trading worked perfectly, but it was weird to see that the first trade of 250 $ equalled a BTC amount of 0.06, which would be more than 1300 $. This didn't make any sense.

Since these different prices were not prone to bitcoin alone, but all coins had different prices in the sandbox, the Kucoin Support was contacted.

The answer (figure 21), confirmed that the prices in the sandbox are just different, and another solution was needed.

Dear Valued KuCoin User,

Thank you for reaching out to KuCoin customer support.

Regarding your issue, we've confirmed with responsible team, and it turns out the market in sandbox and actual is totally different, in this way, it's normal to have the exchange rate and price. For example, the price of BTC in sandbox is 431USD, while the actual price is 2111USD.

Thank you in advance for your understanding. If you have any other questions or concerns, please feel free to let us know!

Kind regards,

KuCoin Customer Care and Support Team

Figure 21: Email from Kucoin regarding the different Bictoin Prices

The solution was to get the correct price from somewhere else and in this case: Binance.

To convert the current BTC Holdings a real-time price was needed.

This is done with a few lines of code and a typical HTML-Request:

```
url = "https://api.binance.com/api/v3/ticker/price?symbol=BTCUSDT"
data = requests.get(url)
data = data.json()
```

To get all the historical data from Binance, the python-binance wrapper was used:

```
frame = pd.DataFrame(binance_client.get_historical_klines(symbol,1,lookback))
frame = frame.iloc[:,:6]
frame.columns= ["Time","Open","High","Low","Close","Volume"]
frame = frame.set_index("Time")
frame.index = pd.to_datetime(frame.index,unit="ms")
frame.index = frame.index + timedelta(hours=2) #utc to local
frame
```

The result looked like this:

Time	Open	High	Low	Close	Volume
2022-08-17 14:00:00	23746.65	23747.91	23728.86	23743.53	110.15727
2022-08-17 14:01:00	23743.53	23760.09	23735.18	23759.22	84.61940
2022-08-17 14:02:00	23760.09	23760.38	23743.77	23747.17	68.60458
2022-08-17 14:03:00	23747.92	23753.01	23723.01	23724.97	106.61761
2022-08-17 14:04:00	23726.01	23739.66	23720.03	23729.27	117.75679
...	...	...	...	...	...
2022-08-17 21:56:00	23274.80	23280.69	23261.72	23265.75	127.68311
2022-08-17 21:57:00	23264.37	23279.68	23262.70	23270.71	96.70607
2022-08-17 21:58:00	23272.11	23292.11	23270.70	23291.21	95.62044
2022-08-17 21:59:00	23291.04	23294.03	23271.18	23275.04	115.32825
2022-08-17 22:00:00	23276.89	23288.00	23263.20	23265.36	98.09844

[481 rows x 5 columns]

Figure 22: Prices (ohlc) and Volume for Bitcoin from Binance

The method `get_historical_klines()` accepts arguments to look for intervals in a past time period. For example, in figure 22 we are looking for the past day ("1d") with intervals of one minute ("1m").

The timestamp and the closing price were then used for further calculations and visualisations.

Visualisation with Streamlit

When scripts are running in the background and acting on their own, a lot of work will never be seen. Since all the tweet and trade data are stored in a database and the sentiment analysis and price charts can be visualised in plots and diagrams, it was specified in the requirements ([FR 60]).

Streamlit is an open-source Framework for building simple and interactive Machine-Learning and Data Science Web-Apps. It was perfect for this project, because it contained all the features for charts and dataframes and was easy to implement and therefore allowed rapid prototyping.

Actually, Streamlit and Jupyter Notebooks were used to test the code before it entered Heroku, because each execution on Heroku consumed dynos and obviously increased the row count in the database. The jupyter notebooks can be found in this folder, but are not used in this documentation and therefore a bit messy.

In this project, Pandas was used for data analysis and Matplotlib to visualise the data. They both work seamlessly with streamlit.

In the following section are explanations for the single parts that have been visualised on a Website as stated in [FR 60].

This site is hosted at streamlit and if it is working correctly, the real-time current sentiment and trading data should be seen [there](#TODO:streamlit link). However, since they won't host it forever and will shut it down eventually, a screenshot of the site was taken at 24th August 2022 and is used to explain everything below.

A video was made to explain the whole visualisation in total. (The video is also in the visualisation folder)

Figure 23: Visualisation of tweet, sentiment and trading metrics

Final Visualisation

The site starts by explaining what a sentiment is and shows the `last collected Sentiment` and the `corresponding Signal`. The site is structured in a way the whole project was planned. First came the collection of tweets, followed by the sentiment analysis and lastly the trading part.

