DATA ENGINEERING INDIVIDUAL COURSEWORK

SPOTIFY PLAYLIST DATABASE:

A SENTIMENT ANALYSIS

April 2022

Word Count:

Repository Link:

TABLE OF CONTENTS

1.0 INTRODUCTION

"Music can heal the wounds which medicine cannot touch", says Debasish Mridha. From many decades ago, music has already been recognised as an opportunity to address mental health challenges (Schriewer and Bulaj, 2016). Nowadays, audience can enjoy the music more conveniently via music streaming services, instead of downloading the original 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 that somehow have some similarities so that meaningful playlists are created. Many playlists have emotion-related tags such as "happy", "moody" and "chilling", and each of them is designed to match the audience's emotions.
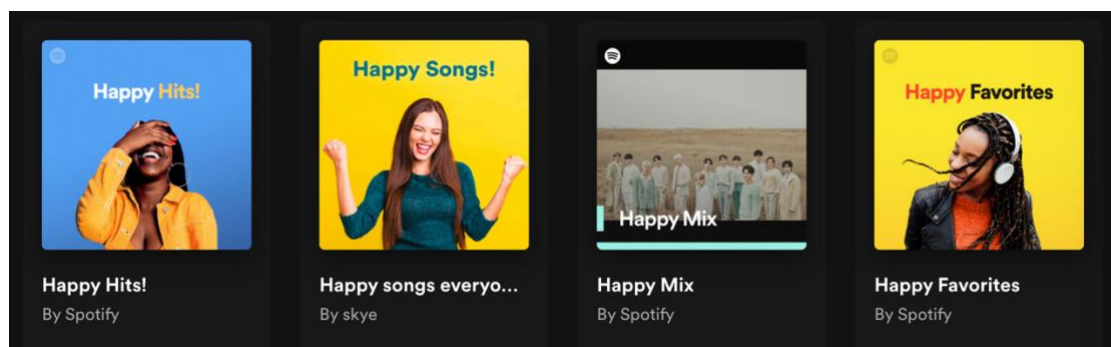

Figure 1: Spotify Curated Playlists Related to "Happy"

While the Spotify playlist is an important feature of the platform in helping the artists reach more of the target audiences, it is worthy of investigation that the exact criterion of how a song is featured on a certain playlist. This project aims to create a database of a specific Spotify playlist that contains a number of 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 sentiment related research on a bigger database beyond the scope of this project.
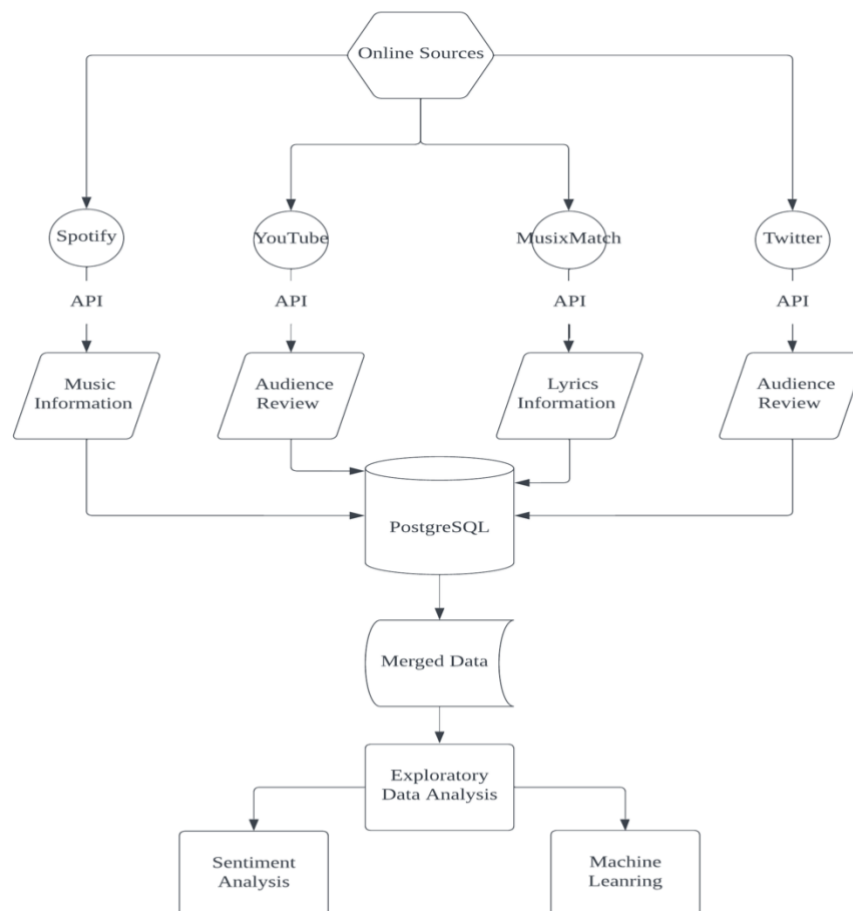

