DATA ENGINEERING INDIVIDUAL COURSEWORK

SPOTIFY PLAYLIST DATABASE:

A SENTIMENT ANALYSIS

April 2022
Word Count: 3,998

Repository Link:
https://github.com/doratian99/MSIN0166-DE-Individual-Project

TABLE OF CONTENTS

1.0 INTRODUCTION

Since many decades ago, music has been recognised as an opportunity to address mental health challenges (Schriewer and Bulaj, 2016). Nowadays, audience can enjoy music more conveniently via music streaming services instead of downloading the audio file of a song. As one of the biggest music streaming platforms, Spotify had over 365 million users by February 2022 (Caddy, 2022).

One important feature of Spotify is the editorial playlists. Spotify's in-house teams curate these playlists by selecting collections of songs with some similarities. Many playlists have emotion-related tags such as "happy", "moody" and "chilling", and each of them is designed to match the audience's emotions.

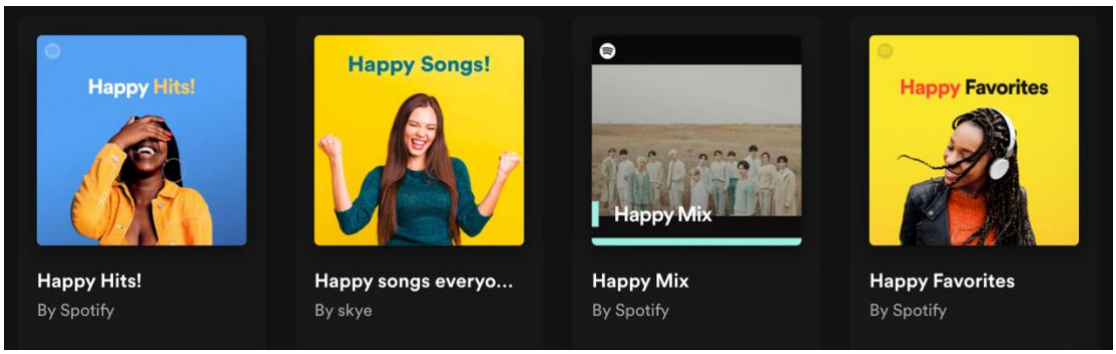


Figure 1: Spotify Curated Playlists Related to "Happy"

This project aims to create a database of a specific Spotify playlist that contains many different attributes of the songs within that playlist and conduct further sentiment analyses of these songs based on the attributes.

In achieving the objective of the project, the following steps are executed and will be explained in detail throughout this report. First, real-time streaming data will be extracted through API scraping from Spotify and several other relevant platforms. The data will then be processed and stored in a suitable way so that further analyses are allowed. This project will then conduct sentiment analyses on these processed data so that we can evaluate whether the “emotion” of the songs in the playlist match with the playlist title. It is believed that successful execution of the project with provide precious guidelines for music streaming services’ sentiment related research on a bigger database beyond the project scope.

Figure 2: Project Workflow

2.0
DATA COLLECTION

This project mainly used the Application Programming Interface (API) technique in Python language to extract data from four different platforms including Spotify, MusixMatch, YouTube and Twitter. The API is an important tool to obtain data from dynamic websites and allows some level of customisations (Medium,2022).

2.1 SPOTIFY DATA

Data from Spotify was scraped first. Spotify provides API services on its developer website and a number of functions can be achieved under the free user plan. On its developer website, a Python library called Spotipy (https://github.com/plamere/spotipy) is recommended for executing the API calls.

First of all, the project wanted to retrieve a list of featured playlists (Editor's Picks) on Spotify to decide which playlist to extract and evaluate. By using the *featured_playlists()* method, a list of playlists which were the Editor's picks on the day when the data was scraped was returned:

```
Editor's picks
0 New Music Friday
1 Feel Good Friday
2 RapCaviar
3 Main Stage
4 I Love My '90s Hip-Hop
5 Mood Booster
6 Dance Hits
7 Today's Top Hits
8 just hits
9 Dance Party
10 Happy 80s
11 young & free
```

Figure 3: Featured Playlists on 8th April

The sixth result was a playlist named ***"Mood Booster"***, which was highly related to the research objective - to analyse the emotion of songs with sentimental analysis techniques and evaluate whether the emotion of the song match with the overall playlist genre.

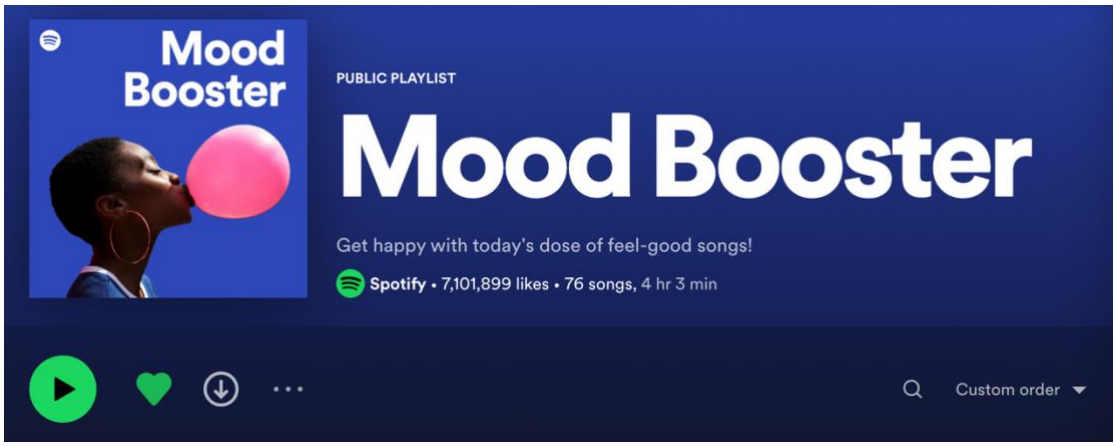


Figure 4: The *Mood Booster* Playlist

According to Spotify's description of this playlist, the songs in the playlist are supposed to make the audience "feel good" and "get happy". This project is curious about if the emotions of the songs in this playlist actually match the purposes. To dig deeper into the tracks in this playlist, the playlist's ID on Spotify needs to be known. Unfortunately, the only feasible way to get a playlist's ID is through getting a user's current playlist. Therefore, this playlist was manually followed on Spotify and added to the profile using a personal Spotify account.

With the playlist id being known, the project was able to retrieve more information about the tracks in this the playlist. A list of 76 song names were first retrieved via the *playlist_items()* method, and the 76 songs' corresponding Spotify IDs were retrieved via similar ways. The songs' Spotify IDs allowed the project to retrieve a number of attributes of the track by using the Spotify API's *track()* method, including the artist information. By calling the method and looping into the nested dictionary (see outcome example in Figure 5) returned by the method, this project was able to get the *artist's name, album, Spotify popularity, release date, duration* information of the track. The method could also tell whether a track contains explicit content or not: most music streaming platforms distinguish and differentiate between tracks that is suitable for mainstream consumption, and those songs that may contain a parental advisory or may be considered explicit content (Soundplate, 2022). On Spotify, a track with explicit content will have an "E" or "Explicit" symbol next to its title. With the API, boolean values of True or False was returned regarding the "explicit" attribute.


```
{'album': {'album_type': 'single',
           'artists': [{'external_urls': {'spotify': 'https://open.spotify.com/artist/2ZmXexIJAD7PgABrj0qQRb'},
                        'href': 'https://api.spotify.com/v1/artists/2ZmXexIJAD7PgABrj0qQRb',
                        'id': '2ZmXexIJAD7PgABrj0qQRb',
                        'name': 'N.Flying',
                        'type': 'artist',
                        'uri': 'spotify:artist:2ZmXexIJAD7PgABrj0qQRb'}],
           'available_markets': ['AD',
                                 'AE',
                                 'AG',
                                 'AL',

'disc_number': 1,
'duration_ms': 210652,
'explicit': False,
'external_ids': {'isrc': 'KRA381900017'},
'external_urls': {'spotify': 'https://open.spotify.com/track/2LwH6T39A5IODRgPv9XitR'},
'href': 'https://api.spotify.com/v1/tracks/2LwH6T39A5IODRgPv9XitR',
'id': '2LwH6T39A5IODRgPv9XitR',
'is_local': False,
'name': 'Rooftop',
'popularity': 61,
'preview_url': 'https://p.scdn.co/mp3-preview/a22310aa8b97d93e7e850c35a6e04f1165b11419?cid=7b1fa7a7eb25461f8d3a4a66e1966de5',
'track_number': 1,
'type': 'track',
'uri': 'spotify:track:2LwH6T39A5IODRgPv9XitR'}
```


Figure 5: Output Example of the *track()* Method

2.2 MUSIXMATCH DATA

MusixMatch is an Italian music data company which has the world's largest database of 14 million lyrics items in various different languages (Baydeer, 2021). With the lyrics data provided by MusixMatch, this project would be able to conduct sentiment analysis on the lyric strings.

