Project report

Timothée Ly, Oliver Chin Wee Gui, Erik Kongpachith
Group 7

1 Project description

Covid-19 has been a major upset worldwide for the past 3 years. For this project, we used the Twitter API together with some of the frameworks and packages covered in the course to perform an analysis on tweets data regarding Covid-19. In addition, we also implemented a simple streaming application for displaying the latest tweets about Covid-19.

2 Dataset

The dataset used for this project is made up of roughly 3 million Twitter Covid-19 tweets posted between December 2019 to May 2022 (30 months period) collected with the Twitter API. The file used for the storage is a `.csv` extension. Each row contains the following fields :

- `text` - (not nullable) - The text field, core of the tweet. That's from there that one can extract information such as the other hashtags used.
- `id` - (not nullable) - The unique identifier for each tweet
- `author_id` - (not nullable) -The unique identifier of the tweet's author.
- `lang` - (not nullable) - It's always useful to know which language is used in the tweet.
- `created_at` - (not nullable) - Date of creation of the tweet, formatted along *TwitterAPI* style.
- `place_id` - (nullable) - Indicates the place where the tweet was posted with a unique id.
- `country` - (nullable) The country name written in Swedish (due to ip located in Sweden)
- `country_code` - (nullable) - The country code which makes it easier to use than the country name.

You can download the dataset here (expires on November 4th. Send an email if any issue arises.)

3 Method

The reader is invited to have the notebook open while reading this report. The notebook contains comments about code implementation details that won't be discussed here. On the other hand, this report cares more about providing an insightful view of the project and the difficulties encountered. Both documents follow the same chronology, the reading is linear. The notebook cells can be executed but not all, be sure to read the first section of the notebook about how to use it.

3.1 An Overview of the Project

Before even scraping any data, we deemed interesting to assess the number of tweets published each month by users about Covid-19.

The TwitterAPI provides an endpoint to do such operations : **"tweets/count/all"**. We pass it the following query : **"(#Covid19 OR #coronavirus OR #Covid-19 OR #covid)"** that defines which hashtag we should find in the `text` field. Note that the query is case insensitive which reduces the number of hashtags to place in it.

3.2 Tweets Scraping Strategy

With the Academic Research access provided by Twitter, we are allowed to scrap a total of 10 M tweets per month. The first month of the study is December 2019. The last month envisioned is May 2022. It makes for 30 months. The strategy is to scrape the tweets only once and then store them into a local `dataset.csv` file to be loaded when necessary. We decided to go for 100,000 tweets per month which cumulates to 3 million. This much is more than enough to be able to extract interesting information about tendencies.

Because we can only get a certain amount of tweets per request, we use the pagination tool for twitter requests : Pagination. How does it work? When using the full-archive mode, each request can only deliver up to 500 tweets. However, if the number of results exceeds this threshold, then a token is provided to get the next batch of result, to the cost of 1 request. The library used to interface Python with the Twitter API is TwitterAPI. It offers a built-in `TwitterPager` class that we decided not to use for two reasons :

- The functionalities it offered were not totally adapted to what we needed.
- It was an interesting exercise that to develop its own paging system and not to rely on a pre-existing solution.

We created our own class called `PagerCount` that implement the paging feature of the Twitter API. `PagerCount` issues requests to the API with a number of tweets to scrape (100,000 per month for us). Each time a batch of tweets is received, it is pre-processed with *Pyspark* and then written in the designated `.csv` file.

One of the most painstaking issue was to resolve the compatibility between *Spark* dataframes and the `csv` format when writing down data. As shameful as it is to admit, the 3 million tweets were scraped 3 times in their entirety (beyond of the test and try) as the `csv` file was corrupted at the end (meanwhile the dataframe showed no sign of error). The issue was revolving around the delimiter used. `csv` file format uses by default the comma to mark the end of a field. The `text` field of the tweets scraped contained a lot of commas, the dataframe structure wouldn't mind it but when writing it down the `csv` would have the `text` field overflow on the other fields. Fixing this issue amounted to replace the delimiter by the pipe | as it is much less used than the comma. Before being written down, we used a trimmer function to remove all the pipes that would be part of the `text` field as a precautionary measure. A similar issue was the newline character '/n' that would create a new row to the `csv` file. Same solution : trim '/n' occurrences out of the `text` field.