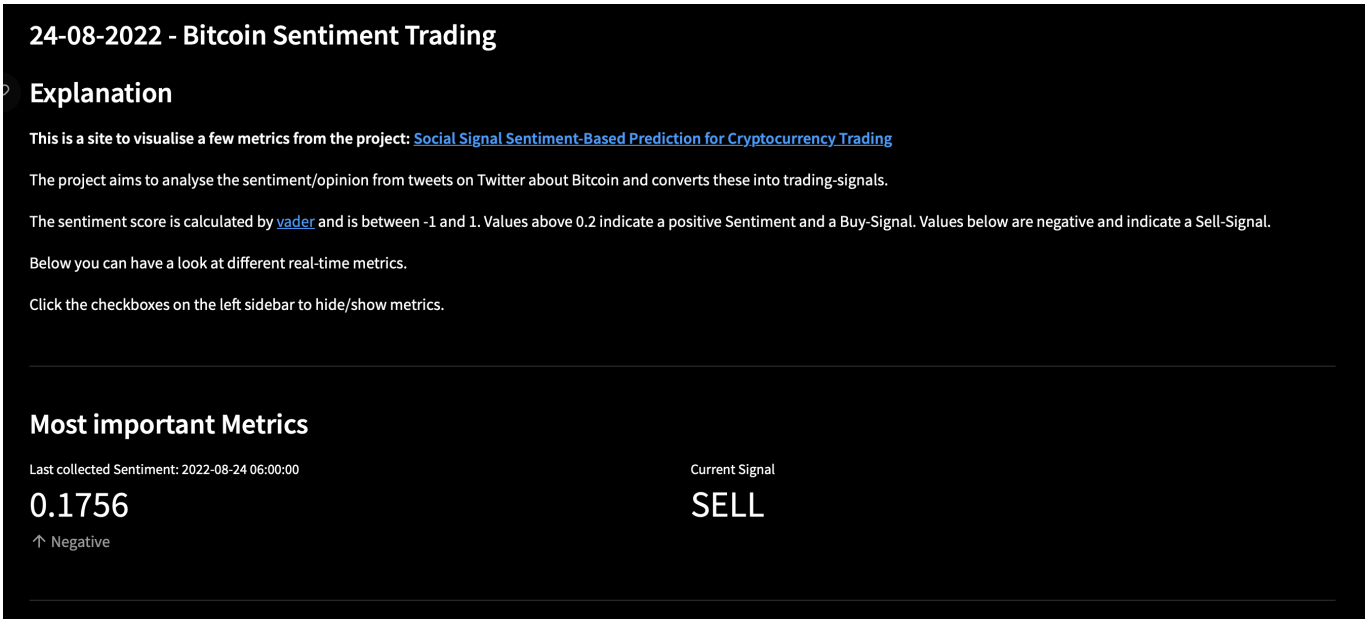
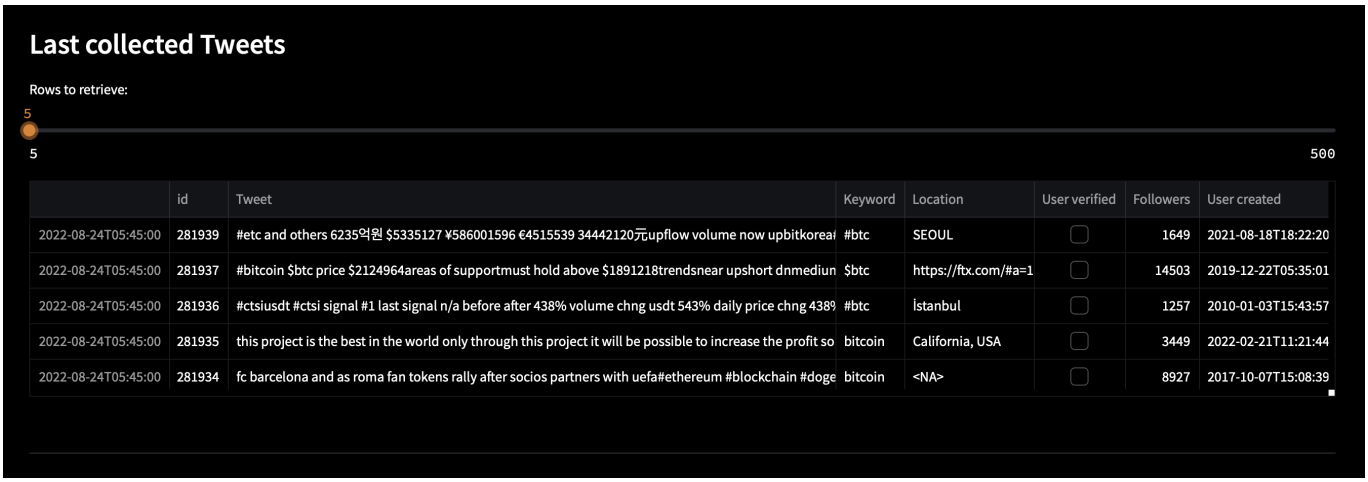


