IE School of Human Sciences & Technology

Master's in Big Data & Business Analytics

Course Name:

MODERN DATA ARCHITECTURE FOR BIG DATA I

Supervised by:

Prof. JORGE CENTENO FERNANDEZ

Submitted by:

MANUEL MARINA HERRERA

TABLE OF CONTENTS

Problem Description

Since March 2020 the world's population is experiencing a worldwide pandemic caused by the coronavirus. It is a highly infectious virus called SARS-CoV-2 which can be spread from an infected person's mouth or nose in small liquid particles by coughing, sneezing or breathing. This pandemic has substantially affected the global population with almost 780'000 known Covid-19 deaths only in the United States and through economic measurements to stop the spread of the virus. According to researchers, besides wearing a mask and conducting other hygienic measures such as disinfecting hands, the Covid-19 vaccination is a key factor to prevent the virus from further mutating and hence, to end the pandemic. About 56.7 percent of the worldwide population has received a vaccination, mainly in first world countries, with a continuing rising trend. However, there is a growing number of people, especially in Europe, dismissing the threat of the virus, categorising it as a media hype, driving "Anti-Vax" myths in social media. Moreover, they are raising their voices against "coercive government policies" such as vaccination campaigns and further lockdowns. There is a clear trend of people polarising in either denying covid or embracing covid measures to end the pandemic. Since the last two presidential elections in the United States, it is known that social media can have a major influence on people's opinion. The goal of our analysis is to explore what role can Twitter play in the formation of people's opinion about Covid-19 vaccination.

Data Sources

Sources of data

The source of data is in this case, a social media platform. It is an external and streaming data source located far from our storage layer (third party data source) as the data comes from Twitter servers.

Original data formats

We used the Search API to get data in JSON format (semi-structured textual data) about some specific hashtags.

Data Ingestion

Ingestion of data for each source

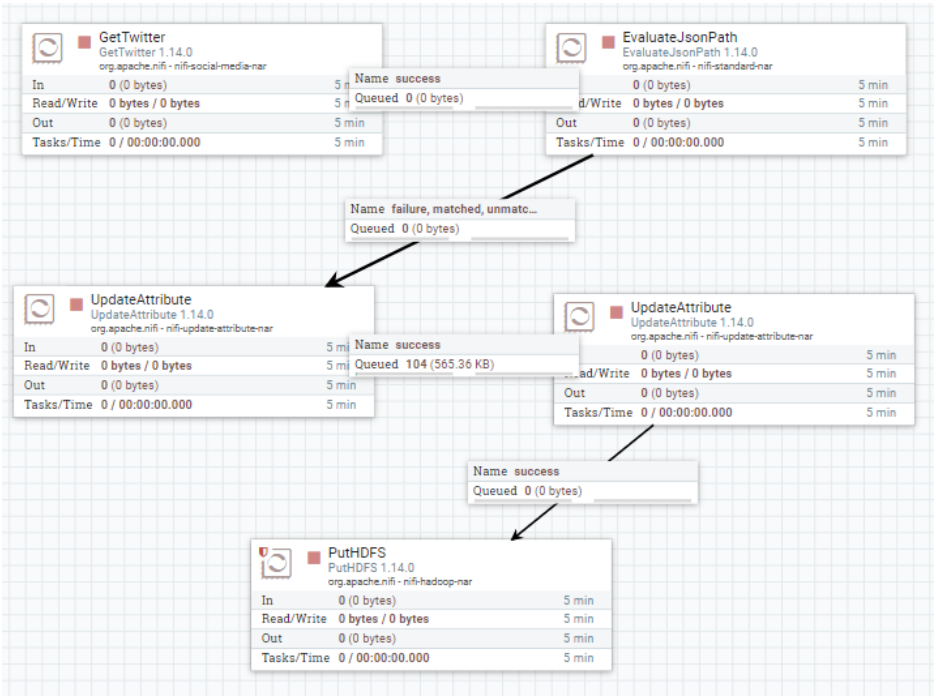
The data ingestion is made creating a data flow that is updated every 5 minutes, using Nifi as software. For this process it is required to get a Twitter developer account and to obtain API key value, API key Secret Value, Token Value and Token Secret Value. We selected five languages (en, es, fr, de, zh), and the following hashtags:

- #CancelCovid
- #HealingStartsHere
- #CovidVaccine
- #ImVaccinated
- #IGotVaccinated
- #VaccinesSaveLives
- #VaccinesWork

- #TheTruthAboutCovid
- #Vaccination
- #NoVaccination
- #zusammengegencorona
- #IYomevacuno
- #Vacuna
- #Vacunate

With the different processors we are able to ingest tweets, and to create a new attribute (created_at:toDate attribute) to order in a better and clearer way, all the tweets in a structured file system.