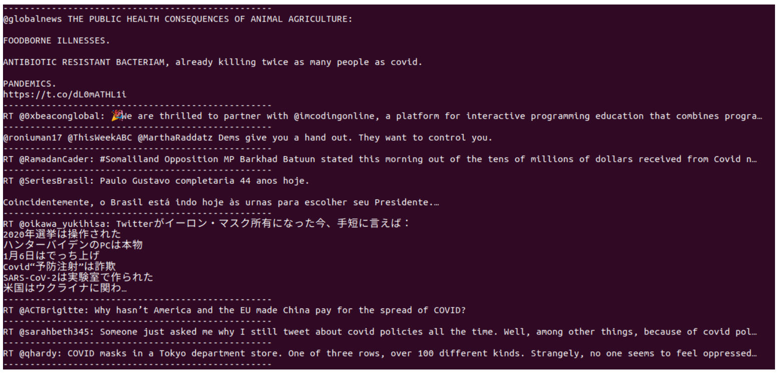
3.3 Statistics and Analysis

In the first part of the project, we have used the Twitter API and Spark SQL to perform an analysis on tweets regarding Covid-19. This was implemented on a Jupyter Notebook using the Twitter and Spark libraries in Python. Using Spark SQL, we created a temporary view of the DataFrame and ran an SQL query on it to find the countries with the highest number of tweets related to Covid-19. We discovered that the top 3 countries with the most tweets were the USA, Great Britain, and Canada. Furthermore, to apply transformations to columns in our DataFrame, we created column-based functions using the udf function from the Spark SQL library. We created such functions to clean text data and create a sentiment label from the text data using VADER - a pre-trained sentiment analysis model. After applying the required transformations, we created visualizations for tweets from each of the top 3 countries. The visualizations we chose were a word cloud, a bar graph displaying the number of tweets with positive, negative, and neutral sentiments, and a line chart showing the number of Covid-19 related tweets over time.

In addition, a correlation analysis was done on the monthly number of covid 19 related tweets, number of recorded covid 19 cases and death in the world for the past 30 months. Scatterplots and Pearsons correlation coefficient were computed to determine if there were any positive or negative correlation between the three variables.

3.4 Streaming tweets

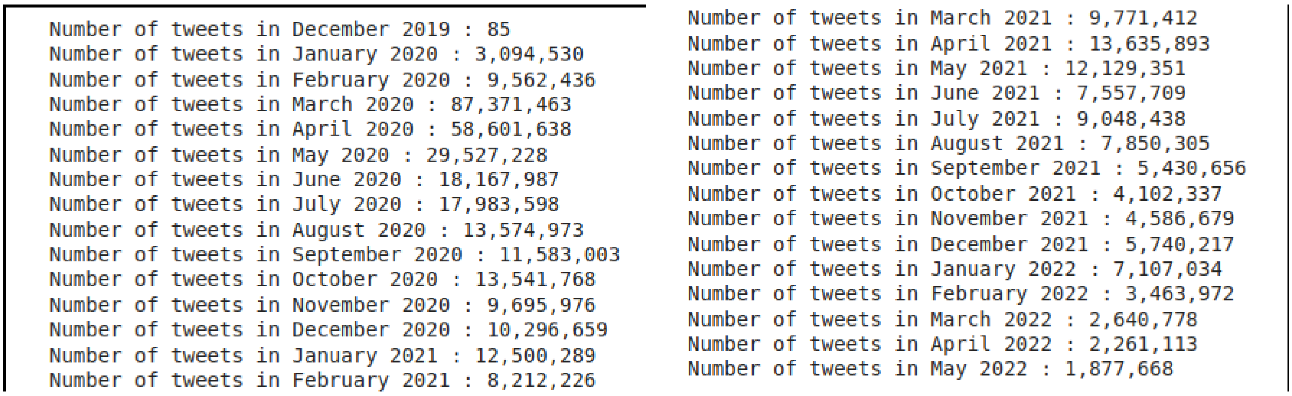
In the second part of the project, we used Python, Kafka and Twitter API to create a simple streaming application. A simple tweets streaming application was implemented using Python and the Tweepy V2 library. A TweetStreaming class is used with the StreamingClient object. When created, it connects to the twitter server using an authentication token and scraps tweets based on filtering on rules. The rule is set to be tweets which contains the keyword "covid" and each received tweet is printed out to terminal with 5 second interval.



```
--------------------------------------------------
@globalnews THE PUBLIC HEALTH CONSEQUENCES OF ANIMAL AGRICULTURE:

FOODBORNE ILLNESSES.

ANTIBIOTIC RESISTANT BACTERIAM, already killing twice as many people as covid.

PANDEMICS.
https://t.co/dL0mATHL1i
--------------------------------------------------
RT @0xbeaconglobal: 🎉We are thrilled to partner with @imcodingonline, a platform for interactive programming education that combines progra…
--------------------------------------------------
@roniuman17 @ThisWeekABC @MarthaRaddatz Dems give you a hand out. They want to control you.
--------------------------------------------------
RT @RamadanCader: #Somaliland Opposition MP Barkhad Batuun stated this morning out of the tens of millions of dollars received from Covid n…
--------------------------------------------------
RT @SeriesBrasil: Paulo Gustavo completaria 44 anos hoje.

Coincidentemente, o Brasil está indo hoje às urnas para escolher seu Presidente.…
--------------------------------------------------
RT @oikawa_yukihisa: Twitterがイーロン・マスク所有になった今、手短に言えば：
2020年選挙は操作された
ハンターバイデンのPCは本物
1月6日はでっち上げ
Covid“予防注射”は詐欺
SARS-CoV-2は実験室で作られた
米国はウクライナに関わ…
--------------------------------------------------
RT @ACTBrigitte: Why hasn’t America and the EU made China pay for the spread of COVID?
--------------------------------------------------
RT @sarahbeth345: Someone just asked me why I still tweet about covid policies all the time. Well, among other things, because of covid pol…
--------------------------------------------------
RT @qhardy: COVID masks in a Tokyo department store. One of three rows, over 100 different kinds. Strangely, no one seems to feel oppressed…
--------------------------------------------------
```