Figure 2: Project Workflow

2.0
DATA COLLECTION

This project mainly used the Application Programming Interface (API) techniques in the 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

As the main subject matter of this project, 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 the developer website, a Python library called Spotipy (https://github.com/plamere/spotipy) is recommended for executing the API calls. This package was utilised by this project throughout the data extraction from Spotify.

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 interested in the if the emotions of the songs in this playlist actually match that purpose. To dig deeper into the tracks in this playlist, the playlist's ID on Spotify needs to be known. Unfortunately, at the moment, 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 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 a "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 further sentiment analysis on the lyric strings.

With the free API plan provided by MusixMatch, the account created was limited to 2,000 API calls per day, and 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 feelings 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 Lyrics 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-GrmUAIX5NV9hGhWN8EfDpZJjEUKAGI1qv2IK_txinFDOwICCmguLzb6ubcA1NPT8vooFsy8SVHhdP1XzhRFTKQjU6Kom3PrrAcarzrp20Og2wa_uZbXCnsCfxWzK1r37BdzYf9bNyvdtAg4TWdAEhHVqCtsoQQqgelzNgedClBRumKD7USjuwb1OEfmAs7y33DidTcj4mq9HbB-wc3O6N2VZ-FpPMvTMqOyFVFWzKBUfP1AbjUHodSkYjQ83eMER1GFDm2QKIAul_eNdcV7S_119kqkWM37bA0HWxwYF2ydULNQ5tQrLVWFPvN1J45D-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_-D8dTbgLa7cuSMifMTbKe6OxPQcJ6YK_1KrAXRIQt7YROF1IwdFzuAe9V4xL0fOqApGAloQtX6NXo9tQ2kc2tW1WI2HCGzDIU3F-YL1mgJBgjF1q9bzVViKIvtVzkhrJGDYj-eF0XKg-15AUszL8wZDXb1w27eJxtbdcMtj2Vd13BdcMAemjB3d9AH9ajFxS1gNYeyJgcPUrHXC0Bumm6Ldn34a5CX1zejC9ngXqqrEjBKaBfhPqJtB8n-9V4idAQN0HQikALWau4gREfuvoFvd1n9R61KTk1fqv8vG4QjFpp4Q95KYUKczLJsz2W1cisibSu03zUEz4b59reMmpO6xcL2JXDGGrOPVqT7vQoGoBii_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 returned.',
      '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.
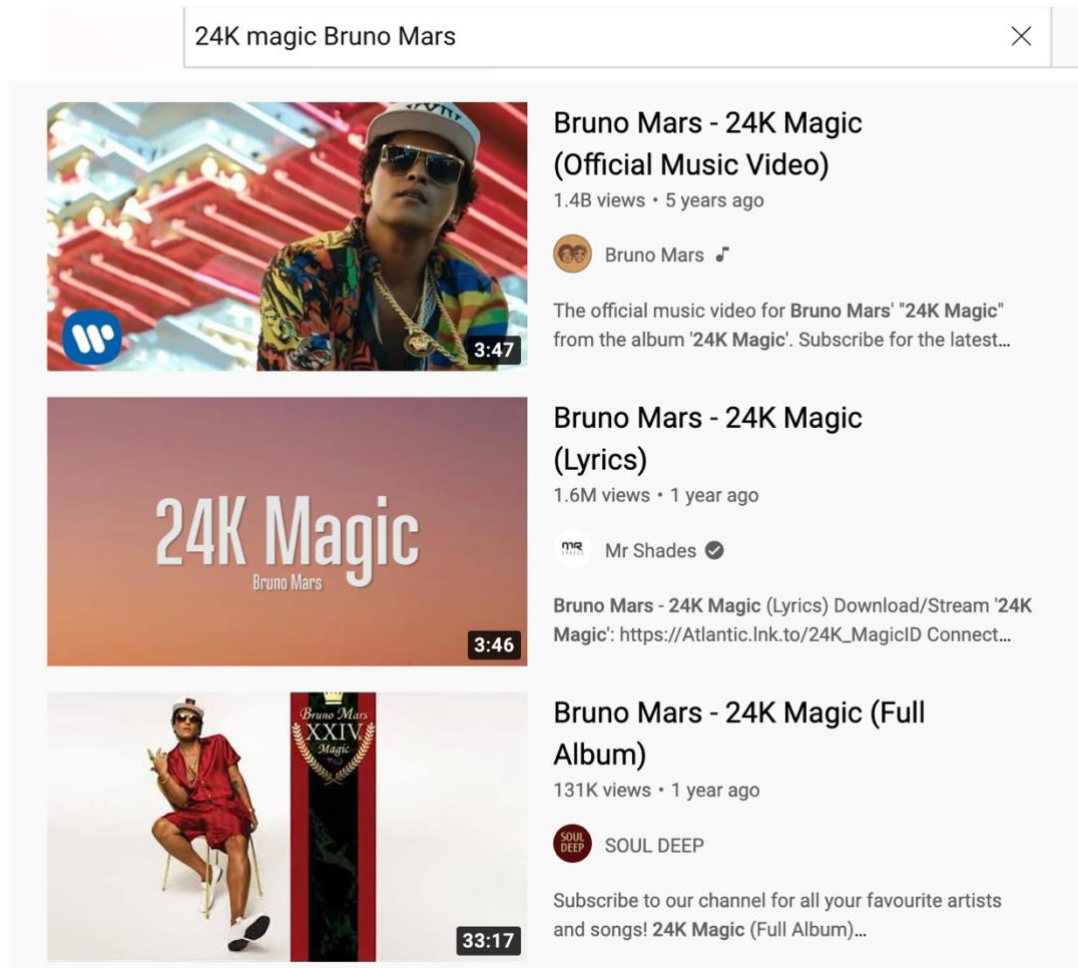
Figure 7: Top3 YouTube Results of Searching "24K Magic" + Bruno Mars

Figure 7 displays the top results of querying the keywords "*24K Magic*" (song 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.

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 developer's 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.

Figure 8: Output Example of the Twitter API

The project is only interested in the "text" section in the requesting result.

Similar to the previous two 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 through these formats in Python, this project was able to generate four dataframes for four of the different platforms. Each dataframe has 76 rows and has the track name and artist's name columns as the foreign keys.

