Golden Album Sentiment Analysis Report

7012DATSCI/ BIG DATA COMPUTING

December 8, 2023

Salma Louhaichy, Manel Mehemli

Supervised by

Dr. Sandra Ortega Martorell

Table of figures

Table of contents

1. Introduction:

As an effort to explore the concepts and technologies studied in the big data computing module, we will be conducting a sentiment analysis on the topic “Streaming and digital music consumption”. In this topic, we have specifically chosen to analyse the public’s reaction towards the Korean artist Jungkook’s newest album sensation “Golden”. To do that, we will be collecting data, processing it then conducting a sentiment analysis on it. Through visualization methods, we will turn the data into information and analyse the trends noticed. In addition, we will be building a dashboard for real time data monitoring.

2. Methods:

2.1. Architecture:

For this project, our aim is to scrape real time (or near real-time data) from Twitter regarding the #Jungkook_Golden album using the Nitter API. These tweets will then be pre-processed using infamous python libraries such pandas, NumPy…etc in Google Colab, which allow us to clean our data from irrelevant words, numbers, links, username tags and other things that might hinder our analysis and/or bias it in any shape of form. The now clean tweets will be put through Textblob to conduct a sentiment analysis. The results we receive will be fed to our interactive dashboard that we have built using Streamlit. Our dashboard provides a visualization of the results of the sentiment analysis on the real-time data streamed at the time of the demonstration using bar plots, graphs and showing the different tweets used for the analysis. Fig.1. below summarizes the architecture used for our project.

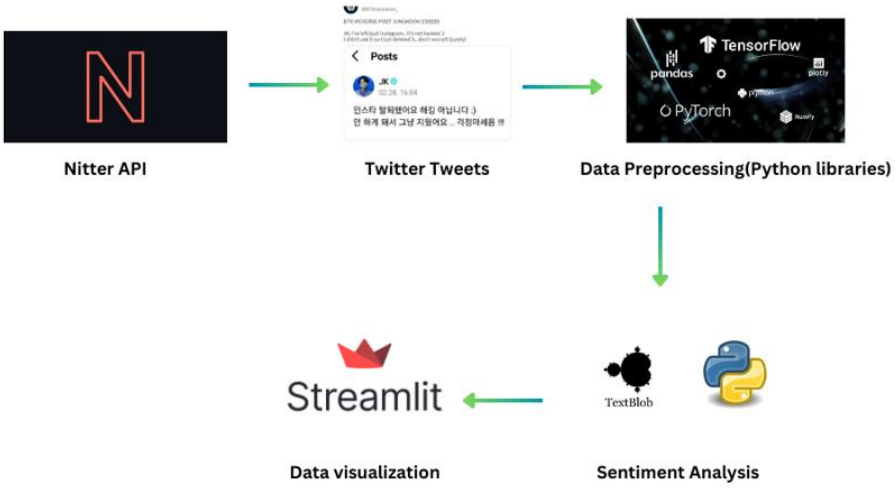

Figure 1: Architecture

2.2. Technologies used:

a. Google Colab:

Figure 2: Google Colab Logo

Due to the group work nature of this project, we found it most suitable to use Google Colab as a simpler method of sharing and updating each other about the code.

Google Colab is a cloud-based platform launched by Google to allow users to write and execute Python code in an environment powered by its browser. It contains all the commonly used Python libraries and makes it easy to install new ones in a timely manner. It offers limited CPU and GPU for its users for the free payment plan, but it was enough for our project as we were dealing with modest amounts of data (*Google Collaboratory*).

b. Python:

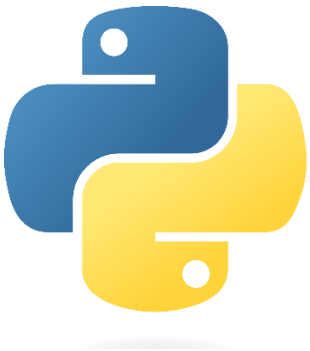

Figure 3: Python Logo

As one of the most commonly – and easier – used programming languages, Python has taken big data and analytics sectors by a storm. This is due to its readability, versatility, and flexibility. Python has many technical advantages which lured us to use it for our project:

- Diverse ecosystem of libraries: Namely Pandas, NumPy, Matplotlib…etc. They offer powerful tools for data manipulation, pre-processing, and visualization: major steps any data analysis project.
- Community support and documentation: Python has a vast community of developers who constantly contribute to the documentation, offering tutorials, identifying, and fixing bugs and proposing numerous projects using the language in data analysis, which offers us extensive support.
- Optimized Data Structures: Thanks to its libraries, Python offers its users optimized data structures and algorithms which facilitate and enhance working with larger data sets.

c. Nitter API:

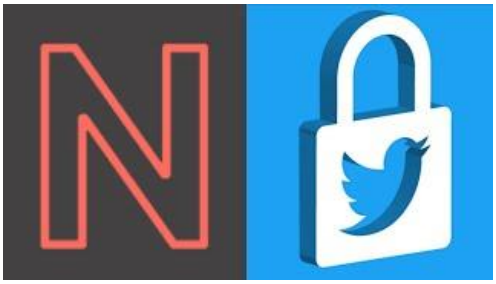

Figure 4: Nitter

As of a recent update, Twitter has put developers in the shackles of paywall. This change to the Twitter API entitles that any developer who wishes to retrieve, create, or engage with its API will have to pay a basic fee and undergo limitations. Elon Musk has

introduced rate limitations and quotas on the number of requests developers can make per day and per month depending on their payment plan. These are efforts to regulate the data use of the platform and to clean it up from the "abusive" bots and scammers (Barnes).