Figure 24: Explanation and most important Metrics

Last collected Tweets

With a slider, the rows of single tweets to retrieve from the database can be adapted (Figure 25). It is basically the same table as in the Data-Acquisition-Section, but now in real-time. It was very good to work with streamlit during the whole development phase to have a look at the current status of the heroku database without needing to go through the whole heroku dashboard (which takes quite a few steps to look at the dataclip of the database).



	id	Tweet	Keyword	Location	User verified	Followers	User created
2022-08-24T05:45:00	281939	#etc and others 6235억원 $5335127 ¥586001596 €4515539 34442120元upflow volume now upbitkorea	#btc	SEOUL	☐	1649	2021-08-18T18:22:20
2022-08-24T05:45:00	281937	#bitcoin $btc price $2124964areas of supportmust hold above $1891218trendsnear upshort dnmediun	$btc	https://ftx.com/#a=1	☐	14503	2019-12-22T05:35:01
2022-08-24T05:45:00	281936	#ctsiusdt #ctsi signal #1 last signal n/a before after 438% volume chng usdt 543% daily price chng 438	#btc	İstanbul	☐	1257	2010-01-03T15:43:57
2022-08-24T05:45:00	281935	this project is the best in the world only through this project it will be possible to increase the profit so	bitcoin	California, USA	☐	3449	2022-02-21T11:21:44
2022-08-24T05:45:00	281934	fc barcelona and as roma fan tokens rally after socios partners with uefa#ethereum #blockchain #doge	bitcoin	<NA>	☐	8927	2017-10-07T15:08:39

Figure 25: Last collected tweets with some metrics

Sentiment Metrics

Figure 26 shows some metrics about sentiment. At the date of collection (24-08-2022), a total of 40k Tweets were collected in the past four days and 8273 in the past 24h. The average follower count was 13.314, which is okay, considered only accounts under 500 followers are filtered. It means there are quite a lot of accounts with a high follower account, speaking for a high quality of tweets.

Over a few days, the average of deleted duplicates has always been around 45%.

> Nearly half of all tweets are duplicates! Presumably by bots.

This is a very high number and show how many bots are tweeting and trying to influence the opinion about Bitcoin. It has been the most eye-opening fact from this project. The cake diagram on the left shows the sentiment for the last timeframe. It corresponds to the first line from the table on the right. Neutral tweets have been removed as being explained in Sentiment. The table shows the last five time periods with the calculated sentiment average and total tweets. This was the base for the trading-part.