Figure 1: Sample tweets streaming application output)

4 Results



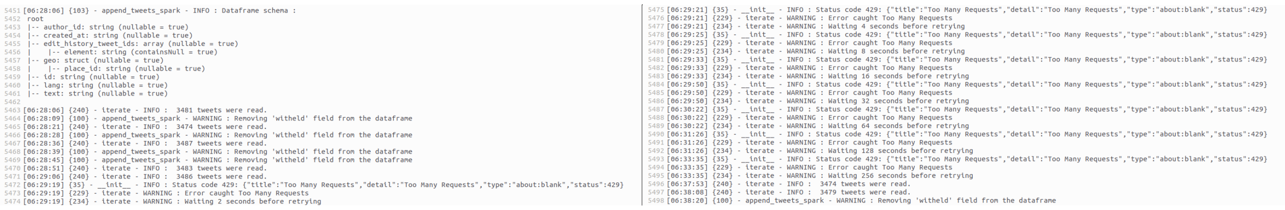
```
Number of tweets in December 2019 : 85
Number of tweets in January 2020 : 3,094,530
Number of tweets in February 2020 : 9,562,436
Number of tweets in March 2020 : 87,371,463
Number of tweets in April 2020 : 58,601,638
Number of tweets in May 2020 : 29,527,228
Number of tweets in June 2020 : 18,167,987
Number of tweets in July 2020 : 17,983,598
Number of tweets in August 2020 : 13,574,973
Number of tweets in September 2020 : 11,583,003
Number of tweets in October 2020 : 13,541,768
Number of tweets in November 2020 : 9,695,976
Number of tweets in December 2020 : 10,296,659
Number of tweets in January 2021 : 12,500,289
Number of tweets in February 2021 : 8,212,226
```

(a)

```
Number of tweets in March 2021 : 9,771,412
Number of tweets in April 2021 : 13,635,893
Number of tweets in May 2021 : 12,129,351
Number of tweets in June 2021 : 7,557,709
Number of tweets in July 2021 : 9,048,438
Number of tweets in August 2021 : 7,850,305
Number of tweets in September 2021 : 5,430,656
Number of tweets in October 2021 : 4,102,337
Number of tweets in November 2021 : 4,586,679
Number of tweets in December 2021 : 5,740,217
Number of tweets in January 2022 : 7,107,034
Number of tweets in February 2022 : 3,463,972
Number of tweets in March 2022 : 2,640,778
Number of tweets in April 2022 : 2,261,113
Number of tweets in May 2022 : 1,877,668
```

(b)