Unfortunately, to us students, this monetization was a big obstacle. However, we stumbled upon Nitter, an alternative front-end for Twitter which offers its users access to Twitter in a more discrete manner, without having to worry about privacy breaches or any tracking. The Nitter API acts as a bridge between Twitter and users, relieving them from the interaction with Twitter's API. Thus, we can use it to retrieve tweets using usernames, hashtags, or specific terms without being restricted to the 1500 tweet/month that Twitter's new regulations constrain.

d. TextBlob:

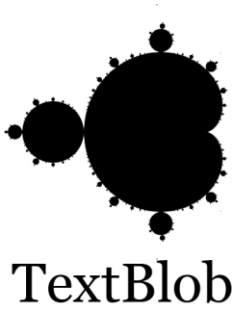

Figure 5: Textblob logo

This is a Python library used for Natural Language Processing (NLP). It is built on top of NLTK (Natural Language Toolkit) and comes with a pre-trained model for sentiment analysis. It provides a high-level interface to carry out typical activities including sentiment analysis, part-of-speech tagging, noun phrase extraction, translation, and more by streamlining a lot of NLP procedures. In addition, this library allows users a level of flexibility as they can customize its lexicon with words specific to the topic of analysis (Muhammad).

e. Streamlit:

Figure 6: Streamlit logo

For two people who don't have much experience with front-end design, Streamlit is a great tool to deploy our model in a web application. It is an open-source python library which allows users to build interfaces using few lines of code. It also offers many templates for dashboards and visualization which use python libraries such as matplotlib. We have used Github to deploy our web application for this project (*Python Tutorial*).

2.3. Initial data collection process:

As explained before, the aim of our project is to conduct a sentiment analysis on Jungkook's newest album "Golden". We have chosen Twitter as our target data collection platform due to the important number of interactions and significant activity of users that follow and support this artist and his boyband BTS. As a matter of a fact, their fans have generated over 370 million tweets using hashtags relating to the boyband in 2022 ("10 Mind-Blowing BTS Facts and Statistics"). Fig.7 depicts the distribution of the hashtags and their corresponding counts of tweets during that year.

Top 10 BTS-related hashtags in 2022

Source: **Brandwatch**

Table shows the top 10 most popular hashtags in conversations relating to BTS.
Data gathered from public posts on Twitter between Jan 1 - Dec 31 2022.

Rank (2022)	Hashtag	All Tweets
1	#bts	171,117,880
2	#bts_butter	34,343,748
3	#방탄소년단	31,630,040
4	#jimin	26,328,794
5	#bts_proof	21,856,964
6	#withyou	20,196,848
7	#btsjimin	19,005,523
8	#yettocome	15,155,579
9	#jungkook	14,440,352
10	#amas	14,349,801
Total		368,425,529

Powered by **Brandwatch Consumer Research**

Figure 7: Top 10 Bts related hashtags in 2022

This attractive flow of tweets is a gold mine to reach the goal of our project. Thus, we have timed our data collection with the first announcement of the new album "Golden", which was on October 3rd, 2023.

In order to obtain a quality batch of data to analyse, we have split the data collection process into two steps: data scraping and data preprocessing.

2.3.1. Data scrapping:

"ntscraper" is an unofficial python library which scrapes Nitter instances of tweets. It is free and open source. Its function "get_tweets ()" allows users to gather tweets either by a specific username, hashtag, term and can get user information. The user can also specify the timeframe of the tweets, location, and language among many other parameters to filter and refine the scraping.

As it is shown in Fig 8, we start by calling "Nitter ()". This function skims the 34 Nitter instances for an active, available one at that time. Once it is determined, "scraper" contains a Nitter instance ready for user. We have defined out own function "get_tweets ()". It takes as parameters the hashtag, the modality of scraping and the number of tweets we want to scrape. This function oversees scraping and storing the information in a data frame "data" that is returned by the end of the operation. The reason behind this is to specify the parameters from

the scraper and load them directly into an organized data frame to keep the process organized and easily spot any mistakes. The parameters we have deemed as relevant to our analysis are username, text, date, number of likes and number of retweets.

```
#importing the necessary libraries for data scraping
from ntscraper import Nitter
import pandas as pd
```

```
#Loading the Nitter instances
scraper = Nitter()
```

```
Testing instances: 100%|██████████| 34/34 [00:55<00:00,  1.63s/it]
```

```
[ ] #this function uses the Nitter scraper to scrape tweets from X and stores the username,
    # text of the tweet, date and number of likes in our dataframe "data"
    def get_tweets(name, modes, no):
      tweets = scraper.get_tweets(name, mode = modes, number =no)
      final_tweets = []
      for tweet in tweets["tweets"]:
          data = [tweet["user"]["username"],tweet["text"], tweet["date"],
                  tweet["stats"]["likes"]]#, tweet["stats"]["retweets"]
          final_tweets.append(data)
      data = pd.DataFrame(final_tweets, columns = ["username", "text",
                                                   "date", "Likes","Retweets"])
      return data
```

```
[ ] #Scraping data
    data = get_tweets('JungKook_GOLDEN', 'hashtag',5000)
```

```
INFO:root:Current stats for JungKook_GOLDEN: 326 tweets, 0 threads...
INFO:root:Current stats for JungKook_GOLDEN: 326 tweets, 0 threads...
INFO:root:Current stats for JungKook_GOLDEN: 332 tweets, 0 threads...
INFO:root:Current stats for JungKook_GOLDEN: 332 tweets, 0 threads...
```

Figure 8: Data scraping python code

For our batch data, we have scraped over 5000 tweets using the hashtag "#JungKook_GOLDEN". We have stored this data frame into a csv file called "tweet3.csv".