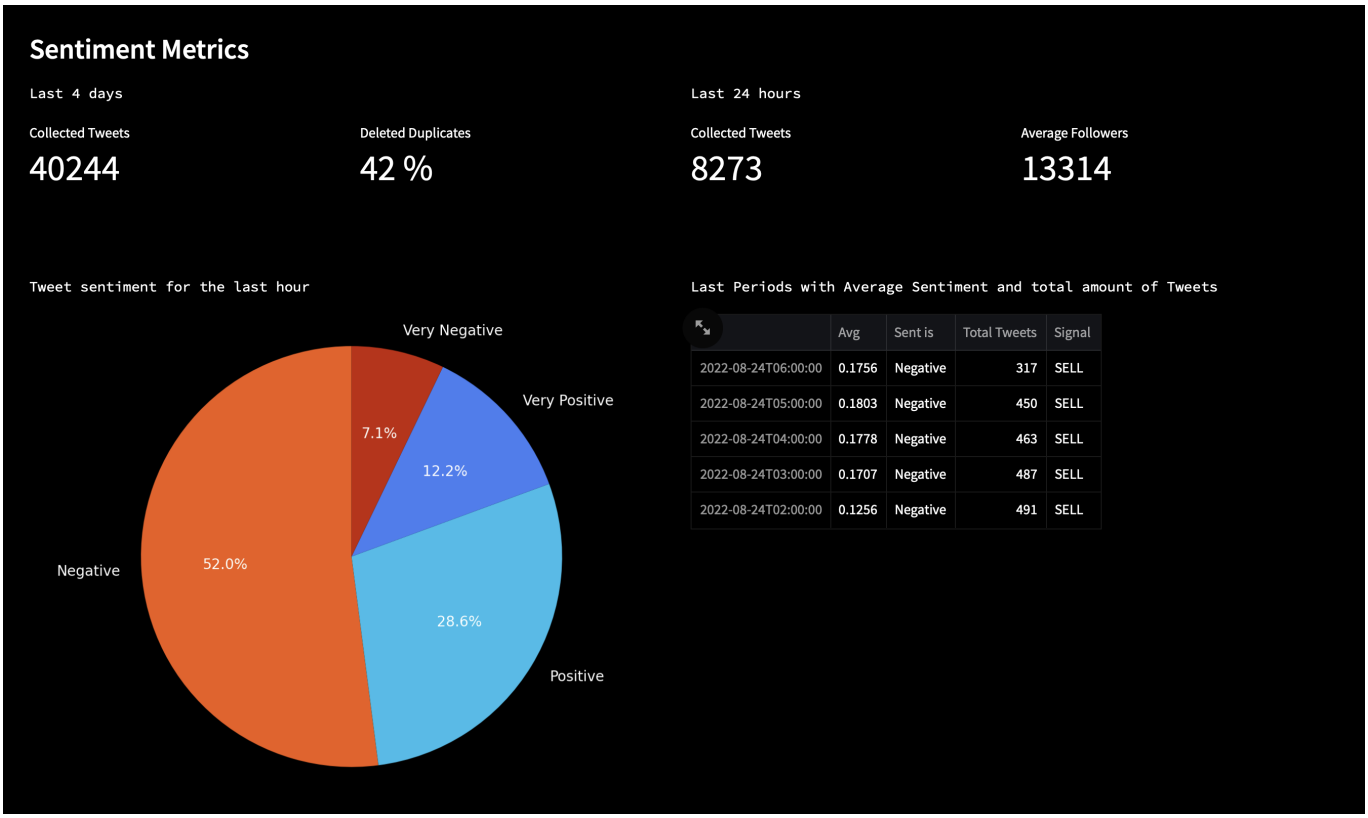


Figure 26: Sentiment Metrics

Word Analysis

The wordcloud on the left visualises the usage and frequency of words in all the tweets from the last time period.

In the middle, the usage of words which signal a buy- or sell-signal is shown. It stems from the following Subdivision of Words:

BUY-Signal	SELL-Signal
- buy & bought	- sell
- bullish & bull	- down
- high	- bearish
- pump	- sold
- growth	- never
- up & uptrend	- bad
- revolution	- low
- hold	- dump
- love	- decline

BUY-Signal	SELL-Signal
- trust	- downfall & downtrend
- success	- decay
- hodl	- recession
- like	- short
- profit	- hate
- gain	- #short
- #pump	- #dump
- long	- loss & lost & lose

It is very interesting to see that most of the words at this time were positive and signaling a buy trade in comparison to the overall negative sentiment of all tweets.

It would be a good idea to build a strategy based on the usage of these words and compare it to the overall sentiment-strategy. Out of time shortage, this couldn't be implemented during this project.