NiFi data pipelines diagram



We ingest tweets using the GetTwitter processor with the languages and hashtags explained before. In addition, we create the attribute “created_at:toDate attribute” in two steps: first we add the new property “created_at” in the evaluateJsonPath processor, and later we update it with the present date using the second UpdateAttribute. Once this is done, we create new properties using a new UpdateAttribute processor: day, month, year and “filename”.json (in our case the filename is created joining (year&month&day). With these new properties we can create the directory structure in our Data Lake with the PutHDFS processor.

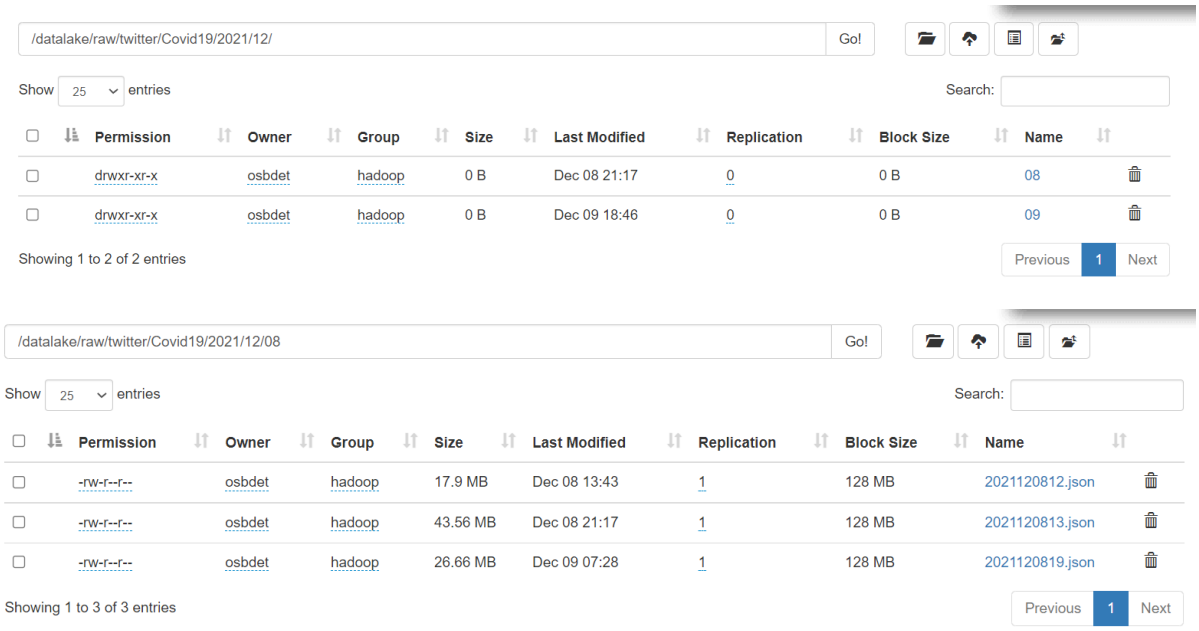
Data Storage

Storage of data

For storage we use Hadoop HDFS (an open source batch storage) creating big files to store all the tweets in real time in our own Data Lake using a distributed file system structure. As the data are not being transformed in the ingestion step, it is necessary to create a raw directory and some specific folders to differentiate our topic: Twitter and Covid19. The last architecture step is to use the properties created in the ingestion part (year,month,day) to create a cascade structure with them after our topic. The final folder structure would be as follow:

/datalake/raw/twitter/Covid19/${year}/${month}/${day}/"filename".json

As we ingest data in different days we have different folders in day-level and also as we ingest a huge amount of tweets different .json files are created by Hadoop.



After creating and ingesting the raw data, we need to promote and transform the data from the raw layer to the standard layer. The first thing to do is to read the raw DataFrame in .json format, inferring the schema, and storing it in a new DataFrame called tweets_raw. Python can be used as an API for Spark, which makes Spark a perfect fit.