With the free API plan provided by MusixMatch, only 30% of the lyrics of a song was accessible. The project would have to assume most songs would have their emotions set in stone in the very first bit.

By inputting the title and the artist's name of the song as the parameters of the MusixMatch request call, JSON styled results were pulled. In Figure 6, this project used *Halsey*'s song "*Drive*" as an example to examine the output.

In [31]:
```
# Uses a random song - "Drive" by Halsey as the input
req = requests.get(url,params = {
    "apikey": musixmatch_key,
    "q_track": "Drive",
    "q_artist": "Halsey"
})


# Outputs in JSON
Drive = req.json()

Drive
```

Out[31]:
```
{'message': {'header': {'status_code': 200, 'execute_time': 0.091995000839233},
  'body': {'lyrics': {'lyrics_id': 27157087,
    'explicit': 0,
    'lyrics_body': 'My hands wrapped around a stick shift\nSwerving on the 405, I can never keep my eyes off this\n\nMy neck, the feeling
of your soft lips\nIlluminated in the light, bouncing off the exit signs I missed\n\nAll we do is drive\nAll we do is think about the fee
lings that we hide\nAll we do is sit in silence waiting for a sign\nSick and full of pride\nAll we do is drive\n...\n\n******* This Lyric
s is NOT for Commercial use *******\n(1409622496242)',
    'script_tracking_url': 'https://tracking.musixmatch.com/t1.0/m_js/e_1/sn_0/l_27157087/su_0/rs_0/tr_3vUCAHd961rAmbAw7ri0-GrmUAIX5NV9hG
hWN8EfDpZJjEUKAGI1qv2IK_txinFDOwICCmguLzb6ubcAlNPT8vooFsy8SVHhdP1XzhRFTKQjU6Kom3PrrAcarzrp20Og2wa_uZbXCnsCfxWzK1r37BdzYf9bNyvdtAg4TWdAEhH
VqCtsoQQqgelzNgedC1BRumKD7USjuwblOEfmAs7y33DidTcj4mq9HbB-wc63O6N2VZ-FpPMvTMqOyFVFWzKBUfPlAbjUHodSkYjQ83eMERlGFDm2QKIAul_eNdcV7S_1l9kqkWM3
7bAOHWxwYF2ydULNQ5tQrLVWFPvNlJ45D-IPH11AbaYZqHUrQPrzu6uEFMqRWm5umrn12FPqi_yWQOs6kdYPvPj7zwlOOvHbex4k8K_C522w4GZIAtpAByRZOJLonUJj1yHwRQ/',
    'pixel_tracking_url': 'https://tracking.musixmatch.com/t1.0/m_img/e_1/sn_0/l_27157087/su_0/rs_0/tr_3vUCALw5i4_-D8dTbgLa7cuSMifMTbKe6O
xPQcJ6YK_1KrAXRIQt7YROFlIwdFzuAe9V4xL0fOqApGAloQtX6NXo9tQ2kc2tWlWI2HCGzDIU3F-YLlmgJBgjF1q9bzVViKIvtVzkhrJGDYj-eF0XKg-15AUszL8wZDXb1w27eJx
tbdcMtj2Vdl3BdcMAemjB3d9AH9ajFxSlgNYeyJgcPUrHXC0Bumm6Ldn34a5CX1zejC9ngXqqrEjBKaBfhPqJtB8n-9V4idAQN0HQikALWau4gREfuvoFvdln9R6lKTk1fqv8vG4Q
jFpp4Q95KYUKczLJsz2W1cisibSu03zUEz4b59reMmpO6xcL2JXDGGrOPVqT7vQoGoBii_b-cw36JOEh-NULN22N8kkOgY0SgUEtvnfhRAFfCkS0HXmK_EFJRke0tWtzV_Of1w/',
    'lyrics_copyright': 'Lyrics powered by www.musixmatch.com. This Lyrics is NOT for Commercial use and only 30% of the lyrics are retur
ned.',
    'updated_time': '2021-12-22T13:51:39Z'}}}}
```

Figure 6: Output Example a Random Song

2.3 YOUTUBE DATA

Other than emotion-related attributes, this project is also interested in the popularity of the official music videos and user-uploaded lyrics videos of the tracks in the playlist on YouTube. This part of the data would indicate the popularity of the tracks over the internet.

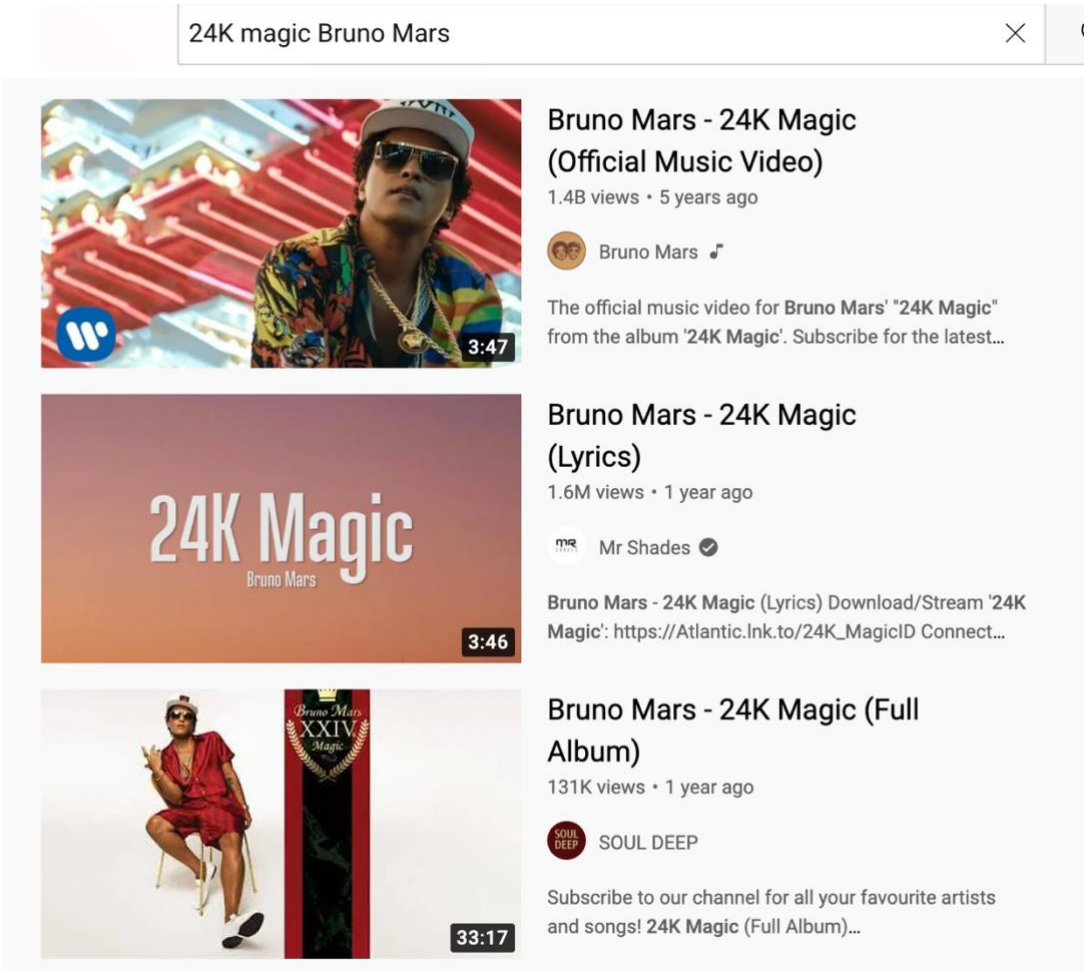


Figure 7: Top3 YouTube Results of Searching "24K Magic" + Bruno Mars

Figure 7 displays the top results of querying the keywords "*24K Magic*" (track title) and "*Bruno Mars*" (artist's name) on YouTube, which includes the official music video of the song uploaded by the artist's own channel as well as user-uploaded lyrics and album videos.