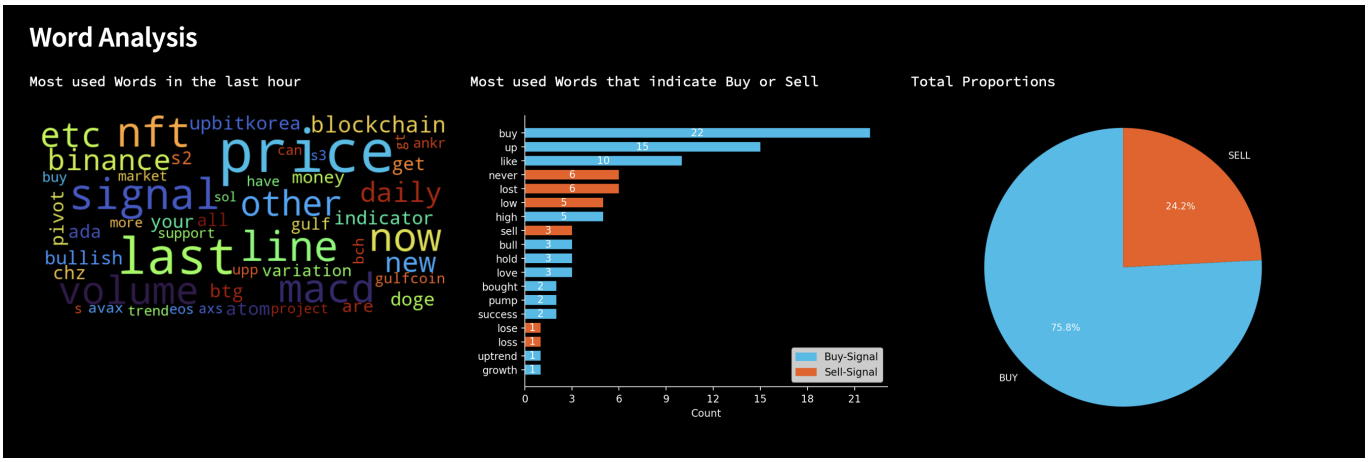


Figure 27: Word Metrics

Trading

For the trading part the first thing to have a look at are the most important metrics. The trading was started on the 16th August 2022 and after 8 days, a total of 90 trades were executed. The current holdings stood at 1785 $ USDT and the Bitcoin Balance was 0.0117 ₿ Bitcoin.

As being said in Challenges in the Trading-Section, the sandbox price is different from the real price of bitcoin. It was attempted to find a factor, but the sandbox price has unusual and incomprehensible price movements. This meant, the results could not be evaluated that easily. At the time of writing (24th August, 9:46) the holdings in bitcoin equalled 250$, which would lead to a total loss of 2965 $.

Figure 28: Kucoin Account Metrics

Figure 28 shows the last trades, visualised with the corresponding real-time bitcoin price. Working with Matplotlib to visualise the data has been an essential part for understanding the collected data and make predictions.

A valuable statement can be made with this chart:

> The sentiment about bitcoin on twitter is as volatile as the bitcoin price.

Buying at the lowest price and buying at the highest is the ultimate goal of trading. Since this is achieved very rarely and near impossible to persevere, fast trading looks for indicators to buy or sell quickly. Long-Term Investors on the other hand normally ride out short-time losses.

The strategy here was very simple and did not really take into account if a position of BTC is already held or if a sell-trade is being executed at a very bad price. The last part of the chart is a great example for this. The Buy-Trade happened on the 23th at 20:00 at the lowest price and then went only up afterwards, but the sentiment was always negative, so every hour a portion was sold. The system could wait for the first price drop after an uptrend and then sell. Same for the other way around. This would hold positions longer and lead to a better cost-average-effect.

This could be one way to improve the strategy. As can be seen below, a lot of trades are made during the day, because the system is scheduled to act every hour. If there wasn't that much price movement in Bitcoin, the trading times would be spread wider apart. For example, every 6 or even 12 hours. But this would mean that the calculated average moves more to *neutral*, which would decrease the significance of the score.