2.2 Read Raw DataFrame

We can infer the schema of the underlying json files by setting this option during the reading operation.
This is not recommended in production workloads as is very expensive (Spark will scan all the files in order to determine all the columns)

In [55]:
```
tweets_raw = spark.read.option("inferSchema","true")\
                       .option("recursiveFileLookup", "true")\
                       .json("hdfs://localhost:9000/datalake/raw/twitter/Covid19/")

tweets_raw.limit(5).toPandas()
```

The inferred schema is not as precise as we would like. Therefore, it is needed to set up a new one during data reading with more columns and more information. In this schema we used the 2.5 point in the "Twitter - RAW to STD - DataFrames" doc but with some

modifications as for answering some questions more information was required (retweet_count, favorite_count, retweeted url).

```
retweeted_status struct <
            quot_count:int,
            reply_count:int,
            retweet_count:int,
            favorite_count:int,
            user:array<struct<url:string>>
            >,
```

Once we have read all the data, we transform it by using pyspark.sql.Functions library transforming the column created_at with its proper timestamp and also creating “year” and “dt” column with the year and date respectively.

```
In [48]: import pyspark.sql.functions as F
tweets_std = tweets_raw\
             .withColumn("created_at",F.to_timestamp(F.col("created_at"),"EEE MMM dd HH:mm:ss ZZZZZ yyyy"))\
             .withColumn("year",F.year("created_at"))\
             .withColumn("dt",F.to_date("created_at"))

tweets_std.limit(5).toPandas()
```

The last step is to promote the data, writing and creating this new DataFrame in our standard layer with a parquet format using the same data folder structure.

Code:

```
In [23]: (tweets_std.coalesce(1)
          .write
          .partitionBy("year","dt")
          .mode("overwrite")
          .parquet("hdfs://localhost:9000/datalake/std/twitter/Covid19/"))
```

Data Structure:

/datalake/std/twitter/Covid19 Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	osbdet	hadoop	0 B	Dec 09 23:33	3	128 MB	_SUCCESS
drwxr-xr-x	osbdet	hadoop	0 B	Dec 09 23:33	0	0 B	year=2021

Parquet format:

/datalake/std/twitter/Covid19/year=2021/dt=2021-12-08 Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	osbdet	hadoop	3.84 MB	Dec 09 23:33	3	128 MB	part-00000-5d1dbf37-d8fe-48e1-b67e-7a8ad21e850b.c000.snappy.parquet

Promoting by month:

Now we can promote a single day, a month or even a whole year.

```
: # Change this date according to your data in HDFS

  #promote_raw2std("2021/12/06")

  promote_raw2std("2021/12")

  #promote_raw2std("2021")
```

Storage format

As we showed before, we store all the data in parquet; good for querying, good compression and column-based. Resulting in the perfect fit.

Data Processing

Processing the data

For processing we use Spark, using a batch processing as we are reading and getting insights from the data once we stopped the ingestion. The first thing to do is read the data in parquet format, creating the DataFrame as we are working using the DataFrames of the structured APIs (High level APIs).

```
In [211]: tweets = (spark.read
                    .parquet("hdfs://localhost:9000/datalake/std/twitter/Covid19/year=2021"))
```

Once we have created the DataFrame we are able to analyze and to apply new functions to extract insights and information from the DataFrame.

Querying of data

We are querying using Python language as we are using High-Level APIs DataFrame.

Spark notebooks

All the analytics are presented in the potential analysis point with its code and results.

Project Results

Insights

Our goal was to identify what kind of impact Twitter can have on shaping people's opinion on Covid-19.

Firstly, we can argue that the data we gathered highlights different patterns about Twitter's usage depending on language: the German speaking population in Switzerland and Germany is less active than the English, Spanish or French speaking population when it comes to covid (Appendix, Figure 2). Our hypothesis, given our side knowledge, is that the German speaking population uses other platforms such as Facebook, YouTube and Instagram in order to share their feelings, opinions and facts about covid.
Secondly, our analysis shows that tweets related to Omicron were considerably less factual and labelled more times positively in the polarity column than the rest of tweets. This is important to point out, because the majority of tweets fell into the neutral category (Appendix, Figure 13).
Thirdly, the data we ingested indicates that there is a greater likelihood to make a top tweet, if the tweet is conveying a message of frustration, indignation, injustice, than a neutral informative one or a positive one inspiring hope. Since 7 out of the 10 tweets that obtained most retweets from our ingested data reflect the latter, it becomes possible to make the hypothesis that a tweet is more likely to go viral with respect to covid if it conveys a rather negative sentiment. (Appendix 11 + 12).