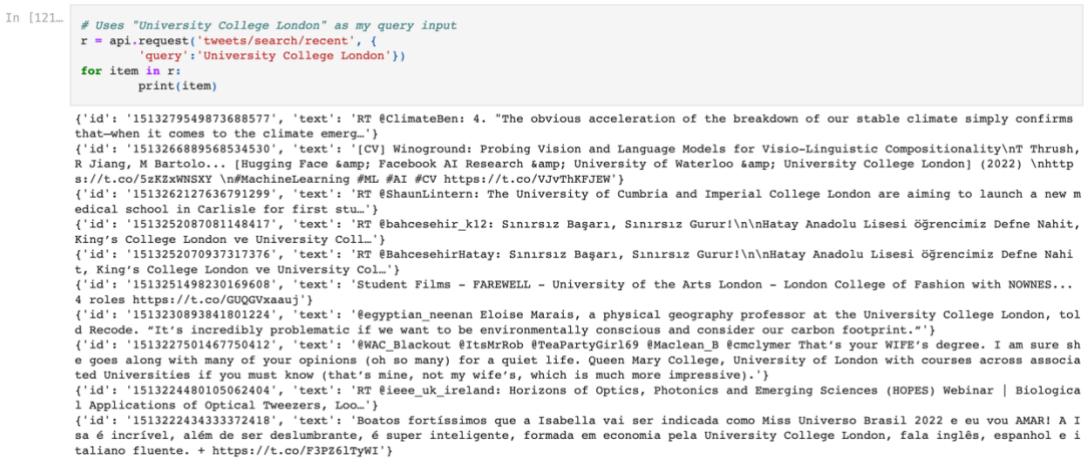
By using the YouTube V3 API's method *Search()*, this project first got the top 10 query results' YouTube video IDs for each of the 76 songs and stored them in a 76*10 dictionary. The number 10 was decided because they must be the most relevant videos to the song. To get the attributes like the view and like count for each video, the YouTube V3 *Statistics()* method was used. This project retrieved a total of four statistics of each of the 760 videos: the view, like, favourite and comment count, and calculates the mean of each count for each of the 76 songs. The four attributes should reflect the popularity of the songs on YouTube.

2.4 TWITTER DATA

Twitter is one of the biggest social networking services in the world where users can post and interact with messages known as "tweets" (Igi-global, 2022). This project is also interested in the emotions associated with the tweets that discuss the tracks in the *Mood Booster* playlist. The tweet strings are expected to tell the user review of the songs, which is an important data source for analysing what do listeners feel about the songs.

To retrieve the tweet strings, the Twitter API was used, and a simple Python wrapper called *Twitter API* (https://github.com/geduldig/TwitterAPI) suggested by the Twitter's developer website was used for making the requests in Python.

The standard result of searching a random query ("University College London" in this case) is displayed in Figure 8. The project is only interested in the "text" section in the requesting result.



```
In [121…
# Uses "University College London" as my query input
r = api.request('tweets/search/recent', {
        'query':'University College London'})
for item in r:
        print(item)
```

```
{'id': '1513279549873688577', 'text': 'RT @ClimateBen: 4. "The obvious acceleration of the breakdown of our stable climate simply confirms
that—when it comes to the climate emerg…'}
{'id': '1513266889568534530', 'text': '[CV] Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality\nT Thrush,
R Jiang, M Bartolo... [Hugging Face &amp;amp; Facebook AI Research &amp;amp; University of Waterloo &amp;amp; University College London] (2022) \nhttp
s://t.co/5zKZxWNSXY \n#MachineLearning #ML #AI #CV https://t.co/VJvThKFJEW'}
{'id': '1513262127636791299', 'text': 'RT @ShaunLintern: The University of Cumbria and Imperial College London are aiming to launch a new m
edical school in Carlisle for first stu…'}
{'id': '1513252087081148417', 'text': 'RT @bahcesehir_k12: Sınırsız Başarı, Sınırsız Gurur!\n\nHatay Anadolu Lisesi öğrencimiz Defne Nahit,
King's College London ve University Coll…'}
{'id': '1513252070937317376', 'text': 'RT @BahcesehirHatay: Sınırsız Başarı, Sınırsız Gurur!\n\nHatay Anadolu Lisesi öğrencimiz Defne Nahi
t, King's College London ve University Col…'}
{'id': '1513251498230169608', 'text': 'Student Films - FAREWELL - University of the Arts London - London College of Fashion with NOWNES...
4 roles https://t.co/GUQGVxaauj'}
{'id': '1513230893841801224', 'text': '@egyptian_neenan Eloise Marais, a physical geography professor at the University College London, tol
d Recode. "It's incredibly problematic if we want to be environmentally conscious and consider our carbon footprint."'}
{'id': '1513227501467750412', 'text': '@WAC_Blackout @ItsMrRob @TeaPartyGirl69 @Maclean_B @cmclymer That's your WIFE's degree. I am sure sh
e goes along with many of your opinions (oh so many) for a quiet life. Queen Mary College, University of London with courses across associa
ted Universities if you must know (that's mine, not my wife's, which is much more impressive).'}
{'id': '1513224480105062404', 'text': 'RT @ieee_uk_ireland: Horizons of Optics, Photonics and Emerging Sciences (HOPES) Webinar | Biologica
l Applications of Optical Tweezers, Loo…'}
{'id': '1513222434333372418', 'text': 'Boatos fortíssimos que a Isabella vai ser indicada como Miss Universo Brasil 2022 e eu vou AMAR! A I
sa é incrível, além de ser deslumbrante, é super inteligente, formada em economia pela University College London, fala inglês, espanhol e i
taliano fluente. + https://t.co/F3PZ6lTyWI'}
```

Figure 8: Output Example of the Twitter API

Similar to the previous requests from MusixMatch and YouTube, this project also used the combination of the "song title" and the "artist's name" as the input parameter. By querying the 76 combinations and only retaining the "text" output, the last part of the database was successfully retrieved.

3.0
DATA PROCESSING

Most of the API result are in JSON formats or in a nested dictionary. By enumerating them in Python, this project was able to generate four dataframes for four different platforms. Each dataframe has 76 rows and has the track name and artist's name columns as the foreign keys.

Figure 9: *Spotify_df*

	track_name	spotify_id	artist_name	album	spotify_popularity	release_date	duration	explicit_content
0	One Right Now (with The Weeknd)	00Blm7zeNqgYLPtW6zg8cj	Post Malone	One Right Now	92	2021-11-05	193506	True
1	dancing in the kitchen	0ohcCrxZkBfFbkuRPOZQZX	LANY	dancing in the kitchen	76	2021-06-25	208599	False
2	Sheesh!	3ddNKnYpVx0ul8vcwbTQ5Y	Surfaces	Sheesh!	75	2021-08-20	148846	False
3	Can I Get It	6w8ZPYdnGajyfPddTWdthN	Adele	30	82	2021-11-19	210384	False
4	Black And White	7rpNuuoMbid56XkDsx2FjE	Niall Horan	Heartbreak Weather	78	2020-03-13	193089	False

The main dataframe, *Spotify_df*, has 8 columns in total and 6 of them are unique Spotify attributes: the Spotify ID, album, Spotify popularity, release date, duration information of a song and whether it contains explicit content.

	track_name	artist_name	lyrics
0	One Right Now (with The Weeknd)	Post Malone	Na-na-na-na, na-na Na-na-na-na, oh no Yeah, ye...
1	dancing in the kitchen	LANY	City lights looking like ice underneath the st...
2	Sheesh!	Surfaces	You know what I'm sayin'? (Sheesh) I be like ...

Figure 10: *Musixmatch_df*

The *Musixmatch_df* dataframe has 3 columns and only the lyrics column is unique which contains the lyric string of each song.

	track_name	artist_name	youtube_views	youtube_likes	youtube_favourites	youtube_comments
0	One Right Now (with The Weeknd)	Post Malone	126618090	2977580	0	104760
1	dancing in the kitchen	LANY	42519900	1020850	0	41860
2	Sheesh!	Surfaces	12180970	252720	0	13190
3	Can I Get It	Adele	52026180	1012500	0	22110
4	Black And White	Niall Horan	126128120	4259400	0	169780

Figure 11: *Youtube_df*

The *Youtube_df* dataframe has 4 unique columns which are the average view, like, favourite and comment counts for the 10 most relevant videos of each of the 76 songs on YouTube.

	track_name	artist_name	tweets
0	One Right Now (with The Weeknd)	Post Malone	I'm obsessed with this bop by The Weeknd and P...
1	dancing in the kitchen	LANY	Hi everyone! One of my favorite songs is danci...

Figure 12: *Twitter_df*

Similar to the *Musixmatch_df* dataframe, the *Twitter_df* only contains one unique column, which is the tweet strings that discuss each of the 76 songs in the playlist.

4.0
DATA STORAGE

4.1 LOCAL STORAGE

The four dataframes are first exported to 4 CSV files for the local storage. However, for more flexible, affordable, and scalable data management, the dataframes need to be stored in a more reliable cloud storage database.