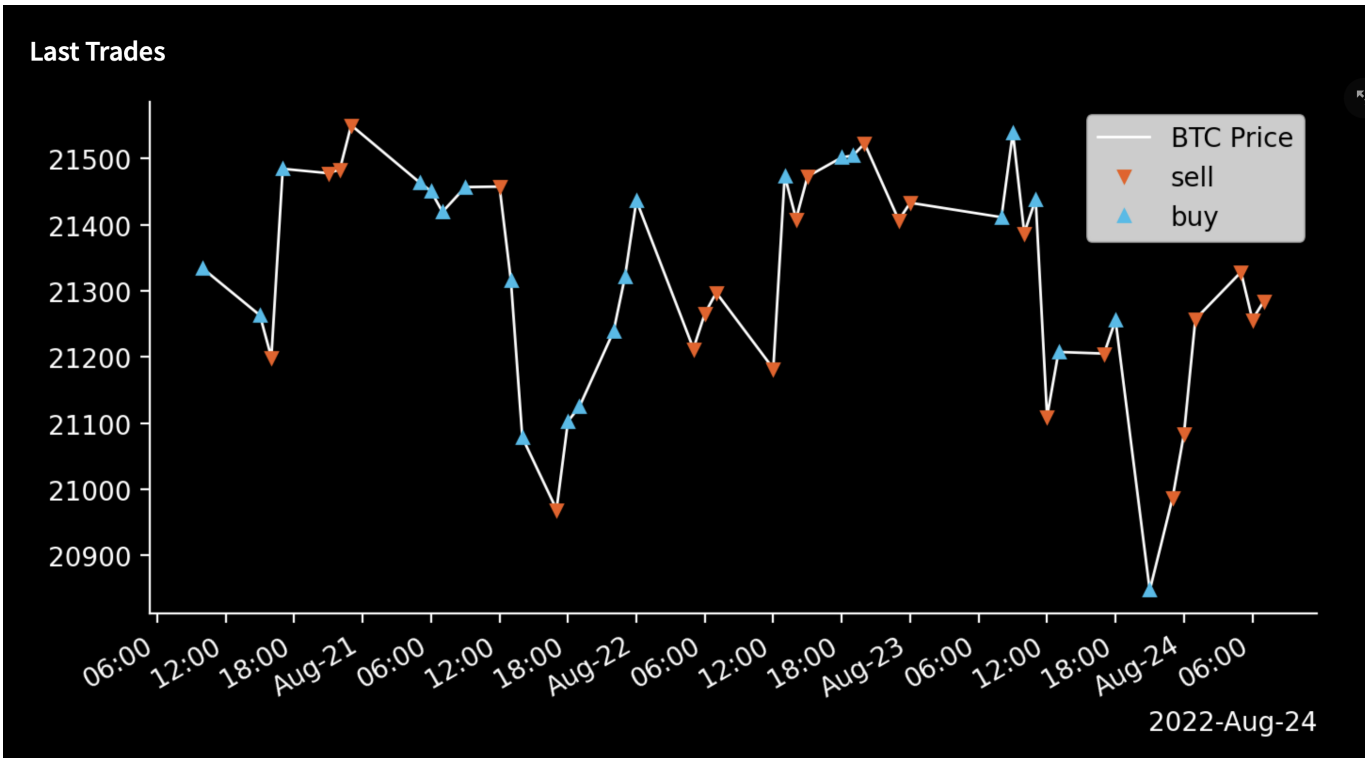
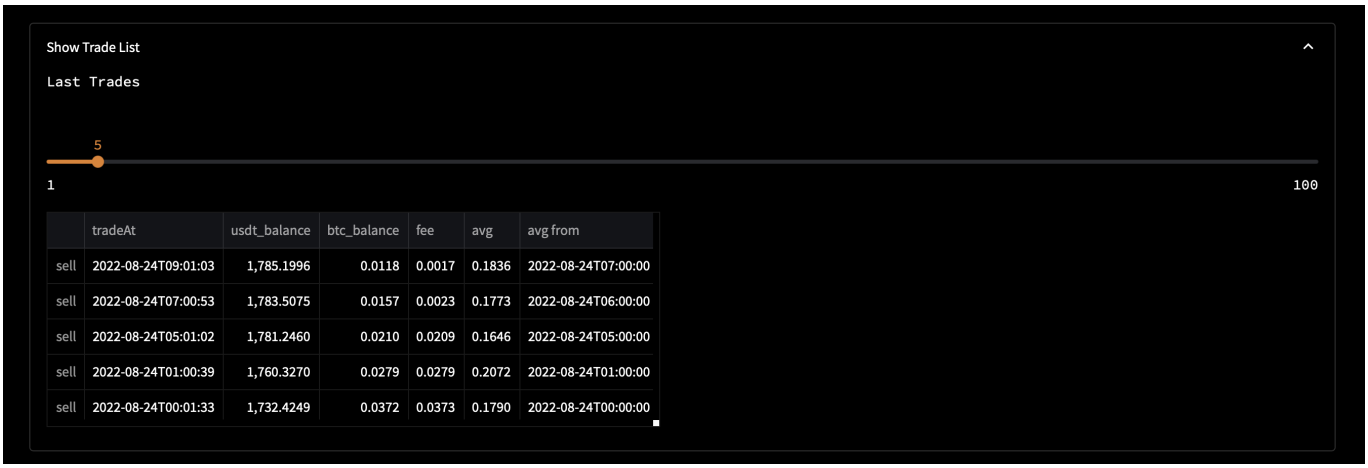


Figure 29: Chart of last trades

Figure 30 shows the list of all trades with some more metrics. The first column shows if the trade was a buy or a sell. "TradeAt", "Avg" and "AvgFrom" are self-explanatory. An important note for the Balances: They are taken after the trade. The fee is very important if it comes to a more detailed calculation of Profit-And-Loss, as being said by Garcia and Schweitzer:

> "Trading costs can potentially erode the profitability of trading strategies, especially if they require many movements." [3]



	tradeAt	usdt_balance	btc_balance	fee	avg	avg from
sell	2022-08-24T09:01:03	1,785.1996	0.0118	0.0017	0.1836	2022-08-24T07:00:00
sell	2022-08-24T07:00:53	1,783.5075	0.0157	0.0023	0.1773	2022-08-24T06:00:00
sell	2022-08-24T05:01:02	1,781.2460	0.0210	0.0209	0.1646	2022-08-24T05:00:00
sell	2022-08-24T01:00:39	1,760.3270	0.0279	0.0279	0.2072	2022-08-24T01:00:00
sell	2022-08-24T00:01:33	1,732.4249	0.0372	0.0373	0.1790	2022-08-24T00:00:00

Figure 30: Table of last trades

Summing up, there is a lot of improvement for different strategies to trade cryptocurrency based on the sentiment, but the foundation of all the technological necessities has been set.