Fourthly, our ingested data reveals that the accounts that made the most tweets (that were the most active) were informational/political accounts. Yet, the ones that had most mentions, were accounts with large audiences (more than 10,000 followers) that tweeted a lot and whose tweets related to covid conveyed frustration or disappointment towards political elites decisions, whether those were about vaccines, about passports, about the management of the covid situation (Appendix, Figure 3 + 4).

Finally, we examined that most tweets about Covid these days use the words: 'vaccine', 'vaccination', 'vaccinated'. If we link these three most used words to the sentiment analysis, we discern that these words are mostly used in positive tweets. We can conclude that most accounts on Twitter now are talking positive about the vaccination instead of negative like the 'Anti-vax movement' suggests. However, keeping in mind that the vaccination rate in the UK (75%) and the US (72%) could be better, they can only hope that positive discussions or posts like these can help the motivation (https://ourworldindata.org/covid-vaccinations). This gets further emphasised by analysing the most used words not related to vaccine, where many tweets talk about Omicron (Appendix, Figure 9). Yet this is another proof that the broad population encourages vaccination, likewise confirmed by the most popular words. Thus, this new variant opens new discussions on Twitter, which could lead to increased vaccination rates (Appendix, Figure 5 + 11).

Conclusions

The Covid-19 vaccination divides the world's population into supporters and opponents. We wanted to gather more information about how users are expressing their point of view on Twitter by analysing the hashtags used. The ingested data emphasises that Twitter's role on how people see covid is similar to that of a catalyst, reinforcing the feeling of Twitter users. It can be noticed by analysing the tweets with more retweets, favorites, the accounts that published the most, the language used and our sentiment analysis, that the platform enables the diffusion of content which has a higher likelihood to become viral. Reactions are created mainly if tweets are in English and if they have the intention to criticize, blame the sanitary situation, vaccines or policymakers. Lastly, in a further research on the same topic about how Twitter's impact on how people may perceive Covid-19 vaccination, we would suggest to ingest data in longer periods, to obtain greater depth about our insights.

Appendix & References

Figure 1: Total number of tweets

```
In [10]: tweets = (spark.read
                       .parquet("hdfs://localhost:9000/datalake/std/twitter/Covid19/year=2021"))

In [11]: total = tweets.count()
         total

Out[11]: 33407
```

Figure 2: Tweets per language distribution

```
In [13]: from pyspark.sql.functions import *

df = (tweets
      .groupBy("lang")
      .agg(count("*").alias("total")))

df.toPandas()
```

	lang	total
0	en	26903
1	de	408
2	es	3088
3	zh	115
4	fr	2893

Figure 3: Top users with more active (more tweets published)

```
In [14]: df = (tweets
           .groupBy("user.screen_name")
           .agg(max("user.statuses_count").alias("tweets_posted"))
           .orderBy(desc("tweets_posted"))
           .limit(10))
df.toPandas()
```

	screen_name	tweets_posted
0	CaraotaDigital	6826888
1	la_patilla	6053684
2	TomthunkitsMind	3069786
3	sectest9	2525955
4	filafresh	2425011
5	eazeee2004	1867318
6	robinsnewswire	1852335
7	stephenoflyf	1850502
8	chidambara09	1763815
9	Afropages	1722637

Figure 4A: Top users with more followers

```
In [20]: df = (tweets
               .groupBy("user.screen_name")
               .agg(max("user.followers_count").alias("followers_count"))
               .orderBy(desc("followers_count"))
               .limit(10))
         df.toPandas()
```

Out[20]:

	screen_name	followers_count
0	washingtonpost	18516190
1	ndtv	16423882
2	BBCNews	12955319
3	el_pais	8218374
4	business	7742074
5	la_patilla	7216411
6	hrw	4817772
7	FraseSimple	4467796
8	CNBC	4422445
9	TODAYshow	4202788

Figure 4B: Top users with more mentions

```
In [21]: df = (tweets
               .select(explode("entities.user_mentions.screen_name").alias("user"))
               .groupBy(lower("user"))
               .agg(count("*").alias("mentions"))
               .orderBy(desc("mentions"))
               .limit(10))
         df.toPandas()
```

Out[21]:

	lower(user)	mentions
0	essexpr	469
1	josplaysenpai	381
2	rwmalonemd	279
3	spectatorindex	261
4	gnev2	243
5	disclosetv	213
6	dehennadavison	209
7	prisonplanet	194
8	chelsisright	191
9	simonjamesjupp	191