Fig 9 shows random tweets that have been scrapped. Due to the wide fanbase of this artist, we have chosen to not delimit the language of the tweets. We felt it would be fairer to scrape tweets using different languages as to not limit the reactions of the fans no matter their background, thus giving us more sentiments to analyse.

```
data
```

	username	text	date	Likes
0	@urmystar_jk	[231120] 골든 쇼케이스 완벽!✨ #JungKook_GOLDEN #Golde...	Nov 26, 2023 · 9:26 AM UTC	2231
1	@Daily_JKUpdate	📊 Jungkook's "GOLDEN" is BACK to #1 on iTunes ...	Nov 25, 2023 · 11:34 PM UTC	2659
2	@1BforJK	เพลงจองกุกบนชาร์ต Spotify global chart (25/11/...	Nov 26, 2023 · 10:00 AM UTC	29
3	@JungkookJapan_	23.11.26 #JUNGKOOK NAVER記事 ジョングク「Standing Nex...	Nov 26, 2023 · 3:42 AM UTC	533
4	@1BforJK	"GOLDEN" จากจองกุกยังคงอยู่ใน Top 5 บนชาร์ตราย...	Nov 26, 2023 · 8:38 AM UTC	113
...	...	...	...	...
4995	@jungkookturf	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	Nov 25, 2023 · 9:38 AM UTC	0
4996	@melkb_55	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	Nov 25, 2023 · 9:38 AM UTC	0
4997	@jjkjeonsz	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	Nov 25, 2023 · 9:38 AM UTC	0
4998	@yuko_stss	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	Nov 25, 2023 · 9:38 AM UTC	0
4999	@hongs130613	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	Nov 25, 2023 · 9:38 AM UTC	0

Figure 9: Random tweets from the Batch data

2.3.2. Data pre-processing:

a. Data translation:

As shown in Fig 9, there exist tweets in various languages, namely Korean, English, Thai, Finnish… which made our first step in the data pre-processing to be translation.

In this step we have used the library “langdetect” to create the function “detect_languages ()”. It helps detect the language of the tweets. This function takes as parameters the file name and the column name where the test is and returns a new file that contain an additional column called “Detected_language”. The need for this extra step is due to the fact that “googletrans” translator cannot detect certain languages. Thus, we have added this extra step to expedite the process and facilitate the translation process for it. We have used ChatGPT’s help for this part of code.

```
[ ]  import pandas as pd
     from langdetect import detect

     def detect_language(text):
         try:
             return detect(text)
         except:
             return 'unknown'

     def detect_languages(input, text_column):
         # Read the CSV file
         df = pd.read_csv(input)

         # Create a new column for detected languages
         df['Detected_Language'] = df[text_column].apply(detect_language)

         # Save the DataFrame with detected languages to a new CSV file
         output_csv = input.replace('.csv', '_with_languages.csv')
         df.to_csv(output, index=False)

     detect_languages('/content/tweet3.csv', 'text')
```

Figure 10: Language detection python function

As a result, we got a new file “tweet3_with_languages.csv” that we will be using in the function” translate_csv ()”. It takes for parameters the file name or path, name of the file where to put the translated text, the column name of where the text is, and the column name of the language. It applied the “Translator ()” function from “googletrans” library.

```
import pandas as pd
from googletrans import Translator
import time

def translate_text(text, src_language, dest_language='en'):
    translator = Translator()
    translation = translator.translate(text, src=src_language, dest=dest_language)
    return translation.text

def translate_csv(input_csv, output_csv, text_column, src_language_column):

    df = pd.read_csv(input_csv)
    df['Translated_Text'] = df.apply(lambda row: translate_text(row[text_column], row[src_language_column]), axis=1)

    # Save the translated DataFrame to a new CSV file
    df.to_csv(output_csv, index=False)

# Translate file
translate_csv('/content/tweet3_with_languages.csv', '/content/transtweets3.csv', 'text', 'Detected_Language')
```

Figure 11: Translate python function.

Fig 12 shows the results of our translation.

	username	date	Likes	Detected_Language	Translated_Text
0	@urmystar_jk	Nov 26, 2023 · 9:26 AM UTC	2231	ko	[231120] Perfect Golden Showcase!✨ #JungKook_G...
1	@Daily_JKUpdate	Nov 25, 2023 · 11:34 PM UTC	2659	en	📊 Jungkook's "GOLDEN" is BACK to #1 on iTunes ...
2	@1BforJK	Nov 26, 2023 · 10:00 AM UTC	29	en	เพลงจองกุกบนชาร์ต Spotify global chart (25/11/...
3	@JungkookJapan_	Nov 26, 2023 · 3:42 AM UTC	533	ja	23.11.26 #JUNGKOOK NAVER Article Jungkook "Sta...
4	@1BforJK	Nov 26, 2023 · 8:38 AM UTC	113	th	"GOLDEN" by Jungkook remains in the Top 5 on S...
...	...	...	...	...	...
4995	@jungkookturf	Nov 25, 2023 · 9:38 AM UTC	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...
4996	@melkb_55	Nov 25, 2023 · 9:38 AM UTC	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...
4997	@jjkjeonsz	Nov 25, 2023 · 9:38 AM UTC	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...
4998	@yuko_stss	Nov 25, 2023 · 9:38 AM UTC	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...
4999	@hongs130613	Nov 25, 2023 · 9:38 AM UTC	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...

5000 rows × 5 columns

Figure 12: Translated tweets from batch data

b. Data cleaning:

An important step in data pre-processing is data cleaning. As shown in Fig 9, raw tweets often come with tags, links, and gibberish which Gen-Z often use as a way of expressing their bafflement or excitement.