Conclusion

Requirements

The requirements from the concept-phase are needed to be checked for fulfillment. It can be said, that all the *Must-Have* and *Should-Have*-Requirements have been achieved.

The system listens and filters real-time tweets, builds signals from a calculated sentiment and trades based on these signals. It can run in the background and stores all important data inside a database and lastly visualises all insightful data on a website.

The use of different strategies and Machine-Learning to improve the strategy couldn't be achieved. Since there were only *Optional*-Requirements, this was accepted.

With research on the used cryptocurrencies and social-media platform and the publication on GitHub, the non-functional requirements have been successfully achieved.

Results

The system started the automated trading based on sentiment at the 16th of August 2022 with a balance of 5000$ USDT. As can be seen in figure 31 below, the account balances didn't really work as a great overview. The Account Balance with the real Bitcoin Price from Binance lead to a way to high balance at first and Kucoin couldn't get any data for the trading times. Even after contacting the Kucoin Support, it is not clear why the Bitcoin Price in the Sandbox has such unusual behaviour. The figure 32 shows this in the column *"kucoin btc price"*. Within a day, the price moved from `431$` to `75.825$`. This doesn't make any sense. This has been the limiting factor when interpreting the results. However, after a small time of profits at the 20th August, the chart shows a clear downward trend for both lines. As being said before, a rather simple strategy was chosen to primarily test if everything works on the technical side and not to make big gains. Although, seeing that this type of trading is easily implemented and can run in the background, moving this system a level up by improving the filters and implementing different strategies, it would lead to better results.

There a many more strategies that could have been tested, and a few are described below in the Outlook-Section.