Figure 5: Top popular hashtags

In [23]:
```
df = (tweets
      .select(explode("entities.hashtags.text").alias("hashtag"))
      .groupBy(lower("hashtag").alias("hashtag"))
      .agg(count("*").alias("total"))
      .orderBy(desc("total"))
      .limit(10))

df.toPandas()

# to normalize (upper & lower case version of the same hashtag)
#.groupBy(lower("hashtag").alias("hashtag"))
```

Out[23]:

	hashtag	total
0	covid19	456
1	omicron	231
2	zusammengegencorona	143
3	vaccine	127
4	pfizer	115
5	covidvaccine	102
6	vaccination	93
7	covid	70
8	munich	56
9	covidvaccination	47

Figure 6: Top popular cashtags

In [24]:
```
df = (tweets
    .select(explode("entities.symbols.text").alias("cashtag"))
    .groupBy(upper("cashtag").alias("cashtag"))
    .agg(count("*").alias("total"))\
    .orderBy(desc("total"))
    .limit(10))

df.toPandas()
```

Out[24]:

	cashtag	total
0	PFE	89
1	BNTX	24
2	NVAX	10
3	SPY	5
4	OCGN	4
5	ES_F	3
6	SPX	3
7	BTNX	2
8	QQQ	2
9	NRXP	2

Figure 7: Hashtag per day and hour:

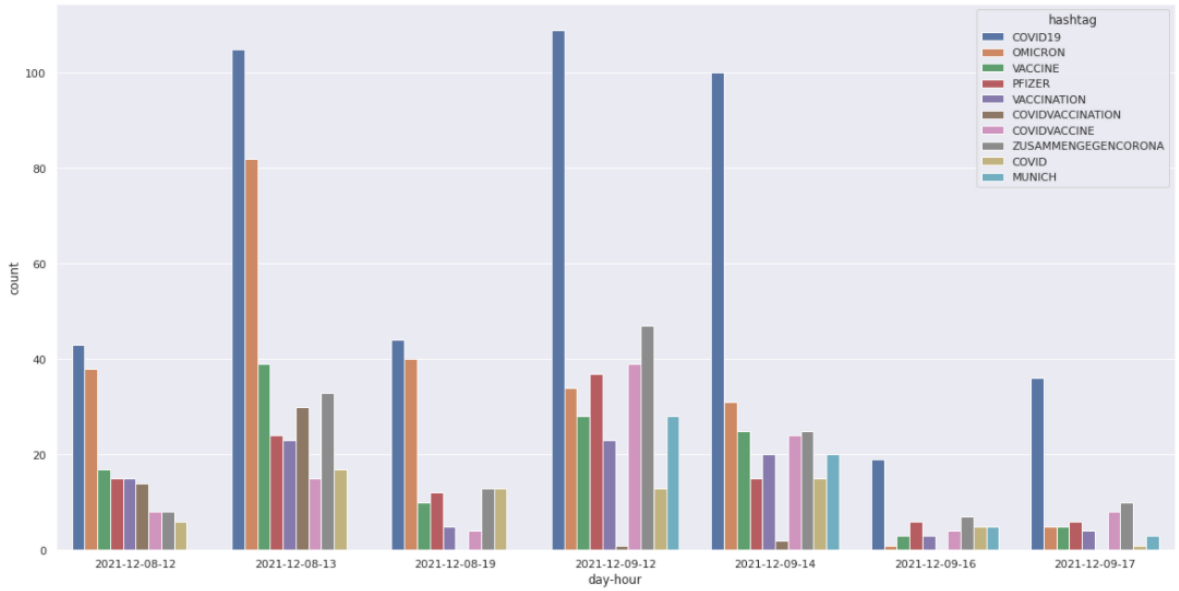


Figure 8: Average number of hashtags/cashtags

Max and average number of hashtags

```
select max(size(entities.hashtags)) as max,
       avg(size(entities.hashtags)) as average
from tweets
```

```
In [55]: (tweets.select(
                max(size("entities.hashtags")).alias("max"),
                avg(size("entities.hashtags")).alias("average")
)).toPandas()
```