4.2 CLOUD DATABASE

This project chose the PostgreSQL as the database management system. PostgreSQL is a powerful open-source object-relational database system with a solid reputation for active development and stability, functional robustness, and good performances for over 30 years (PostgreSQL, 2022). This system would also allow the project to conduct analyses via SQL queries via the Postgres connection and a relational database that had meaningful linkages could be created.

```
# Initialises the db_engine using my own credentials
db_engine = create_engine('postgresql://doratian18:qwerty123@depgdb.crhso94tou3n.eu-west-2.rds.amazonaws.com:5432/doratian18')
```

Figure 13: Initialising the Database Connection

With the user, host name and port number being initialised, the *db_engine* was created for future connections to this database. The four dataframes were then uploaded in SQL form via the connection.

```
doratian18-> \dt
                List of relations
 Schema |        Name          | Type  |   Owner
--------+----------------------+-------+------------
 public | Company_stock_sql    | table | doratian18
 public | PARA_stock_news_sql  | table | doratian18
 public | PARA_stock_sql       | table | doratian18
 public | musixmatch_df        | table | doratian18
 public | spotify_df           | table | doratian18
 public | twitter_df           | table | doratian18
 public | youtube_df           | table | doratian18
(7 rows)
```

Figure 14: Dataframes Stored in PostgreSQL

By connecting to the database in the terminal and using the command line prompts to check the tables, the last four rows in Figure 14 indicates that the four dataframes were successfully stored in the database and had the correct ownership.

5.0 RELATIONAL DATABASE

5.1 SCHEMA

Now that the dataframes were stored on PostgreSQL, a schema needs to be created to make sure that there are valid connections between each of the dataframe in the relational database.

As explained in the previous chapter, there were four relational tables stored in the database in SQL forms.

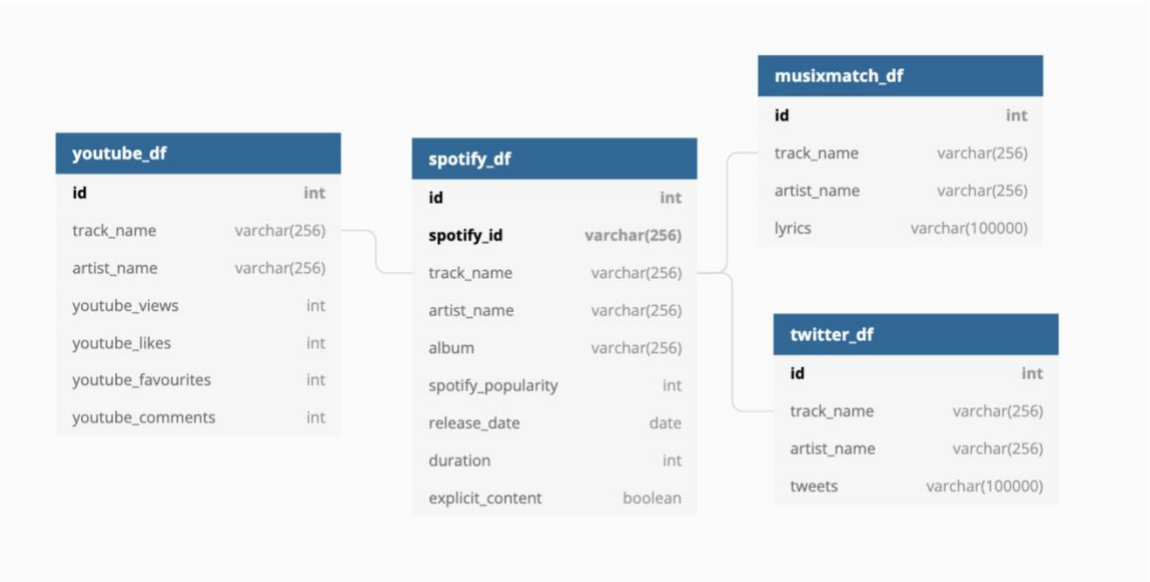


Figure 15: Schema Diagram

As shown in the schema diagram in Figure 15, each of the four tables have ID as the primary key and the track name as the foreign key for referencing attributes in other tables.

The schema was written in a SQL fille and initialised in the Postgres database where the dataframes were stored. Figure 16 shows that the schema was successfully initialised in the database.

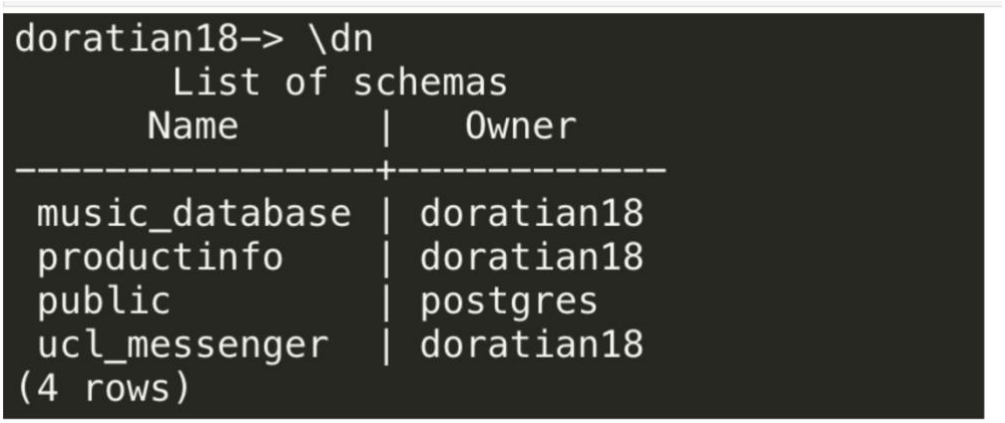


Figure 16: Schema Stored in PostgreSQL

5.2 SQL QUERIES

Some SQL queries were performed based on several mock questions to display the valid linkages and the value of the relational database.

Query 1: Which artist has the most tracks on the playlist?

```
In [220…
pd.read_sql('''
    SELECT spotify_df.artist_name, COUNT(*) as no_of_tracks
    FROM spotify_df
    GROUP BY spotify_df.artist_name
    ORDER BY No_of_tracks DESC LIMIT 15
    ''', db_engine)
```

Out[220…

	artist_name	no_of_tracks
0	Tones And I	2
1	Ed Sheeran	2
2	Gryffin	2
3	Dua Lipa	2
4	Surfaces	2
5	The Weeknd	2
6	Alesso	2
7	Marshmello	2
8	Tai Verdes	2
9	OneRepublic	2
10	John Legend	2
11	Jason Derulo	1
12	BANNERS	1
13	Camila Cabello	1
14	benny blanco	1

There are 10 artists who have 2 tracks on the playlist, which are the greatest number of tracks on the playlist in the current database.

Query 2: What are the release dates of Marshmello's two tracks on the playlist?

In [261…
```
pd.read_sql("""
    SELECT spotify_df.track_name, spotify_df.artist_name, spotify_df.release_date
    FROM spotify_df
    WHERE spotify_df.artist_name = 'Marshmello'
    """, db_engine)
```

Out[261…

	track_name	artist_name	release_date
0	OK Not To Be OK	Marshmello	2020-09-10
1	Leave Before You Love Me (with Jonas Brothers)	Marshmello	2021-05-21

Query 3: Count the number of tracks in the playlist that contain explicit contents.

In [262…
```
pd.read_sql("""
    SELECT COUNT(*)
    FROM spotify_df
    WHERE spotify_df.explicit_content = 'TRUE'
    """, db_engine)
```

Out[262…

	count
0	12

Query 4: Which 10 tracks' related contents on YouTube have the most views?

In [228…
```
pd.read_sql('''
    SELECT youtube_df.track_name, youtube_df.youtube_views
    FROM youtube_df
    ORDER BY youtube_df.youtube_views DESC LIMIT 10
    ''', db_engine)
```

Out[228…

	track_name	youtube_views
0	STAY (with Justin Bieber)	5232250420
1	Levitating (feat. DaBaby)	5032276650
2	Watermelon Sugar	2883769160
3	Heat Waves	2524568090
4	Cold Heart - PNAU Remix	2333816600
5	Head & Heart (feat. MNEK)	2328607480
6	Save Your Tears (with Ariana Grande) (Remix)	1997663290
7	My Universe	1917765650
8	Shivers	1654463310
9	Love Again	1597556210

Query 5: Get the lyrics of the track with the highest popularity score on Spotify.

In [235…

```
pd.read_sql('''
    SELECT spotify_df.track_name,spotify_df.spotify_popularity, musixmatch_df.lyrics
    FROM spotify_df
    JOIN musixmatch_df
    ON spotify_df.track_name = musixmatch_df.track_name
    GROUP BY spotify_df.track_name,spotify_df.spotify_popularity,musixmatch_df.lyrics
    ORDER BY spotify_df.spotify_popularity DESC LIMIT 1
    ''', db_engine)
```

Out [235…

	track_name	spotify_popularity	lyrics
0	THATS WHAT I WANT	97	One, two, three, four Need a boy who can cudd...