To make the cleaning easier, we started by turning all the text to lower case. Then, we removed all unnecessary characters such as hashtags, mentions, links, numbers, words with numbers, parentheses, punctuation, … As a result, to some data visualization challenges, we found it useful to transform emojis to their textual format using the library emoji and change the dates format.

```
#cleaning
import re
def clean_tweet(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)# delete text between []
    text=re.sub(r'#\w+', '', str(text))#delete hashtags
    text = re.sub(r'@\w+', '', text)#delete mentions
    text = re.sub(r'http\S+', '', text)#delete links
    text = re.sub('<.*?>+', '', text)# delete text between <>
    text = re.sub(r'\([^)]*\)', '', text)# delete text between ()
    text = re.sub(r'[.,;]', '', text)# delete any ; . ,
    text= re.sub(r'\d+', '', text)#delete numbers
    text = re.sub('\n', '', text)#delete new line
    text = re.sub('\w*\d\w*', '', text)#delete words with numbers
    text= text.replace('—', '')
    text= text.replace('-', '')
    text= text.replace("'s", "")
    text= text.replace("’s", "")
    text= text.replace("''", "")
    text= text.replace('"', '')

    return text
```

```
for i in range(len(data1)):
    cleared_txt=[]
    txt = data1.loc[i]["Translated_Text"]
    data1.at[i,"Clean_Text"]=clean_tweet(txt)
```

Figure 13: Data cleaning python function

```
[ ] import emoji

    def extract_emojis(text):
        return [char for char in text if char in emoji.UNICODE_EMOJI]

    data1['Clean_Text'] = data1['Clean_Text'].apply(lambda x: emoji.demojize(str(x)))
```

```
[ ] #fix date format
    data1['date'] = pd.to_datetime(data1['date'].str.replace(' · ', ' '), format='%b %d, %Y %I:%M %p %Z')
```

Figure 14: Emoji to Text python function

As a result, we obtain the data frame below:

	username	date	Likes	Detected_Language	Translated_Text	Clean_Text
0	@urmystar_jk	2023-11-26 09:26:00+00:00	2231	ko	[231120] Perfect Golden Showcase!✨ #JungKook_G...	perfect golden showcase!:sparkles:
1	@Daily_JKUpdate	2023-11-25 23:34:00+00:00	2659	en	📊 Jungkook's "GOLDEN" is BACK to #1 on iTunes ...	:bar_chart: jungkook "golden" is back to on i...
2	@1BforJK	2023-11-26 10:00:00+00:00	29	en	เพลงจองกุกบนชาร์ต Spotify global chart (25/11/...	เพลงจองกุกบนชาร์ต spotify global chart sta...
3	@JungkookJapan_	2023-11-26 03:42:00+00:00	533	ja	23.11.26 #JUNGKOOK NAVER Article Jungkook "Sta...	naver article jungkook "standing next to you...
4	@1BforJK	2023-11-26 08:38:00+00:00	113	th	"GOLDEN" by Jungkook remains in the Top 5 on S...	"golden" by jungkook remains in the top on sp...
...	...	...	...	...	...	...
4995	@jungkookturf	2023-11-25 09:38:00+00:00	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	stream golden on spotify
4996	@melkb_55	2023-11-25 09:38:00+00:00	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	stream golden on spotify
4997	@jjkjeonsz	2023-11-25 09:38:00+00:00	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	stream golden on spotify
4998	@yuko_stss	2023-11-25 09:38:00+00:00	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	stream golden on spotify
4999	@hongs130613	2023-11-25 09:38:00+00:00	0	en	STREAM GOLDEN ON SPOTIFY #JungKook_GOLDEN #Sta...	stream golden on spotify

Figure 15: Ready-to-use Data

2.4. Sentiment Analysis:

Sentiment Analysis is a process that entails determining the mood and emotions of a certain audience. The sentiment gathered provides a strong tool in analysing and predicting trends in various fields depending on the goal of the research. To analyse the sentiment of the tweets in hand, we have used TextBlob.

This library uses the function "polarity ()" to give a polarity value to each tweet. For textual data, the sentiment is based on the intensity of the lexicon used and its focus on semantics. Textblob turns back its pre-defined lexical resources which have been classified into positive and negative. Each tweet is presented to Textblob as a bag of words and each word is graded on its sentiment score. Finally for each tweet, we gather the score of the individual words used in it and average the final sentiment. If polarity= 0, then the tweet is neutral. When the score is anywhere in]0,1], it is positive, and when it is in [-1,0[, it becomes negative.

```
[ ]  #DETECT POLARITY OF TWEETS
     def detect_polarity(text):
       return TextBlob(str(text)).polarity
```

```
[ ]  data1['Polarity'] = data1['Clean_Text'].apply(detect_polarity)
```

Figure 16: Sentiment Analysis

The polarity is then stored in a new column in the data frame.