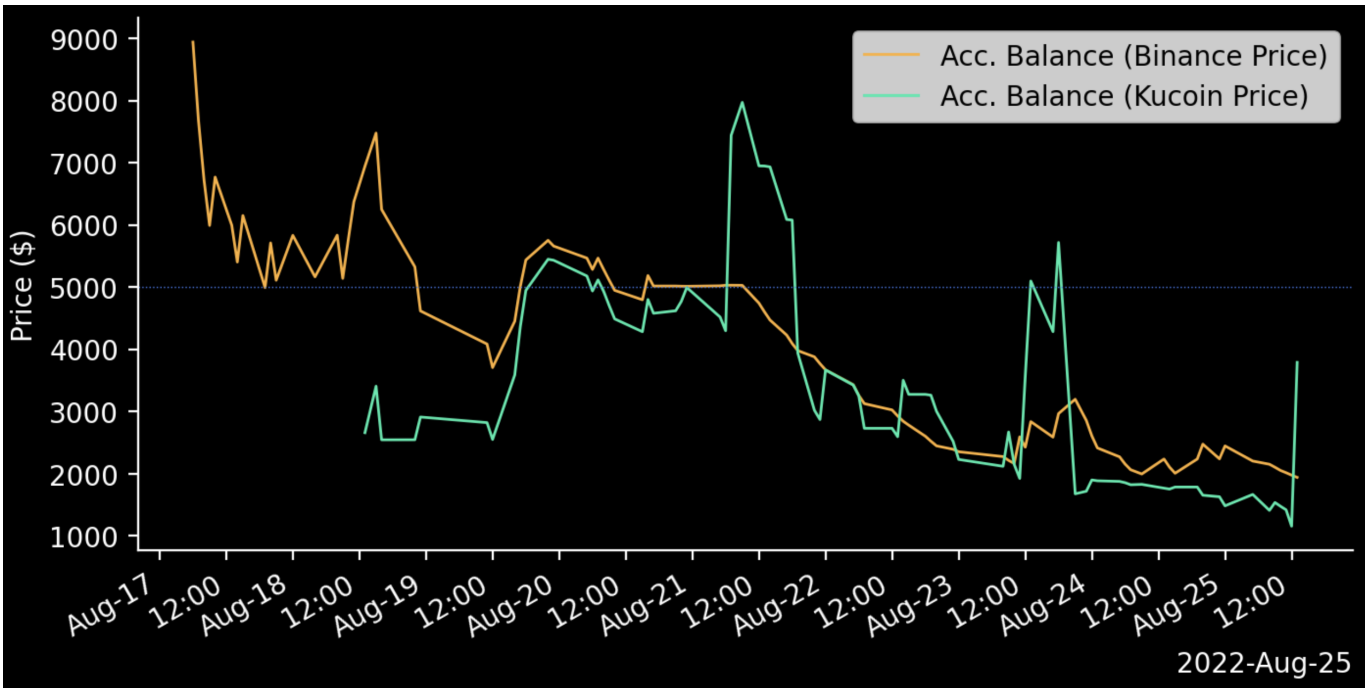


Figure 31: Chart of Account Balance with Binance and Kucoin Prices for Bitcoin

↓	side	binance btc price	kucoin btc price	binance_total	kucoin_total
2022-08-24T14:00:00	sell	24,407.6900	432.0000	2,114.9043	1,707.2912
2022-08-24T13:00:00	buy	23,774.3900	75,825.7500	2,236.4104	3,416.4720
2022-08-24T09:00:00	sell	23,808.7200	21,008.4000	1,996.9226	1,972.1844
2022-08-24T07:00:00	sell	23,743.5300	431.0000	2,064.9926	1,790.2768
2022-08-24T06:00:00	sell	23,670.2400	4,088.0000	2,155.4603	1,847.7442
2022-08-24T05:00:00	sell	23,432.1700	430.0000	2,272.2237	1,790.2353
2022-08-24T01:00:00	sell	23,381.7400	414.0000	2,413.4455	1,771.8637
2022-08-24T00:00:00	sell	23,449.5800	<NA>	2,605.7447	<NA>
2022-08-23T23:00:00	sell	23,306.6900	<NA>	2,852.6569	<NA>
2022-08-23T21:00:00	buy	23,452.9100	<NA>	3,198.4277	<NA>

Figure 32: Binance and Kucoin Prices for Bitcoin (in $) at Timestamps of Trades

Outlook

Better Strategies

Because of the short time of this project, further strategies or improvements of the sentiment strategy were not implemented. A very simple improvement would be a check, if a position of BTC is already held or if a sell-trade is being executed at a very bad price.

A few other strategies were:

- Looking for single words for "bullish" or "bearish" sentiment (like the word analysis)
- Using technical indicators like the 20-SMA (20 day single moving average) in combination with sentiment analysis
- Calculating the weighted average instead of a normal average
- Ratio of positive / negative sentiment crossing a particular threshold

Since there are way too many Cryptocurrencies and no way to have an overview over all of them, a good approach for trading with sentiment could be to listen to plenty of coins at the same time and then trade the coin that gets the most hype and then sell everything when the hype stops.

Building a modular system was actually the first approach and the infrastructure to listen to multiple tweets is basically set. The factor that prevented the final implementation of this modular system was the row limit of the database. The Twitter API also has a limit for keywords, but this wasn't reached.

Since the first approach to clean old entries in the database were a failure, the system was simplified to only listen to Tweets about Bitcoin.

The insights about Social Media have been very profound. Having nearly half of all the tweets being duplicates was a big surprise, as well as the fact, that many tweets are just nonsense (and I cannot see a reason for this).

Learnings

It has been an awesome and interesting project from the start and I have learned a lot of things:

- Building a whole python project from start to finish by myself
- Running scripts in the background on cloud provider Heroku
- Building and adding to a real-time database with Postgres
- Building a strategy for trading and testing it in real-time inside a sandbox from a crypto brokerage
- Visualisation with Matplotlib

There are a few things, I would do different next time:

I shouldn't have spent that much time figuring out a beautiful documentation platform (mkdocs and gitbook), that wasn't used in the end, but rather look for a simpler solution, that is easily implemented.

Some improvements in time-management are also needed. I would make a better plan with SMART goals that are not too hard to reach, so you can see a lot of progress very fast.

I shouldn't have spent that much time on visualisation when most functionality wasn't done yet.

However, this project has sparked my interest in data analysis and visualisation in such a fun way, that I can picture myself in this career path.

For this, I really want to thank Yannik Heinze and Robert Manzke for giving me the opportunity to work on this project and hopefully be of use for further projects.

Appendices

[1] Schoen, H., Gayo-Avello, D., Takis Metaxas, P., Mustafaraj, E., Strohmaier, M., & Gloor, P. (2013). The power of prediction with social media. Internet Research, 23(5), p. 528–543. https://doi.org/10.1108/intr-06-2013-0115

[2] Kristoufek, L. (2013). BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. Scientific Reports, 3(1). https://doi.org/10.1038/srep03415

[3] Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of Bitcoin. Royal Society Open Science, 2(9), p.8. https://doi.org/10.1098/rsos.150288

[4] Chainsulting Website