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:

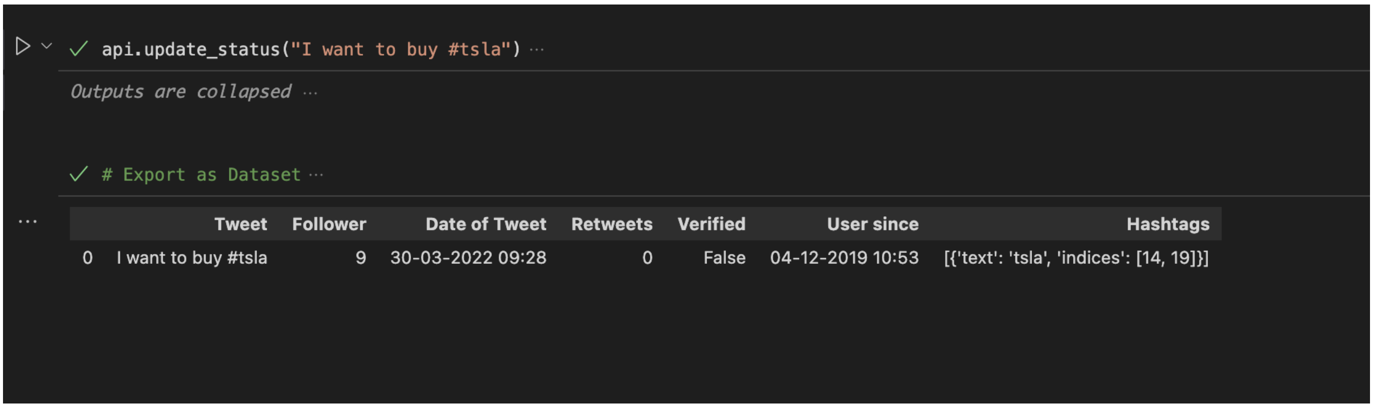
```
import tweepy
auth = tweepy.OAuth1UserHandler(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True)
posts_for_elon_musk = api.user_timeline(screen_name="elonmusk",
count=5,tweet_mode = "extended")
print(posts_for_elon_musk)
```

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:

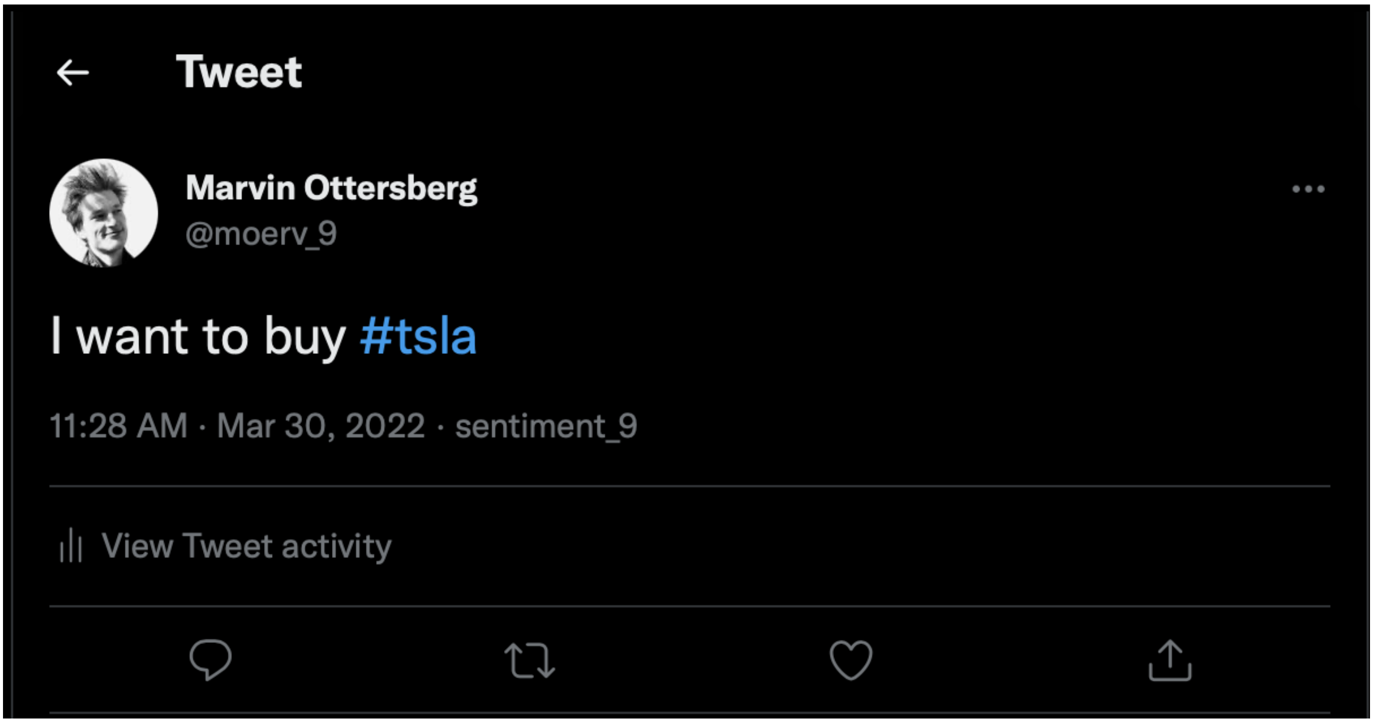


```
api.update_status("I want to buy #tsla") ···
Outputs are collapsed ···

# Export as Dataset ···
```