	Unnamed: 0	username	date	Likes	Clean_Text	Polarity
0	0	@urmystar_jk	2023-11-26 09:26:00+00:00	2231	perfect golden showcase!:sparkles:	0.650
1	1	@Daily_JKUpdate	2023-11-25 23:34:00+00:00	2659	:bar_chart: jungkook “golden” is back to on i...	0.100
2	2	@1BforJK	2023-11-26 10:00:00+00:00	29	เพลงจองกุกบนชาร์ต spotify global chart sta...	0.000
3	3	@JungkookJapan_	2023-11-26 03:42:00+00:00	533	naver article jungkook “standing next to you...	0.000
4	4	@1BforJK	2023-11-26 08:38:00+00:00	113	“golden” by jungkook remains in the top on sp...	0.325
...	...	...	...	...	...	...
4995	4995	@jungkookturf	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300
4996	4996	@melkb_55	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300
4997	4997	@jjkjeonsz	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300
4998	4998	@yuko_stss	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300
4999	4999	@hongs130613	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300

Figure 17: Example of Polarity Scores

We defined a function called tweet_sentiment () that takes as parameter the text of the tweet and returns positive, negative, or neutral.

```
[ ]  def tweet_sentiment(tweet):
       tweet_analysis = TextBlob(clean_tweet(tweet))
       if tweet_analysis.polarity > 0:
         return 'positive'
       elif tweet_analysis.polarity == 0:
         return 'neutral'
       else:
         return 'negative'
```

```
[ ]  data1['Sentiment'] = data1['Clean_Text'].apply(tweet_sentiment)
```

Figure 18: Sentiment textual python function

The results are as follows:

	username	date	Likes	Clean_Text	Polarity	Sentiment
0	@urmystar_jk	2023-11-26 09:26:00+00:00	2231	perfect golden showcase!:sparkles: guki	0.650	positive
1	@Daily_JKUpdate	2023-11-25 23:34:00+00:00	2659	:bar_chart: jungkook “golden” is back to on i...	0.100	positive
2	@1BforJK	2023-11-26 10:00:00+00:00	29	เพลงจองกุกบนชาร์ต spotify global chart sta...	0.000	neutral
3	@JungkookJapan_	2023-11-26 03:42:00+00:00	533	naver article jungkook “standing next to you...	0.000	neutral
4	@1BforJK	2023-11-26 08:38:00+00:00	113	“golden” by jungkook remains in the top on sp...	0.325	positive
...	...	...	...	...	...	...
4995	@jungkookturf	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300	positive
4996	@melkb_55	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300	positive
4997	@jjkjeonsz	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300	positive
4998	@yuko_stss	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300	positive
4999	@hongs130613	2023-11-25 09:38:00+00:00	0	stream golden on spotify	0.300	positive

Figure 19: Data frame with sentiment analysis results

2.5. Real time monitoring:

As much freedom Nitter’s API has given us, the processing power of cloud-based platforms (especially free plans) has slowed down the process of monitoring real-time data.

Using Streamlit, we have built a dashboard that visualizes the sentiment analysis of a data frame into various graphs. The web application file is called “streamlit_app.py”. In a separate python file “fetch_data.py”, we have recycled the function used earlier for the initial data collection process. The new version of “get_tweets ()” function, processes every single tweet as soon as it scraped. It is cleaned, translated, and put through sentiment analysis. The problem with Nitter scraper is that once it has started, it cannot hand over the tweets dynamically. We had to wait for it to scrape the number of data we asked for then it is stored in our data frame. We have tried linking it to an SQL database, but the problem remained the same.

fetch_data.py > get_tweets

```
49
50
51  def get_tweets(name, modes,no, lang):
52      final_tweets = []
53      # Scraping the data
54      tweets = scraper.get_tweets(name, mode=modes,number=no,language=lang)
55      for tweet in tweets["tweets"]:
56          # Cleaning the data
57          ctext = clean_tweet(tweet["text"])
58          # Sentiment analysis
59          sentiment = tweet_sentiment(ctext)
60          polarity = TextBlob(ctext).polarity
61          # Gathering information from the tweets
62          data = [
63                      tweet["user"]["username"],
64                      ctext,
65                      tweet["date"],
66                      tweet["stats"]["likes"],
67                      tweet["stats"]["retweets"],
68                      sentiment,
69                      polarity
70                      ]
71          final_tweets.append(data)
72      # Final dataframe outside the loop
73      data = pd.DataFrame(final_tweets, columns=["username", "text", "date", "Likes", 'Retweets', "Sentiment", "Polarity"])
74      return data
75
```

Figure 20: Updated get_tweets ()

We proceeded in treating the two python files as separate threads that had to run in parallel. The web application would run and visualize the initial batch data it had and then pause. This downtime is enough for the fetch_data thread to go through the data collection and sentiment analysis process for a limited number of tweets and would update the data frame fed to the dashboard. Once this thread is over, the web application refreshes, showing the visualization of the near-real time data updated.

thread.py > ...

```python
import threading
import subprocess
import time

def run_script(script_name):
    subprocess.Popen(['python', script_name])

def thread_one():
    while True:
        print("Thread 1 running...")
        run_script('streamlit_app.py')
        time.sleep(120)
        print("Thread 1 restart...")

def thread_two():
    for _ in range(3):  # Loop thread 2 three times
        print("Thread 2 running...")
        run_script('fetch_data.py')
        time.sleep(120)  # Simulating some work
        print("Thread 2 restart...")

# Create threads
thread1 = threading.Thread(target=thread_one)
thread2 = threading.Thread(target=thread_two)

# Start threads
thread1.start()
```

Figure 21: Threading

3. Results:

3.1. Sentiment analysis of batch data:

As a result of analysing 5000 tweets of Jungkook's fans about his album GOLDEN, we can see that overall sentiments are positive: with more than 3500 tweets that have positive polarity which is more than the half of the number of tweets as the histogram of polarity below shows in Fig 22.