Figure 2: Monthly count of tweets about Covid-19



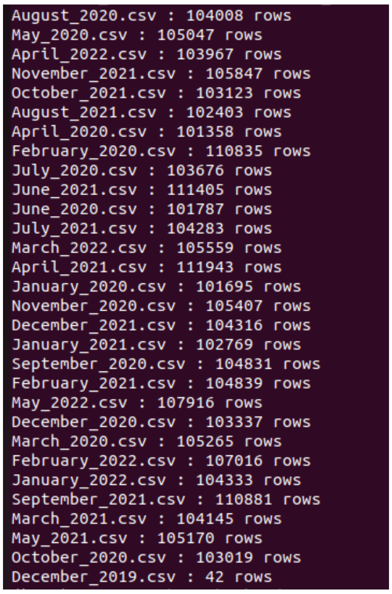
```
[06:28:06] {103} - append_tweets_spark - INFO : Dataframe schema :
 root
 |-- author_id: string (nullable = true)
 |-- created_at: string (nullable = true)
 |-- edit_history_tweet_ids: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- geo: struct (nullable = true)
 |    |-- place_id: string (nullable = true)
 |-- id: string (nullable = true)
 |-- lang: string (nullable = true)
 |-- text: string (nullable = true)

[06:28:06] {240} - iterate - INFO :  3481 tweets were read.
[06:28:09] {100} - append_tweets_spark - WARNING : Removing 'witheld' field from the dataframe
[06:28:21] {240} - iterate - INFO :  3474 tweets were read.
[06:28:28] {100} - append_tweets_spark - WARNING : Removing 'witheld' field from the dataframe
[06:28:36] {240} - iterate - INFO :  3487 tweets were read.
[06:28:39] {100} - append_tweets_spark - WARNING : Removing 'witheld' field from the dataframe
[06:28:45] {100} - append_tweets_spark - WARNING : Removing 'witheld' field from the dataframe
[06:28:51] {240} - iterate - INFO :  3483 tweets were read.
[06:29:06] {240} - iterate - INFO :  3486 tweets were read.
[06:29:19] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:29:19] {229} - iterate - WARNING : Error caught Too Many Requests
[06:29:19] {234} - iterate - WARNING : Waiting 2 seconds before retrying
```

(a)

```
[06:29:21] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:29:21] {229} - iterate - WARNING : Error caught Too Many Requests
[06:29:21] {234} - iterate - WARNING : Waiting 4 seconds before retrying
[06:29:25] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:29:25] {229} - iterate - WARNING : Error caught Too Many Requests
[06:29:25] {234} - iterate - WARNING : Waiting 8 seconds before retrying
[06:29:33] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:29:33] {229} - iterate - WARNING : Error caught Too Many Requests
[06:29:33] {234} - iterate - WARNING : Waiting 16 seconds before retrying
[06:29:50] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:29:50] {229} - iterate - WARNING : Error caught Too Many Requests
[06:29:50] {234} - iterate - WARNING : Waiting 32 seconds before retrying
[06:30:22] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:30:22] {229} - iterate - WARNING : Error caught Too Many Requests
[06:30:22] {234} - iterate - WARNING : Waiting 64 seconds before retrying
[06:31:26] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:31:26] {229} - iterate - WARNING : Error caught Too Many Requests
[06:31:26] {234} - iterate - WARNING : Waiting 128 seconds before retrying
[06:33:35] {35} - __init__ - INFO : Status code 429: {"title":"Too Many Requests","detail":"Too Many Requests","type":"about:blank","status":429}
[06:33:35] {229} - iterate - WARNING : Error caught Too Many Requests
[06:33:35] {234} - iterate - WARNING : Waiting 256 seconds before retrying
[06:37:53] {240} - iterate - INFO :  3474 tweets were read.
[06:38:08] {240} - iterate - INFO :  3479 tweets were read.
[06:38:20] {100} - append_tweets_spark - WARNING : Removing 'witheld' field from the dataframe
```

(b)

Figure 3: A few lines of the `.log` file



```
August_2020.csv : 104008 rows
May_2020.csv : 105047 rows
April_2022.csv : 103967 rows
November_2021.csv : 105847 rows
October_2021.csv : 103123 rows
August_2021.csv : 102403 rows
April_2020.csv : 101358 rows
February_2020.csv : 110835 rows
July_2020.csv : 103676 rows
June_2021.csv : 111405 rows
June_2020.csv : 101787 rows
July_2021.csv : 104283 rows
March_2022.csv : 105559 rows
April_2021.csv : 111943 rows
January_2020.csv : 101695 rows
November_2020.csv : 105407 rows
December_2021.csv : 104316 rows
January_2021.csv : 102769 rows
September_2020.csv : 104831 rows
February_2021.csv : 104839 rows
May_2022.csv : 107916 rows
December_2020.csv : 103337 rows
March_2020.csv : 105265 rows
February_2022.csv : 107016 rows
January_2022.csv : 104333 rows
September_2021.csv : 110881 rows
March_2021.csv : 104145 rows
May_2021.csv : 105170 rows
October_2020.csv : 103019 rows
December_2019.csv : 42 rows
```

Figure 4: The number of rows per month (the result for December 2019 was expected.)

5 How to run

- `Scraping the dataset` - Follow all the steps in the provided Jupyter notebook.
- `Twitter analytics` - Follow all the steps in the provided Jupyter notebook.
- `Tweets streaming application` - Run the twitterstreaming.py program file in terminal using the command **python3 twitterstreaming.py**.