	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



Marvin Ottersberg
@moerv_9

I want to buy #tsla

11:28 AM · Mar 30, 2022 · sentiment_9

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:

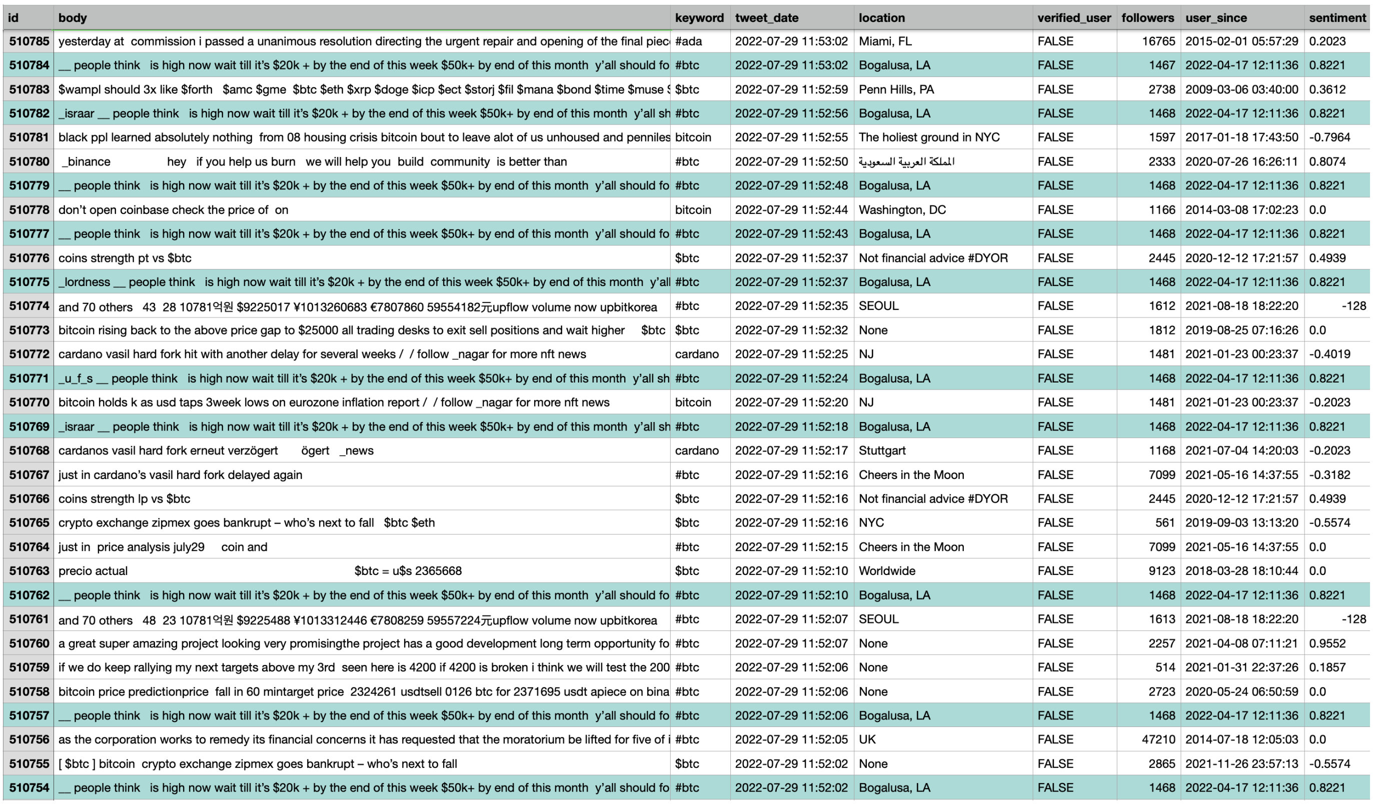
> "_boii &amp people think is high now wait till it’s $20k + by the end of this week $50k+ by end of 734982 this month y’all should follow he's a super underrated !bitcoiner i’ve been :following her tweets and tips seriously i’ve been doing really great! #btc #ada"

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 penniles	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.