Query 6: Get the relevant tweet strings of the track with the longest duration.

In [263…

```
pd.read_sql('''
    SELECT spotify_df.track_name, spotify_df.duration, twitter_df.tweets
    FROM spotify_df
    JOIN twitter_df
    ON spotify_df.track_name = twitter_df.track_name
    GROUP BY spotify_df.track_name, spotify_df.duration, twitter_df.tweets
    ORDER BY spotify_df.duration DESC LIMIT 1
    ''', db_engine)
```

Out [263…

	track_name	duration	tweets
0	All I Know So Far	277413	RT @BaddCompani: It Is What It Is✌\n\nP!NK - A...

6.0 EXPLORATORY DATA ANALYSIS

In this section, some exploratory analyses including data visualisations are conducted to illustrate a rough overview of the data in the database.

6.1 MERGING THE DATASET

As the four dataframes all have one-to-one relationship with each other with the track_name being the foreign key, they are joined together into a combined dataframe for a more complete overview. The head of the combined dataframe is presented in Figure 17.

	track_name	spotify_id	artist_name	album	spotify_popularity	release_date	duration	explicit_content	lyrics	tweets	youtube_views	youtube_likes	youtube_favourites	youtube_comments
0	One Right Now (with The Weeknd)	00Blm7zeNqgYLPtW6zg8cj	Post Malone	One Right Now	92	2021-11-05	193506	True	Na-na-na-na, na-na Na-na-na-na, oh no Yeah, ye...	I'm obsessed with this bop by The Weeknd and P...	126618090	2977580	0	104760
1	dancing in the kitchen	0ohcCrxZkBfFbkuRPOZQZX	LANY	dancing in the kitchen	76	2021-06-25	208599	False	City lights looking like ice underneath the st...	Hi everyone! One of my favorite songs is danci...	42519900	1020850	0	41860
2	Sheesh!	3ddNKnYpVx0ul8vcwbTQ5Y	Surfaces	Sheesh!	75	2021-08-20	148846	False	You know what I'm sayin'? (Sheesh) I be like ...	Surfaces & Tai Verdes - Sheesh! RT @kevs_s...	12180970	252720	0	13190

Figure 17: Combined Dataframe

The combined dataframe 76 rows and 14 columns, which contain all the attributes from the four dataframes.

6.2 DATA OVERVIEW

In the combined dataset four attributes are string object, one is a boolean, one is in datetime format, and the rest of them are all integers, which matches the data type specified in the schema.

```
track_name                      object
spotify_id                      object
artist_name                     object
album                           object
spotify_popularity               int64
release_date           datetime64[ns]
duration                         int64
explicit_content                  bool
lyrics                          object
tweets                          object
youtube_views                    int64
youtube_likes                    int64
youtube_favourites               int64
youtube_comments                 int64
dtype: object
```

Figure 18: Column Data Types

For the integer attribute, the summary statistics are displayed in Figure 19.

	spotify_popularity	duration	youtube_views	youtube_likes	youtube_favourites	youtube_comments
count	76.000000	76.000000	7.600000e+01	7.600000e+01	76.0	7.600000e+01
mean	76.894737	191788.078947	5.748229e+08	9.046667e+06	0.0	2.885083e+05
std	18.481399	29788.918409	1.009646e+09	1.699900e+07	0.0	6.261912e+05
min	1.000000	132000.000000	3.687000e+03	1.480000e+02	0.0	3.000000e+00
25%	75.000000	170250.750000	4.116204e+07	6.713475e+05	0.0	1.565750e+04
50%	80.000000	193018.000000	1.664838e+08	2.697695e+06	0.0	7.318000e+04
75%	87.000000	211233.750000	5.247465e+08	7.623832e+06	0.0	2.481975e+05
max	97.000000	277413.000000	5.232250e+09	8.669817e+07	0.0	4.220280e+06

Figure 19: Summary Statistics

The *youtube_favourites* column only contains values of zeros. After some research in different online communities, it was found that many users have claimed that the “favourite” function has secretly removed by YouTube, which was probably why the YouTube API could only retrieve values of zeros for this attribute.

6.3 DATA VISUALISATION

To show the distribution of the data more clearly, histograms are plotted for several attributes.

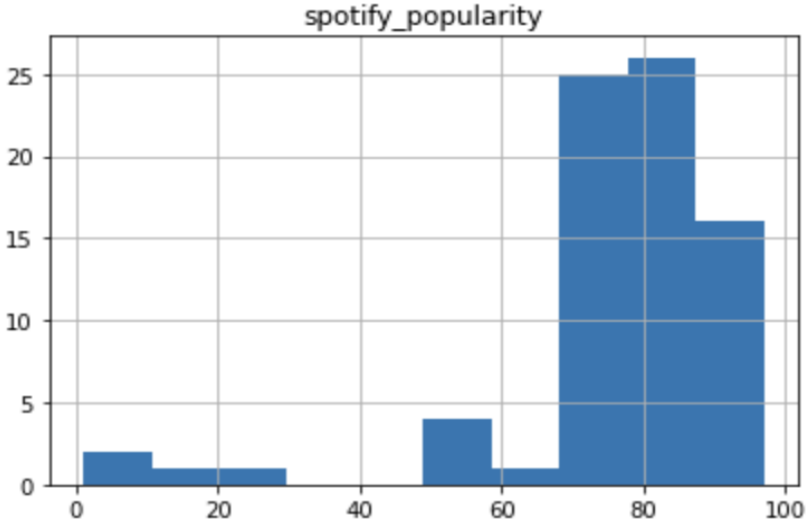


Figure 20: *Spotify_popularity* Distribution

The popularity of the songs in the playlist on Spotify is left skewed and aggregates at around 70 to 90. The Spotify popularity Index is a 0-to-100 score (Loudlab, 2022) that indicates how popular is the song compared to other songs on Spotify. The plot may suggest that the many songs in this playlist are almost the most popular songs on Spotify.

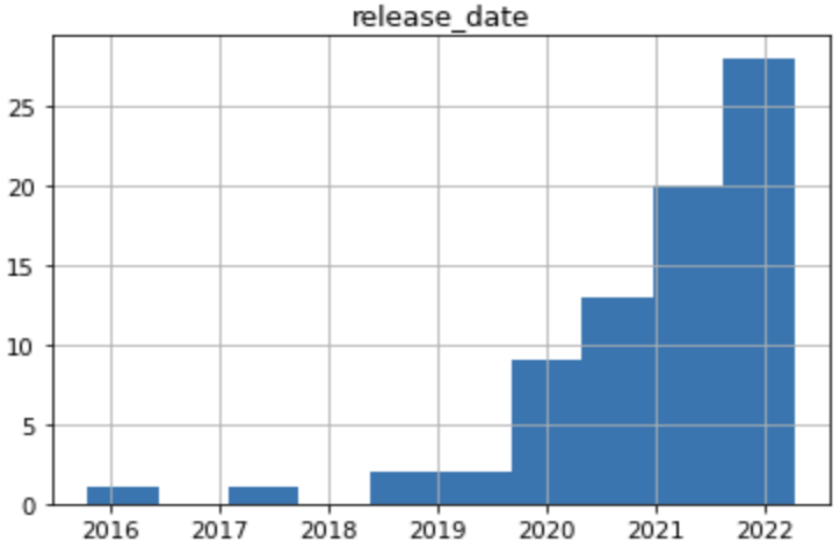


Figure 20: *Release_date* Distribution

The histogram in Figure 20 suggests that the release date of the songs in the playlist are aggregated in between 2020 to 2022. This indicates that the songs in the playlist are very new and up-to-date.