	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

Figure 9: *Spotify_df*

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 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.
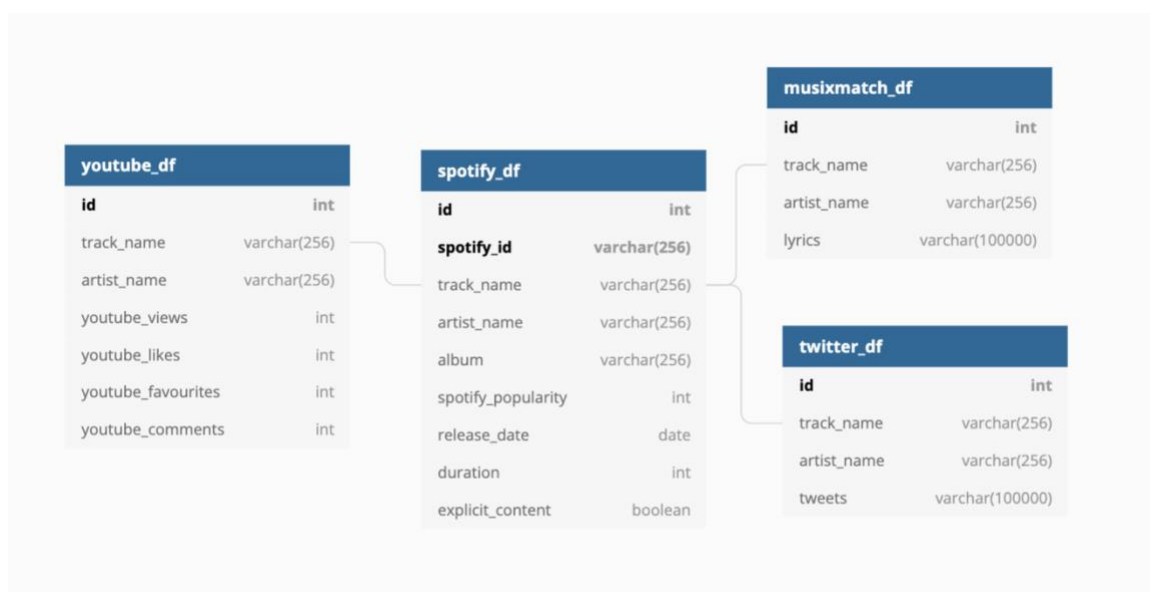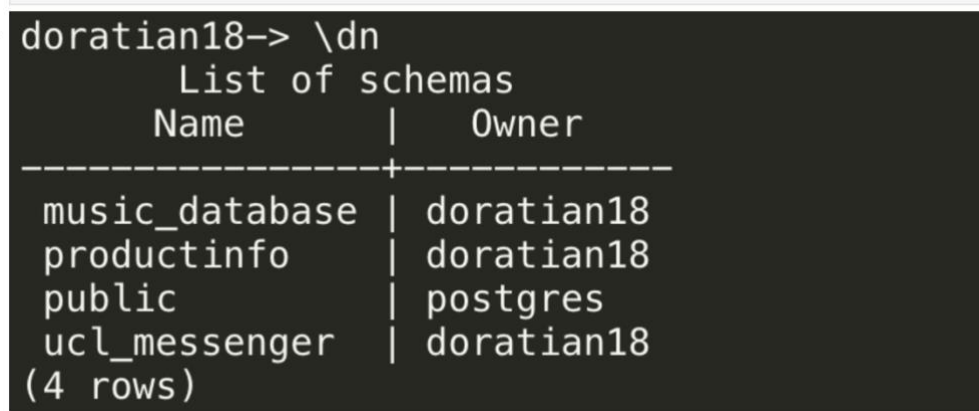

Figure 15: Schema Diagram

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?

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.

14

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

15

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

```python
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)
```

	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.

```python
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)
```

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

16

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!	3ddNKnYpVx0uI8vcwbTQ5Y	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 of the attributes from all of the four dataframes.