Out[55]:

	max	average
0	11	0.13581

Figure 9: Top popular words

```
In [186]: stop_words = list(stopwords.words('english')) + list(stopwords.words('spanish'))

          stop_words_final = stop_words.append("rt")
          stop_words_final2 = stop_words.append("-")
          words_in_tweet = [tweet.lower().split() for tweet in tweetsdf["text"]]

          all_words_no_urls = list(itertools.chain(*words_in_tweet))
          words2 = [word for word in all_words_no_urls if word not in stop_words]
          counts_no_urls = collections.Counter(words2)
          counts_no_urls.most_common()

Out[186]: [('vaccine', 11033),
           ('vaccination', 2395),
           ('vaccinated', 2152),
           ('covid', 2103),
           ('people', 1802),
           ('passports', 1765),
           ('omicron', 1753),
           ('get', 1681),
           ('pfizer', 1565),
           ('covid-19', 1243),
           ('et', 1232),
           ('mandate', 1175),
           ('doses', 1175),
           ('vacuna', 1110),
           ('says', 1105),
           ('&amp;', 1034),
           ('breaking:', 993),
           ('des', 904),
           ('fully', 826),
```

Figure 10: Average number of words per tweet

```
In [53]: tweets.select(avg(size(split("text", " "))).alias("avg_words")).toPandas()
```

Out[53]:

	avg_words
0	18.752597

For the next two points we had to change the schema in the RAW to STD DataFrames to include some values needed. We added the following code:

```
retweeted_status struct <
            quot_count:int,
            reply_count:int,
            retweet_count:int,
            favorite_count:int,
            user:array<struct<url:string>>
            >,
```

Figure 11: Top tweets with more favorites

In [154]:
```
df = (tweets
          .groupBy("text")
          .agg(max("retweeted_status.favorite_count").alias("favorite_count"))
          .orderBy(desc("favorite_count"))
          .limit(10))
df.toPandas()
```

Out[154]:

	text	favorite_count
0	RT @SamIAm2021MD: I am blown away by this COVID vaccine video, one of the coolest things I have seen in a long time https://t.co/yREIETqtWh	504255
1	RT @ChelsIsRight: Not the Impossible Vaccine	171718
2	RT @TheKaranMenon: The Omnicron variant is just another example of how COVID will be prolonged if rich nations and Big Pharma keep blocking…	118112
3	RT @aggonzalez03: can we just reflect on the fact that there are people buying fake vaccination cards so they can travel, but they're the s…	98704
4	RT @abbygov: finance bros are so far removed from reality and like... ethics. i was venting about the US not lifting vaccine patents earlie…	79930
5	RT @RealCandaceO: Dear @innoutburger and @ChickfilA, \nThank you for serving the unvaccinated AND the vaccinated. \nBusiness without discrim…	75470
6	RT @NBSaphierMD: I'm not sure who needs to hear this but, opposing vaccine mandates is not equivalent to opposing vaccines.	61020
7	RT @POTUS: The best protection against Omicron is simple: Get fully vaccinated. Get a booster shot.	60990
8	RT @narendramodi: Every Indian would be proud of today's record vaccination numbers. \n\nI acknowledge our doctors, innovators, administrator…	55577
9	RT @GNev2: He just needed to apologise.Instead he gave us more lies, threw his team under a bus,vaccine passports,introduced the potential…	55317

Figure 12: Top tweets with more retweets

In [151]:
```
df = (tweets
          .groupBy("text")
          .agg(max("retweeted_status.retweet_count").alias("retweet_count"))
          .orderBy(desc("retweet_count"))
          .limit(10))
df.toPandas()
```

Out[151]:

	text	retweet_count
0	RT @SamIAm2021MD: I am blown away by this COVID vaccine video, one of the coolest things I have seen in a long time https://t.co/yREIETqtWh	173661
1	RT @TheKaranMenon: The Omnicron variant is just another example of how COVID will be prolonged if rich nations and Big Pharma keep blocking…	43736
2	RT @aggonzalez03: can we just reflect on the fact that there are people buying fake vaccination cards so they can travel, but they're the s…	37568
3	RT @GBNEWS: 'They aren't going to publish their findings, they are concerned about losing research money' \n\nDr Aseem Malhotra reveals a car…	22355
4	RT @DrEliDavid: 🚨 Breaking: Israeli Ministry of Health announced today that it will soon approve the 4th vaccine shot.\n\nIt will mean that a…	20194
5	RT @ChelsIsRight: Not the Impossible Vaccine	20092
6	RT @KonstantinKisin: You're struggling to understand why some people are vaccine hesitant. The "let me help you" megathread:\n\nImagine you're…	17769
7	RT @ake2306: 9 months ago today I collapsed at home 14 days after my 1st AstraZeneca. I'm here when I shouldn't be. I lost my left leg from…	16291
8	RT @PhilsandJenn: I am 5 hours into my 12 hour shift and have already treated 3 vax injuries. Heart attack, 36y/o, Stroke, 44y/o and a lowe…	12612
9	RT @mtgreenee: American Heart Association:\nWe conclude that the mRNA vacs dramatically increase inflammation on the endothelium and T cell…	11790