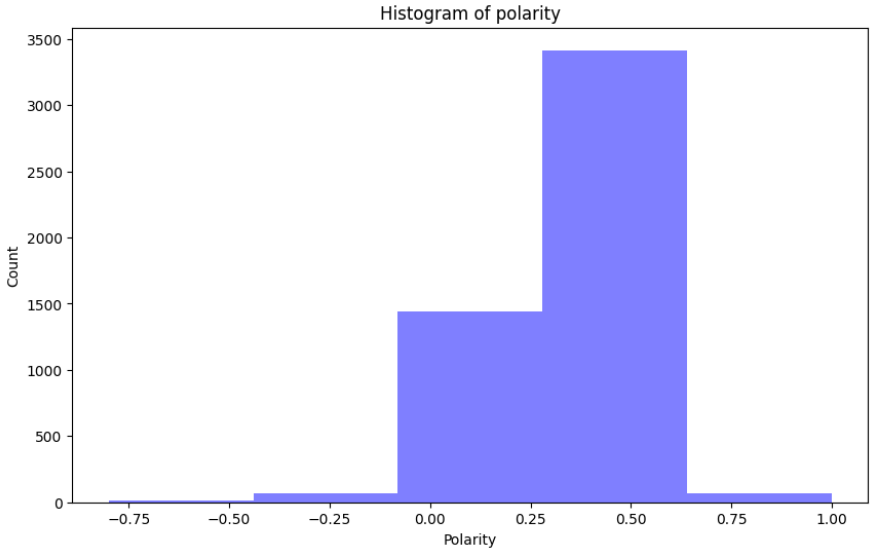

Figure 22: Polarity Histogram

The following table and pie chart confirm the results in the histogram by showing that 3946 tweets are positive, which is equivalent to 78.7% of the total number of tweets, 18.7% neutral and 2.3% negative.

	Sentiment	Clean_Text
2	positive	3946
1	neutral	940
0	negative	114

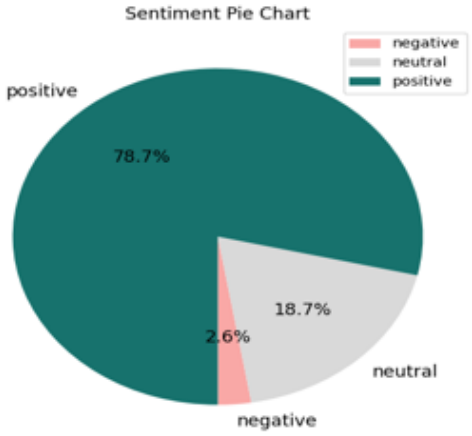

Figure 23: Sentiment Distribution

For more visualization and better understanding, we made an interactive line graph (see file called sentiment_plot.html). This graph has been a great indicator in drawing the relationship between then current events and the change of trends in sentiments.

On October 3rd, we notice a spike in positive tweets as a reaction to the announcement of the new album and the beginning of the promotion period for it. Throughout that timeframe, sentiments spiked and lowered down depending on the event. Most of the surges in positive tweets are associated with the following dates:

- 03/11/2023: The release of the album and the official music video for the main song.
- 20/11/2023: Jungkook performed his first solo showcase with 2800 fan in place and more than 1.2 million online. This date was exceptional as, sentiments have reached a polarity score of 1 for the first time as fans were very vocal and expressive about their love for the event.

However, after that event, there have been rumours about the artist having to enlist in the Korean government's mandatory military service of 2 years. This means that the live performance would be his last until June 2025. This particularly disappointing news have reflected on the graph showing a surge in the negative polarity scores, especially on November 22nd, where the announcement was made official. Nevertheless, fans have also been showing their artist their support by positive messages and encouraging other to stream his album to get him music awards as Fig 24 shows, the top common words are "stream" and "Spotify".

Common_words	count
stream	2769
spotify	1215
live	251
domination	211
rise	210
stage	198
recomeback	181
album	147
:sparkles:	130
love	120
thank	98

Figure 24: Common words

Using word cloud, we could further analyse the content of the tweets for each type of sentiments. In tweets with positive sentiments, we realised that fans were cheering for their artist by focusing on steaming his songs on Spotify as mentioned before, using emojis like purple heart which is the love symbol between ARMYs (the name of the fanbase) and BTS (the band).

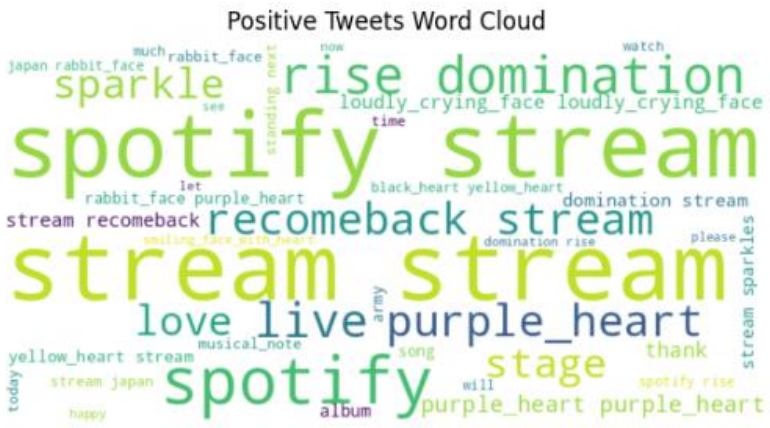

Figure 25: Positive word cloud

For the most used words in the negative tweets, we can see the word sad bigger than others which reflects the fans' feelings about not being able to see their favourite artist performing for the next two years. The emojis "rabbit_face" (Jungkook has a bunny smile)," purple_heart" and loudly crying face emoji.