6.2 DATA OVERVIEW

Out of the 14 columns 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 zero. 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.
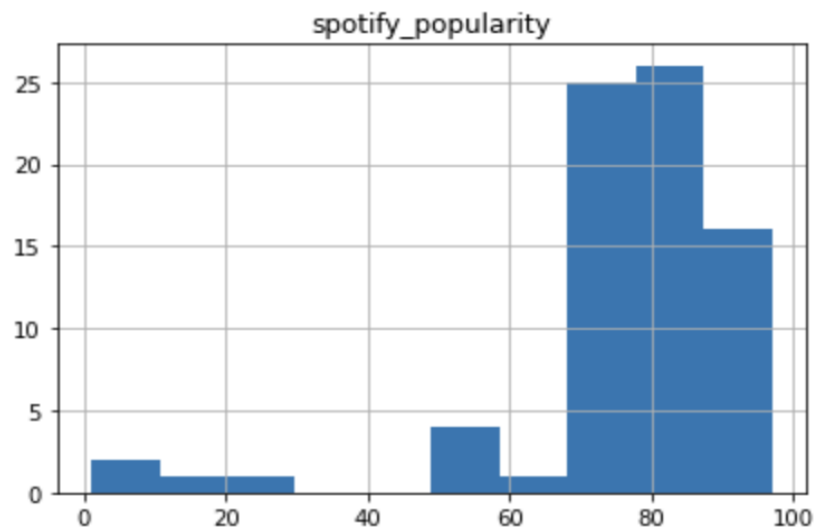

Figure 20: *Spotify_popularity* Distribution

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 compare to other songs on Spotify. The plot may suggest that the many songs in this playlist are almost the most popular songs on Spotify.
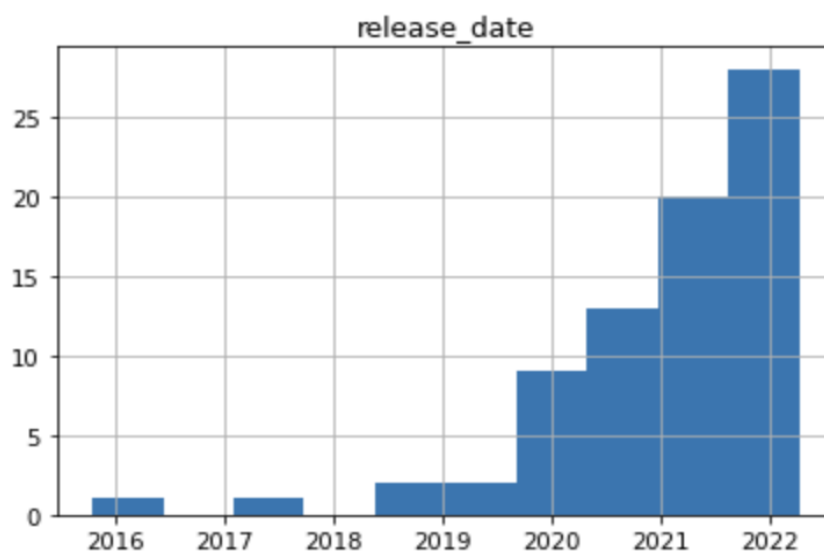