Figure 13: Tweet sentiment analysis

We were able to perform a sentiment analysis using some new libraries and modifying a python script found.
https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d

Our results show that:

```
total number:  33407
positive number:  9212
negative number:  9591
neutral number:  14604
```

Also per day-hour and hashtag

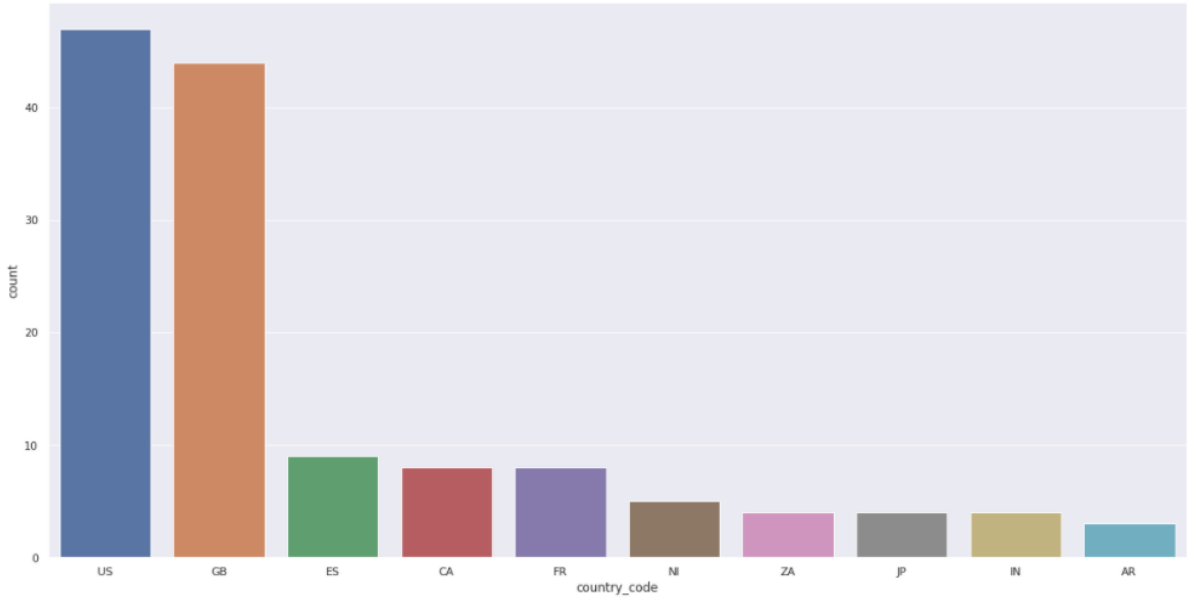
	day-hour	hashtag	polarity	subjectivity	polarity_category	subjectivity_category
0	2021-12-08-12	PFIZER	0.036458	0.119792	neutral	more factual
1	2021-12-08-12	COVIDVACCINE	0.003125	0.145833	neutral	more factual
2	2021-12-08-12	COVID19	0.092311	0.292328	positive	more factual
3	2021-12-08-12	VACCINE	0.088725	0.190441	positive	more factual
4	2021-12-08-12	COVIDVACCINATION	0.092857	0.371429	positive	more factual
5	2021-12-08-12	COVID	0.022727	0.075758	neutral	more factual
6	2021-12-08-12	OMICRON	0.086157	0.220033	positive	more factual
7	2021-12-08-13	ZUSAMMENGEGENCORONA	0.106250	0.218750	positive	more factual
8	2021-12-08-13	COVIDVACCINE	0.124697	0.327879	positive	more factual
9	2021-12-08-13	COVIDVACCINATION	0.117778	0.364444	positive	more factual
10	2021-12-08-13	OMICRON	0.099495	0.216321	positive	more factual
11	2021-12-08-13	COVID19	0.157791	0.291777	positive	more factual
12	2021-12-08-13	COVID	0.093939	0.284659	positive	more factual
13	2021-12-08-13	VACCINATION	0.051010	0.312626	positive	more factual
14	2021-12-08-13	VACCINE	0.195877	0.314569	positive	more factual
15	2021-12-08-13	PFIZER	0.010833	0.021667	neutral	more factual
16	2021-12-08-19	VACCINATION	0.000000	0.000000	neutral	more factual
17	2021-12-08-19	OMICRON	0.330894	0.516455	positive	more personal opinion
18	2021-12-08-19	COVIDVACCINE	0.062500	0.333333	positive	more factual
19	2021-12-08-19	COVID	0.007639	0.178472	neutral	more factual
20	2021-12-08-19	PFIZER	-0.010417	0.363542	neutral	more factual
21	2021-12-08-19	VACCINE	-0.135000	0.305000	negative	more factual
22	2021-12-08-19	COVID19	0.110942	0.411648	positive	more factual
23	2021-12-09-12	ZUSAMMENGEGENCORONA	0.000000	0.000000	neutral	more factual
24	2021-12-09-12	MUNICH	0.000000	0.000000	neutral	more factual
25	2021-12-09-12	VACCINATION	-0.036185	0.187879	neutral	more factual
26	2021-12-09-12	COVIDVACCINATION	0.000000	0.000000	neutral	more factual
27	2021-12-09-12	COVID19	0.085372	0.261531	positive	more factual
28	2021-12-09-12	COVID	0.001042	0.302083	neutral	more factual
29	2021-12-09-12	OMICRON	0.110687	0.323619	positive	more factual
30	2021-12-09-12	VACCINE	0.061932	0.152471	positive	more factual
31	2021-12-09-12	PFIZER	0.007576	0.199621	neutral	more factual
32	2021-12-09-12	COVIDVACCINE	0.000000	0.020513	neutral	more factual
33	2021-12-09-14	PFIZER	0.136068	0.192523	positive	more factual
34	2021-12-09-14	OMICRON	0.171567	0.446922	positive	more factual
35	2021-12-09-14	COVIDVACCINATION	0.050000	0.200000	positive	more factual