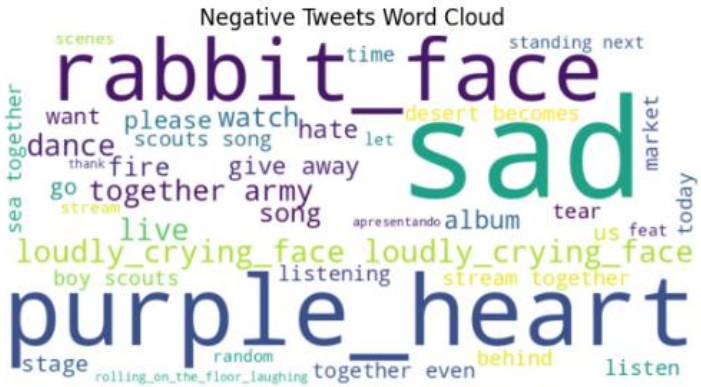

Figure 26: Negative word cloud

For the neutral tweets, "STANDING NEXT TO YOU" was most repeated because it's the title of the main song in the album along with the word "stream". This means that maybe the neutral tweets were speaking about how many streams and views the main song is getting with using emojis like rabbit face emoji which refers to Jungkook and purple heart emoji that refers to ARMY.

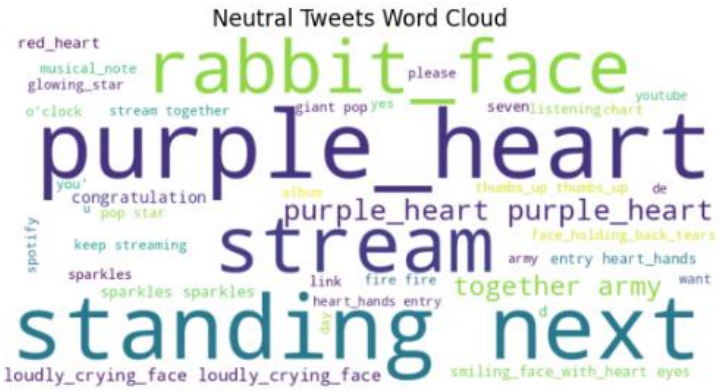

Figure 27: Neutral word cloud

3.2. Real-time data monitoring:

The figures below show the sentiment analysis visualization of the real time data collected on the day of the demonstration, November 29th. Our dashboard consists of 5 different graphs each visualizing polarity over time, over count, sentiments by number of likes, sentiments by number of retweets and the dynamic plot of sentiments.

As the data frame was still empty – only containing 200 tweets – the polarity scores tip over to neutral sentiment. Nevertheless, we notice that positive sentiment tweets show a higher number of like and retweet count than their negative counterpart, Fig 29.