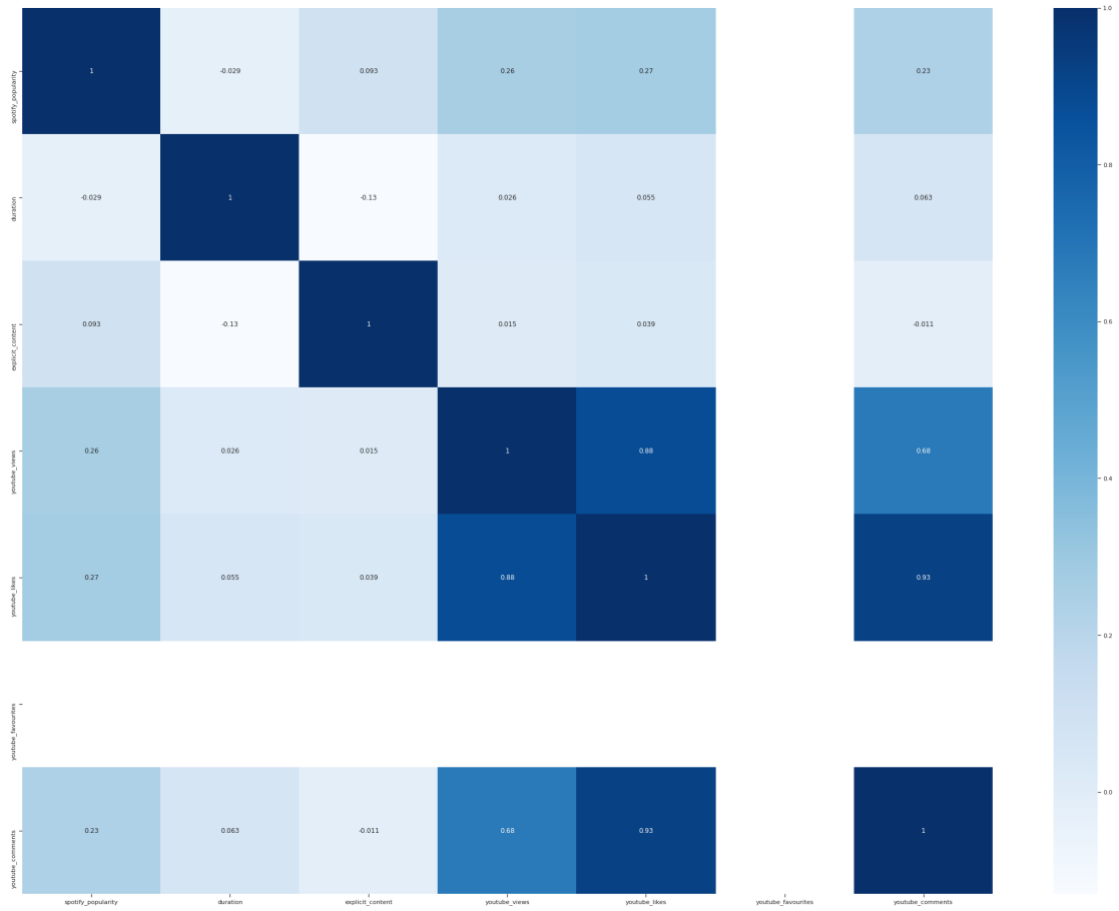


Figure 21: Multicollinearity Matrix

There are only very slight multicollinearities between the integer attributes in the dataset, apart from the three YouTube attributes with real values.

7.0 SENTIMENT ANALYSIS

One of the main objectives of this project is to conduct sentiment analysis on the data collected. In this section, two methods will be displayed to analyse the emotions associated with the lyrics and relevant tweets of the songs in the *Mood Booster* playlist.

7.1 WORDCLOUD

Wordclouds are visualisations of keywords used to aggregate and present textual contents. By generating word clouds for the most frequent keywords in the lyric and tweet strings in the database, the general emotions, and vibes of the songs in the playlist can be analysed.

This project used the Python *wordcloud* package to plot the wordclouds. To make the result valid and only contain meaningful words from the lyric and tweet strings, a few stopwords were defined in advance to avoid being shown in the wordclouds. A default list of stopwords such as “a”, “and” and “if” was set first and unique Twitter characters like “RT”, “@“, “https” were also updated in the stopwords list. The strings in the list would not show up in the wordclouds.

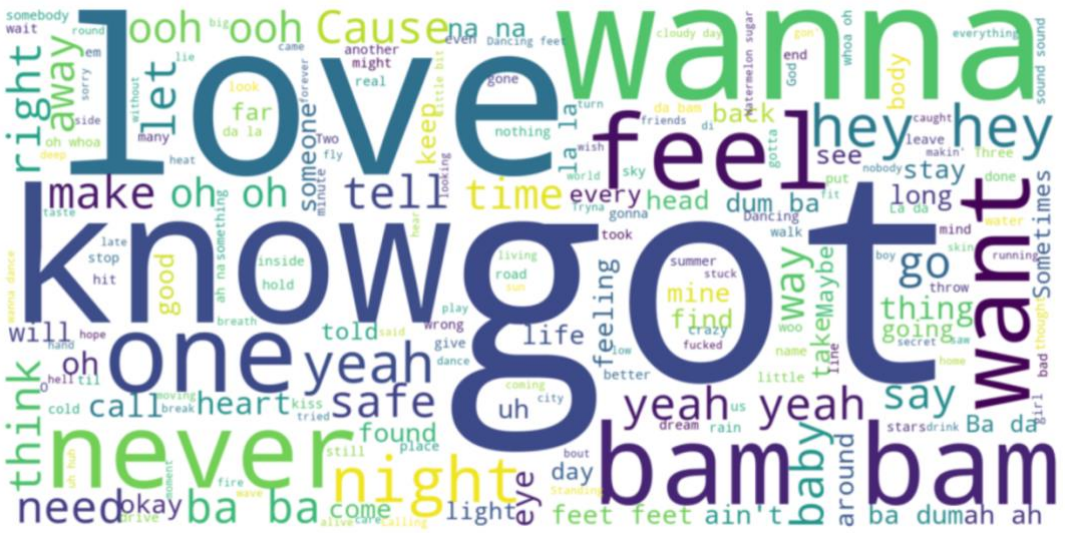


Figure 22: Lyrics Wordcloud

Out of the most frequent words in the lyrics of the songs, most of them give a rather neutral feeling. “Love” is probably the most apparent word that actually has a positive meaning. A lot of the filler words like “oh”, “bam” and “hey” don’t really have a meaning inside but tend to appear more in those songs with faster beats. No words that have an obvious negative feeling like “tears”, “sad” or “cry” are detected.

Figure 23: Tweets Wordcloud

The wordcloud for the tweet strings looks a lot messier than the one for the lyrics, since they are from Twitter users’ posts which contain many informal uses of language, while most lyrics of a song are written with consideration and have gone through content check. The most meaningful positive word is still “Love”, with “Fancy and “Best” that also suggests a positive feeling of Twitter users when they discuss about the songs in the playlist.

7.2 NLTK SENTIMENT ANALYSIS

As the two wordclouds don't really lead to an overall conclusion of the emotions associated with the songs, this project decided to use the Natural Language Tool Kit to further explore the linguistic data. By employing the NLTK algorithms through their powerful built-in machine learning operations to obtain insights from the textual data (Mogyorosi, 2022), a sentiment analysis of scoring the positive and negartive engagement of the data can be done.

By utilising the *SentimentIntensityAnalyzer* from the *nltk.sentiment.vader* library, the score of "positive", "neutral" and "negative" feelings associated with the data can be calculated. By looping through the 76 rows of musixmatch_df (the lyrics dataframe), three columns were appended to the end of the dataframe and the values of the columns represent the corresponidng emotion scores for each song.

	track_name	artist_name	lyrics	negative	neutral	positive
0	One Right Now (with The Weeknd)	Post Malone	Na-na-na-na, na-na Na-na-na-na, oh no Yeah, ye...	0.160	0.635	0.205
1	dancing in the kitchen	LANY	City lights looking like ice underneath the st...	0.043	0.822	0.135
2	Sheesh!	Surfaces	You know what I'm sayin'? (Sheesh) I be like ...	0.017	0.823	0.160
3	Can I Get It	Adele	Pave me a path to follow And I'll tread any da...	0.106	0.698	0.196
4	Black And White	Niall Horan		0.000	0.000	0.000

Figure 24: Lyrics Dataframe with 3 Emotion Scores

The average positive score of the lyrics is 0.138, ranging from 0 to 0.452, while the average negative score is 0.06, ranging from 0 to 0.34.