36	2021-12-09-14	COVID	0.064015	0.091162	positive	more factual
37	2021-12-09-14	COVIDVACCINE	0.077174	0.147826	positive	more factual
38	2021-12-09-14	ZUSAMMENGEGENCORONA	0.000000	0.000000	neutral	more factual
39	2021-12-09-14	MUNICH	0.000000	0.000000	neutral	more factual
40	2021-12-09-14	VACCINE	0.039810	0.193952	neutral	more factual
41	2021-12-09-14	COVID19	0.083225	0.230808	positive	more factual
42	2021-12-09-14	VACCINATION	0.133333	0.277083	positive	more factual
43	2021-12-09-16	COVIDVACCINE	0.000000	0.000000	neutral	more factual
44	2021-12-09-16	VACCINATION	0.033333	0.333333	neutral	more factual
45	2021-12-09-16	COVID	-0.100000	0.066667	negative	more factual
46	2021-12-09-16	PFIZER	-0.166667	0.341667	negative	more factual
47	2021-12-09-16	VACCINE	0.125000	0.125000	positive	more factual
48	2021-12-09-16	MUNICH	0.000000	0.000000	neutral	more factual
49	2021-12-09-16	OMICRON	0.000000	0.000000	neutral	more factual
50	2021-12-09-16	COVID19	0.001573	0.334518	neutral	more factual
51	2021-12-09-17	OMICRON	0.084375	0.546875	positive	more personal opinion
52	2021-12-09-17	COVID	0.000000	0.000000	neutral	more factual
53	2021-12-09-17	PFIZER	0.062500	0.111111	positive	more factual
54	2021-12-09-17	COVID19	0.061251	0.332767	positive	more factual
55	2021-12-09-17	VACCINATION	0.000000	0.000000	neutral	more factual
56	2021-12-09-17	VACCINE	0.000000	0.013333	neutral	more factual
57	2021-12-09-17	MUNICH	0.000000	0.000000	neutral	more factual
58	2021-12-09-17	COVIDVACCINE	0.068750	0.029167	positive	more factual

Figure 14: Tweet geolocation analysis

Count of tweets from each country in which geolocation sharing is enabled

	country_code	country	count
0	US	United States	47
1	GB	United Kingdom	44
2	ES	España	9
3	CA	Canada	8
4	FR	France	8
5	NI	Nicaragua	5
6	ZA	South Africa	4
7	JP	日本	4
8	IN	India	4
9	AR	Argentina	3


40
30
20
10
0
count
US
GB
ES
CA
FR
NI
ZA
JP
IN
AR
country_code