Figure 20: *Release_date* Distribution

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.
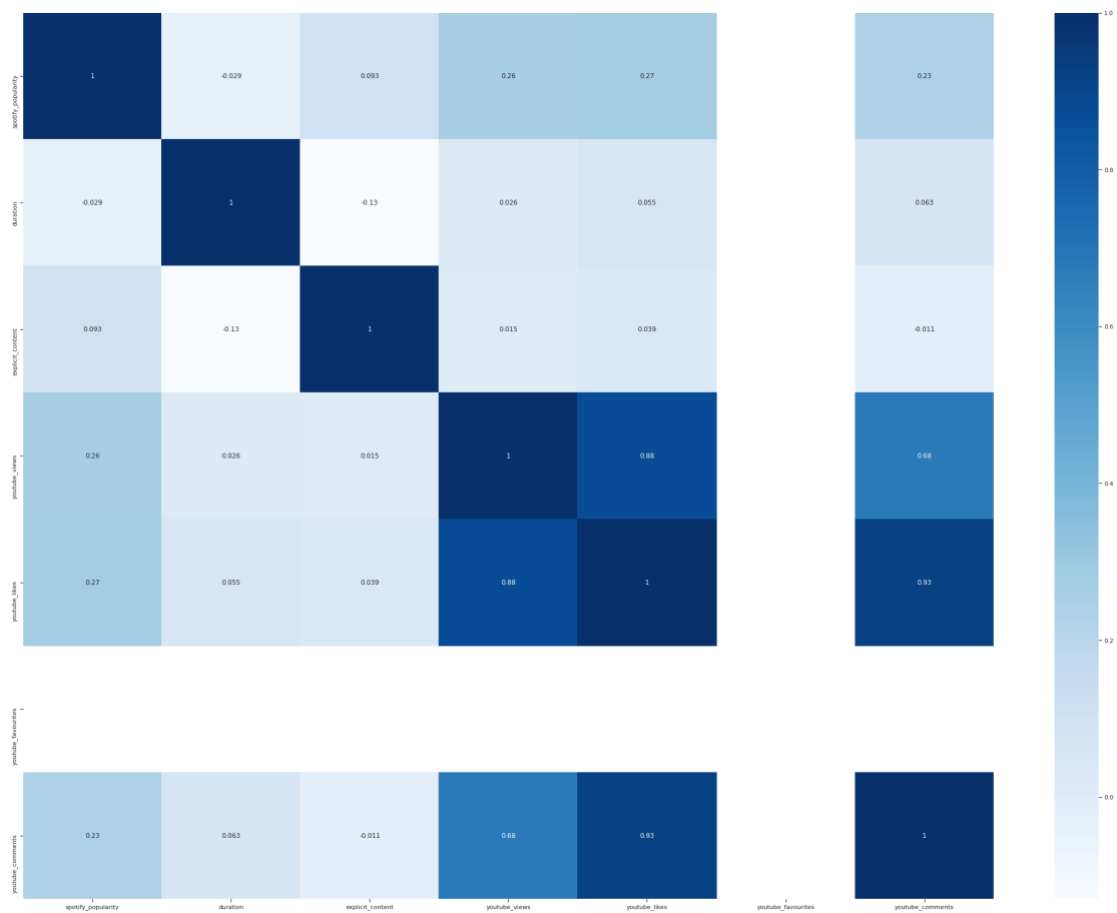
Figure 21: Multicollinearity Matrix

There are only very slight multicollinearities between the integer attributes in the dataset, apart from the three YouTube attributes with real values. This may be explained by that for each comment and likes of a video, the user needs to first watch the video, so that may lead to the correlated increment of the attributes.

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 WORD CLOUD

1234

7.2 NLTK SENTIMENT ANALYSIS

1234

8.0
REGRESSION MODEL

HEADING 2

View and edit this document in Word on your computer, tablet or phone. You can edit text, easily insert content such as pictures, shapes and tables, and seamlessly save the document to the cloud from Word on your Windows, Mac, Android or iOS device.

9.0
FUTURE OPPORTUNITIES

HEADING 2

View and edit this document in Word on your computer, tablet or phone. You can edit text, easily insert content such as pictures, shapes and tables, and seamlessly save the document to the cloud from Word on your Windows, Mac, Android or iOS device.

10.0
CONCLUSION

HEADING 2

View and edit this document in Word on your computer, tablet or phone. You can edit text, easily insert content such as pictures, shapes and tables, and seamlessly save the document to the cloud from Word on your Windows, Mac, Android or iOS device.

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].

12.0
APPENDIX

HEADING 2

View and edit this document in Word on your computer, tablet or phone. You can edit text, easily insert content such as pictures, shapes and tables, and seamlessly save the document to the cloud from Word on your Windows, Mac, Android or iOS device.