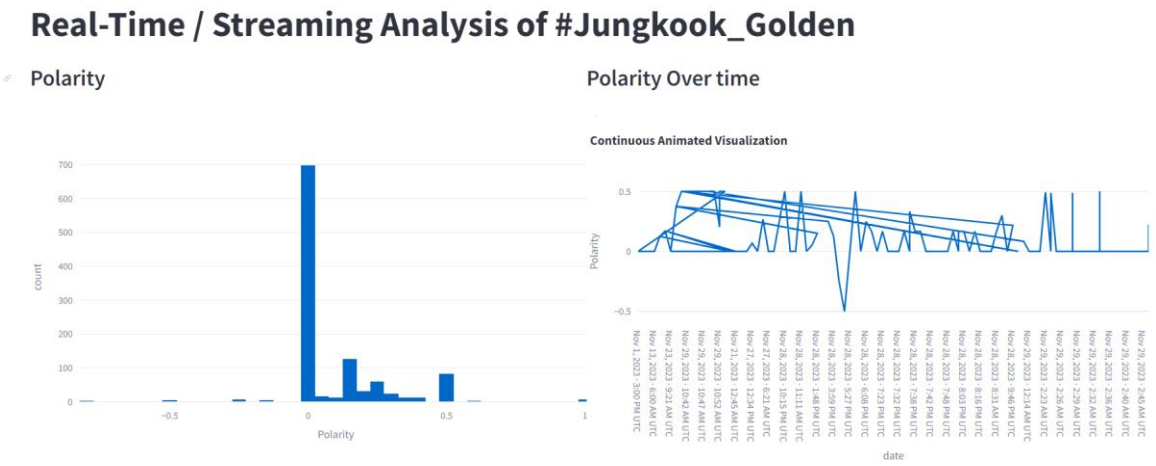

Figure 28: Dashboard 1

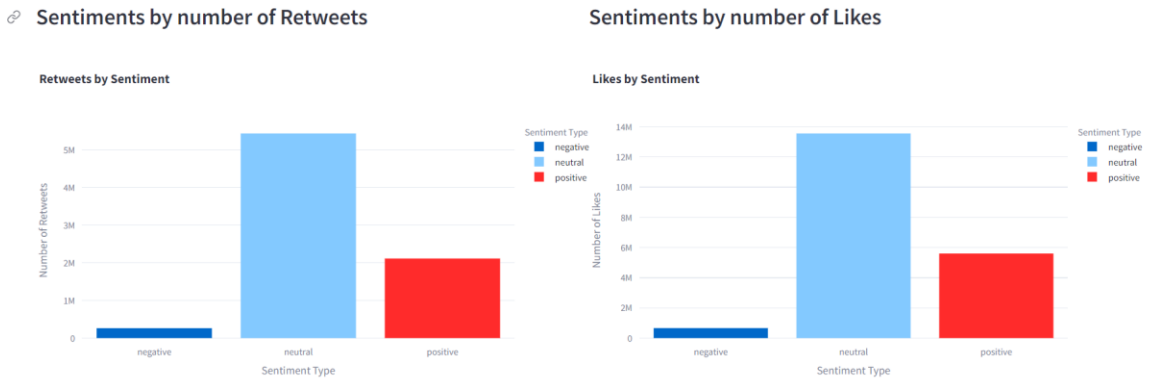

Figure 29: Dashboard 2

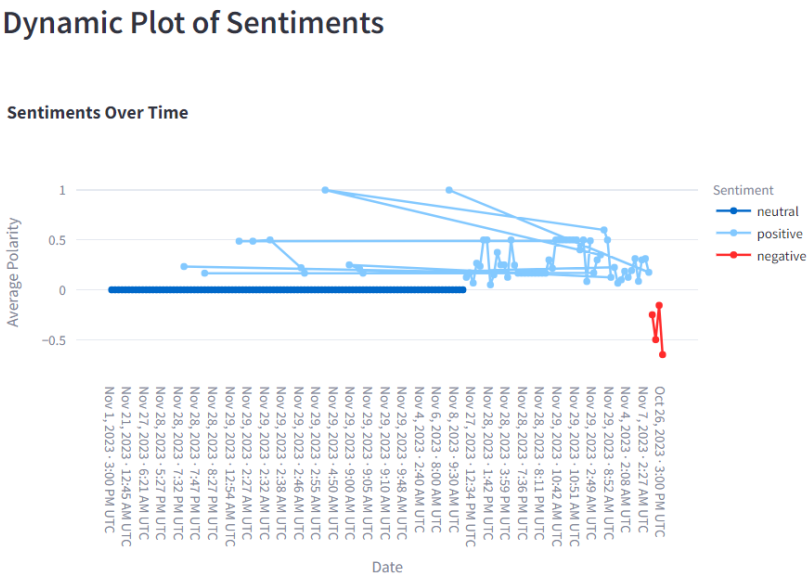

Figure 30: Dashboard 3

Detailed View of Scraped Tweets

	username	text	date	Likes	Retweets	Sentiment	Polarity
11	@esteekoo	have a safe flight we love you ✈️❤️❤️❤️	Nov 29, 2023 · 10:43 AM UTC	0	0	positive	0.5
12	@kifit379	have safe flight	Nov 29, 2023 · 10:43 AM UTC	0	0	positive	0.5
13	@Fernand59802220	have a safe flight we love you ✈️	Nov 29, 2023 · 10:43 AM UTC	0	0	positive	0.5
14	@purple_is_fun	have a safe flight we love you ✈️❤️❤️❤️	Nov 29, 2023 · 10:46 AM UTC	0	0	positive	0.5
15	@SevenJK071423	have a safe flight we love you ✈️❤️❤️❤️	Nov 29, 2023 · 10:47 AM UTC	0	0	positive	0.5
16	@BamieGuks	have a safe flight we love you ✈️❤️❤️❤️	Nov 29, 2023 · 10:49 AM UTC	0	0	positive	0.5
17	@pn1997jk	have safe flight	Nov 29, 2023 · 10:51 AM UTC	0	0	positive	0.5
18	@jkjuly_svn7	have safe flight	Nov 29, 2023 · 10:51 AM UTC	0	0	positive	0.5
19	@AoJjk_3Y3s4_JjK	✨have a safe flight ᴋʀ✈️ᴜs✨ has a heart of gold ✧ he's been working really hard	Nov 29, 2023 · 10:52 AM UTC	0	0	positive	0.2028
20	@JjkLaine	have a safe flight we love you	Nov 29, 2023 · 10:52 AM UTC	0	0	positive	0.5

Figure 31: Dashboard 4

4. Conclusion:

In conclusion, we can say that the overall sentiments towards this album were positive with 79% from the 5000 tweets collected in the period starting from 3rd of October to 26th of November 2023. The change in sentiments towards GOLDEN was affected by every event related to the promotions of the album and rumours or news about the private life of the artist. In this project, we have learned how to manipulate data from collecting it to interpreting the trends. We have also learned that you must look at the data from multiple different aspects to get the full picture behind it and accurately retell its story.

References

Amrrs, 2020. Python_Matplotlib_Animated_Bar_Chart_Race.ipynb. Available at: https://github.com/amrrs/youtube-r-snippets/blob/master/Python_Matplotlib_Animated_Bar_Chart_Race.ipynb

Dr Sandra, O.M. (2023) "Sentiment Analysis (FIFA dataset)" [Activity 3 week 6], 7012DATSCI: Big Data Computing, Faculty of Engineering and Technology LJMU

Lorenzo. B, and Patricio. S, 2022. ntscrapper. Available at: https://github.com/bocchilorenzo/ntscraper?search=1

Mr_KnowNothing, 2019. "Twitter sentiment Extaction-Analysis,EDA and Model", Available at: https://www.kaggle.com/code/tanulsingh077/twitter-sentiment-extaction-analysis-eda-and-model#Most-Common-words-in-Text

Ntscraper, Available at: https://pypi.org/project/ntscraper/

"10 Mind-Blowing BTS Facts and Statistics." *Brandwatch*, https://www.brandwatch.com/blog/bts facts-and-statistics/. Accessed 8 Dec. 2023.

Barnes, Jenae. "Twitter Ends Its Free API: Here's Who Will Be Affected." *Forbes*, https://www.forbes.com/sites/jenaebarnes/2023/02/03/twitter-ends-its-free-api-heres-who-will-be-affected/. Accessed 7 Dec. 2023.

Google Colaboratory. https://colab.research.google.com/notebooks/basic_features_overview.ipynb. Accessed 7 Dec. 2023.

Muhammad, Umar Sani. "A Comparison of NLTK and TextBlob for Text Analysis." *Medium*, 25 June 2023, https://medium.com/@umarsmuhammed/a-comparison-of-nltk-and-textblob-for-text-analysis-bd9ebcd0ecd9.

Python Tutorial: Streamlit. https://www.datacamp.com/tutorial/streamlit. Accessed 8 Dec. 2023.