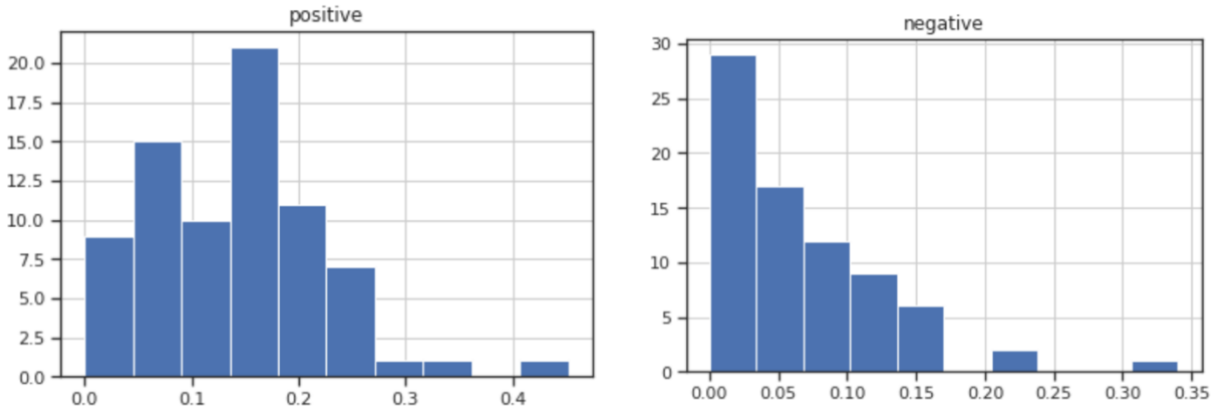


Figure 25: Lyrics Emotion Scores Distribution

Same methods were applied to the Twitter dataframe and emotion scores were appended to reach row.

	track_name	artist_name	tweets	negative	neutral	positive
0	One Right Now (with The Weeknd)	Post Malone	I'm obsessed with this bop by The Weeknd and P...	0.042	0.958	0.000
1	dancing in the kitchen	LANY	Hi everyone! One of my favorite songs is danci...	0.011	0.835	0.154
2	Sheesh!	Surfaces	Surfaces & Tai Verdes - Sheesh! RT @kevs_s...	0.031	0.837	0.132
3	Can I Get It	Adele	RT @retriever_lover: Adele - Can I Get It (Off...	0.008	0.902	0.091
4	Black And White	Niall Horan		0.000	0.000	0.000

Figure 26: Tweet Dataframe with 3 Emotion Scores

The average positive score of the lyrics is 0.1, ranging from 0 to 0.525, while the average negative score is 0.02, ranging from 0 to 0.246.

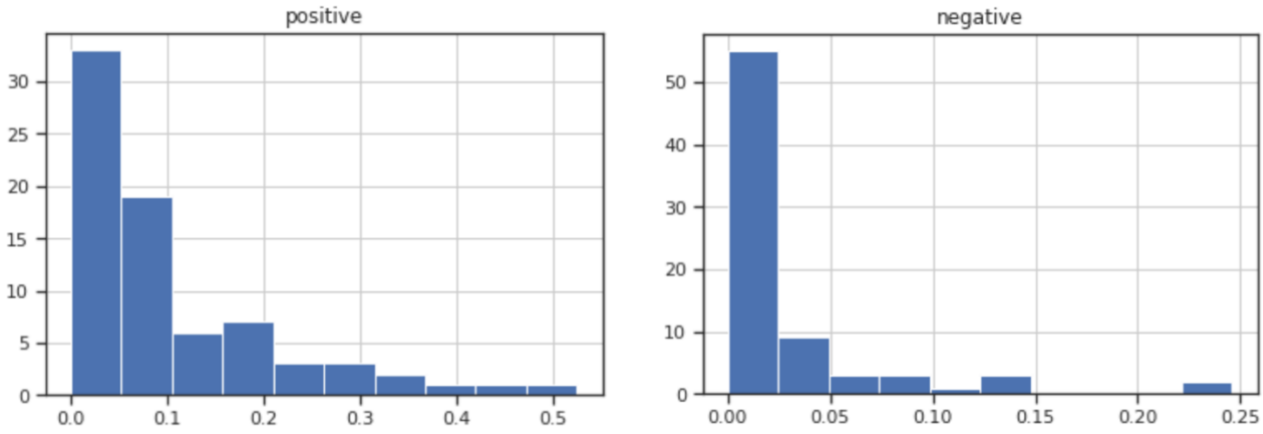


Figure 27: Tweets Emotion Scores Distribution

Generally, the mean and maximum score for the negative emotion in both lyrics and tweet strings are much lower than the positive score, indicating the songs in the playlist have positive vibes in themselves, and will bring positive feelings to their listeners.

In Figure 28 and 29, stacked bar plots are plotted to display the disbution of positive and negative emotions for both dataframes.

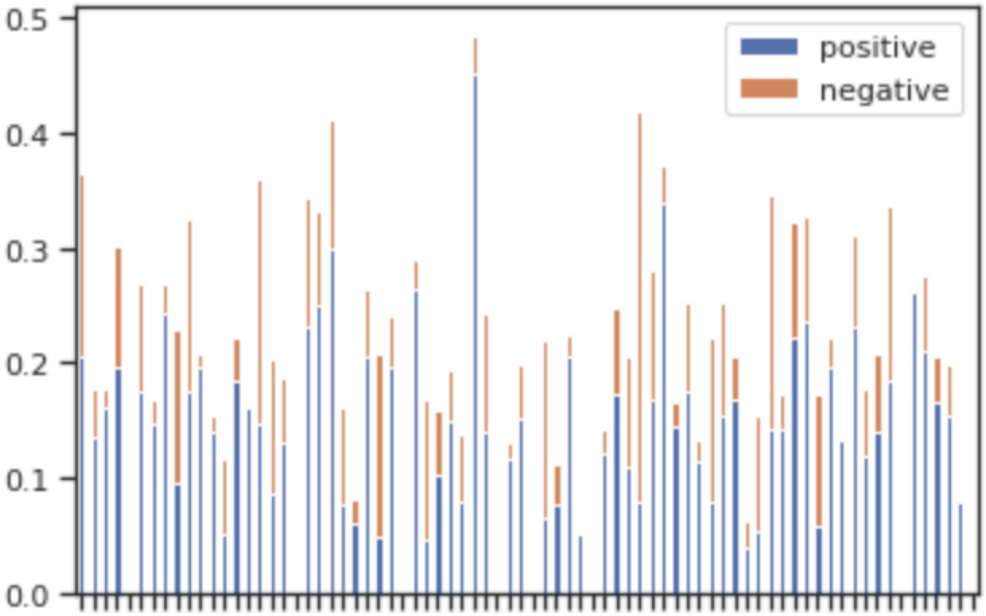


Figure 28: Lyric Emotions Stacked Plot

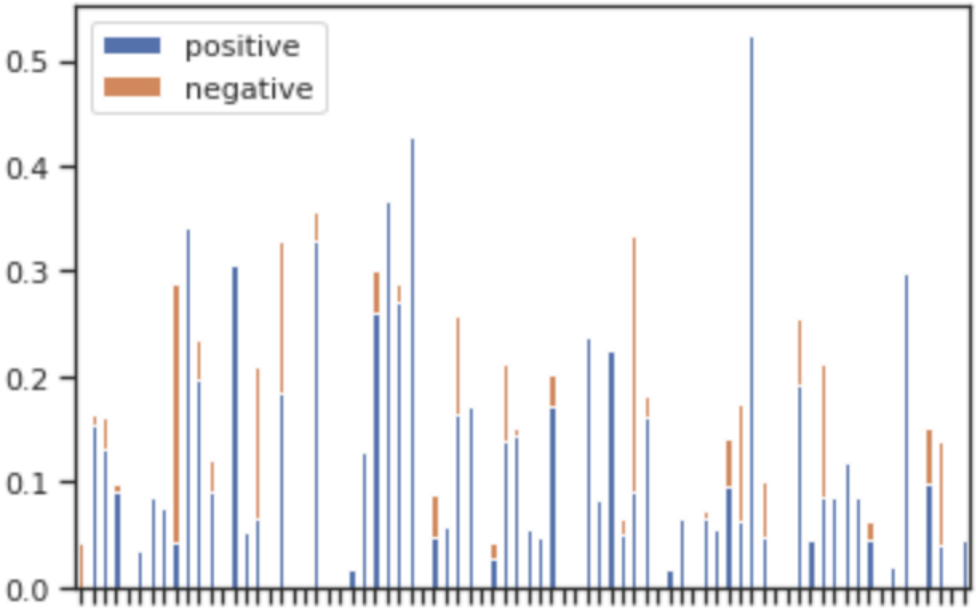


Figure 29: Tweet Emotions Stacked Plot

Out of all 76 songs, 55 have more positive words contained in the lyrics, and 50 have more positive words contained in the related tweet strings. This result indicates that the songs generally give a more positive vibe to their listeners, which matches the description initiated in the playlist.

8.0 REGRESSION MODEL

After the sentiment analysis, it is also worthwhile to build a simple regression model to explore the possible associations among the variables in the database. Due to the time limitation, only linear regression is conducted.

8.1 DATA PREPARATION

All string objects are dropped from the dataset and the value of the boolean variable "*explicit_content*" are replaced by the integer 1 (TRUE) and 0 (FALSE).

Apart from the data retrieved by website APIs, the positive scores calculated in the previous section for the lyrics and tweets are also inserted in the dataset. Therefore, the final dataset used for model training looks likes this:

	spotify_popularity	duration	explicit_content	youtube_views	youtube_likes	youtube_comments	positive_lyrics	positive_tweets
0	92	193506	1	126618090	2977580	104760	0.205	0.000
1	76	208599	0	42519900	1020850	41860	0.135	0.154
2	75	148846	0	12180970	252720	13190	0.160	0.132
3	82	210384	0	52026180	1012500	22110	0.196	0.091
4	78	193089	0	126128120	4259400	169780	0.000	0.000

Figure 30: Dataset for Machine Learning

The dependent variable of the regression will be *youtube_likes*. The project would like to know other attributes of the song are correlated with the listener's intentions of watching and liking relevant videos on YouTube.

8.2 MODEL PERFORMANCE

This project used the *Scikit-learn Linear Model* package to build the linear regression model and used 70% of the final dataset as the training data. The R-squared of the model is 0.96, which means over 96% of the data points in the dataset could be explained by the linear regression model built, which is a pretty good performance.

Figure 31 displays the OLS Regression Results of the linear model.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.975
Model:	OLS	Adj. R-squared:	0.973
Method:	Least Squares	F-statistic:	380.3
Date:	Mon, 18 Apr 2022	Prob (F-statistic):	6.46e-52
Time:	01:06:08	Log-Likelihood:	-1232.3
No. Observations:	76	AIC:	2481.
Df Residuals:	68	BIC:	2499.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.028e+06	2.61e+06	-0.776	0.440	-7.24e+06	3.19e+06
X[0]	6445.8835	1.84e+04	0.351	0.727	-3.03e+04	4.31e+04
X[1]	6.4724	11.121	0.582	0.562	-15.719	28.664
X[2]	1.761e+06	9.06e+05	1.943	0.056	-4.72e+04	3.57e+06
X[3]	0.0077	0.000	17.224	0.000	0.007	0.009
X[4]	16.6563	0.715	23.292	0.000	15.229	18.083
X[5]	-2.553e+06	3.91e+06	-0.652	0.516	-1.04e+07	5.26e+06
X[6]	1.909e+06	2.96e+06	0.645	0.521	-4e+06	7.82e+06

Omnibus:	17.685	Durbin-Watson:	2.084
Prob(Omnibus):	0.000	Jarque-Bera (JB):	82.887
Skew:	-0.308	Prob(JB):	1.00e-18
Kurtosis:	8.079	Cond. No.	1.43e+10

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.43e+10. This might indicate that there are strong multicollinearity or other numerical problems.

Figure 31: OLS Regression Results

8.3 RESULT PRESENTATION

By listing and ranking the coefficients of each regressor, this project can interpret which attribute of a song is mostly associated with the relevant YouTube like counts and the magnitude of the impact.

	variable	coefficient
5	positive_lyrics	-3.117497e+06
6	positive_tweets	1.588545e+06
2	explicit_content	3.404405e+05
0	spotify_popularity	7.192198e+03
4	youtube_comments	1.990951e+01
1	duration	1.972561e+01
3	youtube_views	5.611690e-03

Figure 32: Regression Coefficients

The positivity of the lyrics has the largest association with the YouTube views out of all 7 regressors, however, by a negative correlation. Surprisingly, the model suggests that the more positive the lyrics are, the less listeners will like relevant videos on YouTube.

The second important attribute is the positivity in Tweet strings and by a positive coefficient. As Tweets could often be seen as the express of feelings of a listener, it could be interpreted as the more positive listener's feelings of a song, the more they will like relevant videos on YouTube.

If a song contains explicit content, people will tend to like relevant videos on YouTube more. *Spotify_popularity* is the fourth important feature, and it matches the common sense of popularities are likely to be balanced across different platforms.

The duration of a song and the comment and view count of the relevant videos have the least association with the YouTube like count.

9.0 FUTURE OPPORTUNITIES

Due to the limitation in the time scope, the project didn't proceed any further, however, it is believed there are various possibilities of this database beyond the current project scope.

9.1 DATA PIPELINE

The current database only contains one playlist worth of information. However, with the current data pipeline, the whole process of this project can be repeated automatically. Apache Airflow may be used to achieve this. The frequency of retrieving data can be defined based on the needs, for example, the pipeline may automatically retrieve data from the pre-defined sources every day, week, or month.

9.2 DATASETS

Apart from the 4 websites that were used as the database source, it is also possible to enrich the database with more data from other websites, for example, we may compare the popularity of a song across Spotify and other music streaming services like Apple Music. It is also worthwhile to know the other playlists a song is featured given that it appears in the current playlist investigated to explore the standard of a song being featured on playlists.

9.3 SENTIMENT CLASSFICATION

The project used the NLTK package for a simpler and more automated sentiment analysis. It is also possible to use Natural Language Processing techniques such as Vector Space Model to pre-process the lyrics and use Naïve Bayes and Support Vector Machine to conduct sentiment classfication models. Other than “positive” and “negative” that were used in the project, more different emotion categories may be considered, for example: “light-hearted” and “heavy-hearted”, ‘hyperbeat” and “relaxing”, or “motivating” and “restful”.

10.0 CONCLUSION

10.1 PROJECT VALUE

As music streaming services become more demanding since the last decades, it is important for the service providers to build and improve a more accurate music recommendation system. The sentiment analysis of songs is the key technique of music classification and intelligent recommendation, as assigning specific emotion tags to songs will be the key for user querying and recommendation systems.

This project built a database that contains different attributes (including the basic track attributes, lyrics, popularity, and user reviews) of the songs in the chosen Spotify playlist *Mood Booster*, and successfully conducted sentiment analysis of the lyrics and Twitter reviews of the songs. This creates values and directions for music streaming service providers to collect information about large numbers of songs and conduct sentiment analysis for them. With more information collected, the emotion labels attached to the songs would be more accurate and the recommendation systems will tend to give more tailored music recommendations for better user experience.

10.2 PROJECT LIMITATIONS

The biggest limitation of the project is that it only uses lyrics and user reviews as the sources for sentiment analysis. However, a song may express the emotions via different aspects, for example, the melody, the music instrument used, or the way the artist sings the song. Therefore, there are problems of one-sidedness in labelling the emotions of a song only by two sources. Some songs do not have obvious emotional tendencies from the lyrics, but after being performed by the singer, they can express strong emotions. This part of the data is usually hard for the algorithms to learn, though in the project it was compensated by the analysis of user. However, sentiment analysis is always a subjective classification method, it will thus lead to bias in the experimental results.

There are also limitations in the data collected. Due to budget restrictions, only 30% of the lyrics was collected from MusixMatch and the training set only has a very limited data size. Therefore, there might be bias in the results due to sample size insufficiency.

11.0 BIBLIOGRAPHIES

Schriewer, K. and Bulaj, G., 2016. Music Streaming Services as Adjunct Therapies for Depression, Anxiety, and Bipolar Symptoms: Convergence of Digital Technologies, Mobile Apps, Emotions, and Global Mental Health. Frontiers in Public Health, 4.

Caddy, B., 2022. *The best music streaming services 2022: Spotify, Apple Music, Tidal and more*. [online] TechRadar. Available at: <https://www.techradar.com/best/the-best-music-streaming-services-2021> [Accessed 18 April 2022].

Medium. 2022. *Using an API for Web Scraping: A List of the Best Advantages*. [online] Available at: <https://medium.com/api-world/using-an-api-for-web-scraping-a-list-of-the-best-advantages-586e9fec2660> [Accessed 19 April 2022].

Soundplate. 2022. *What Does 'Explicit Content' Mean on Spotify, Apple Music & Other Music Streaming Platforms*. [online] Available at: <https://soundplate.com/what-does-explicit-content-mean-on-spotify-apple-music-other-music-streaming-platforms/> [Accessed 19 April 2022].

Baydeer, J., 2021. Let the Music Speak. [online] Medium. Available at: https://medium.com/swlh/let-the-music-speak-8c524ed45809 [Accessed 9 April 2022].

Igi-global.com. 2022. *What is Twitter | IGI Global*. [online] Available at: <https://www.igi-global.com/dictionary/i-found-myself-retweeting/30754> [Accessed 20 April 2022].

PostgreSQL. 2022. *PostgreSQL*. [online] Available at: <https://www.postgresql.org/> [Accessed 20 April 2022].

Loudlab. 2022. *Spotify Popularity Index: A Little Secret to Help You Leverage the Algorithm*. [online] Available at: <https://www.loudlab.org/blog/spotify-popularity-leverage-algorithm/> [Accessed 21 April 2022].

Mogyorosi, M., 2022. *Sentiment Analysis: First Steps With Python's NLTK Library – Real Python*. [online] Realpython.com. Available at: <https://realpython.com/python-nltk-sentiment-analysis/> [Accessed 22 April 2022].