SI 206 WN 23 Final Project Report

Yuyu Yang 3201 7809
Christina Ng 7340 2943
Github Repo: https://github.com/cwjng/si206-final-project

1. **The goals for your project including what APIs/websites you planned to work with and what data you planned to gather**

The project's objective was to examine Twitter engagement levels with the top Spotify music and artists. The project intended to use the Spotipy and Tweepy APIs to accomplish this purpose. The most streamed musicians and songs of 2021 were identified via the Spotipy API. According to their streaming amounts, intended to compile information on the top musicians and songs and store it in a database. The Tweepy API was then used to determine how many tweets, retweets, and mentions were posted that contained song titles or artists. Additionally to the streaming data, the initiative intended to track the amount of tweets, retweets, and mentions for each artist and song. The project's goal was to discover trends in social media and its relationship with song and artist popularity by examining the Twitter engagement rates for the most popular songs and artists on Spotify.

2. **The goals that were achieved including what APIs/websites you actually worked with and what data you did gather**

Using two APIs, Spotipy and Tweepy, we compared Twitter engagement levels with Spotify popularity. Data for the top artists of 2021, including the number of streams and followers each artist had, was gathered using the Spotipy API. We obtained information about the number of tweets, retweets, and mentions that contained artist's names using the Tweepy API. By analyzing Twitter engagement levels with the top Spotify artists, we got an understanding of the popularity of each artist across social media Twitter.

3. **The problems that you faced**

Problem 1: Repeat of artists in database

- Our original idea for this project was to evaluate correlations between artists of the top 100 songs and the number of artist twitter engagement. However, popular artists like SZA have multiple songs on the top 100 list. Therefore, it was possible for the same artist to be entered multiple times into the database. To solve this issue, we ended up changing our project goal and accessed the top 100 most streamed artists instead. This way we can make sure that there are no repeats within the dataset

Problem 2: Rate limit of the Twitter API

- Twitter has a limit of around 1500 requests per 15 minutes. It was difficult at first to create a code that implemented the capacity properly (25 per run), so it would quickly time out. This heavily held me back from creating the code and debugging. Once the function had been appropriately configured, it would still time out. For example, if I

wanted to run 4 times to get 100 rows, I would be put in a 15 grace period multiple times before finishing.

- With the rate limit, I could not pull many tweets. The capacity was about 200 tweets per artist. This was the only way I could pull a sizable amount of data without taking too much time to request for all 100 artists.

Problem 3: 50 items per request limit for Spotify API

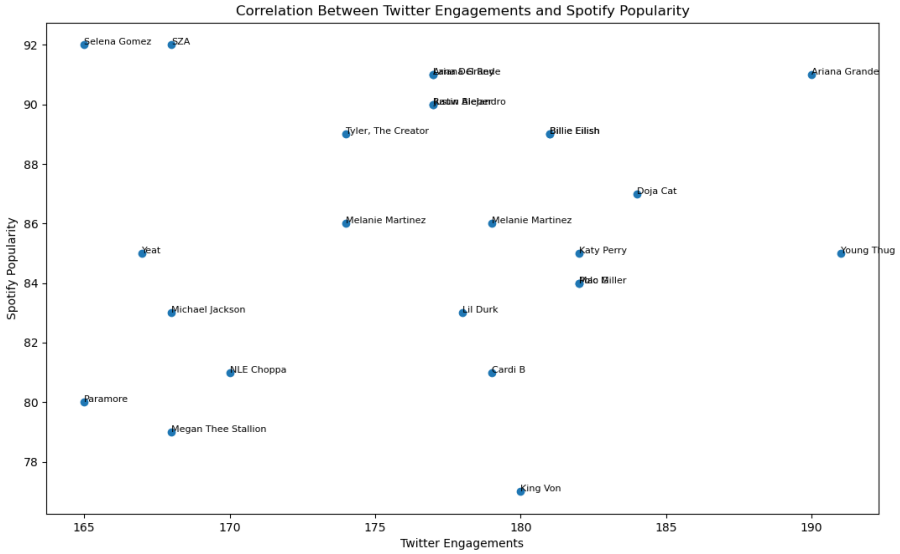
- Spotify API has a limit of 50 items per request. This makes it difficult to gather data for a large number of songs or artists in a single API call. To work around this issue, we use pagination and make multiple requests to the API, each with a different offset, to retrieve all the desired data.

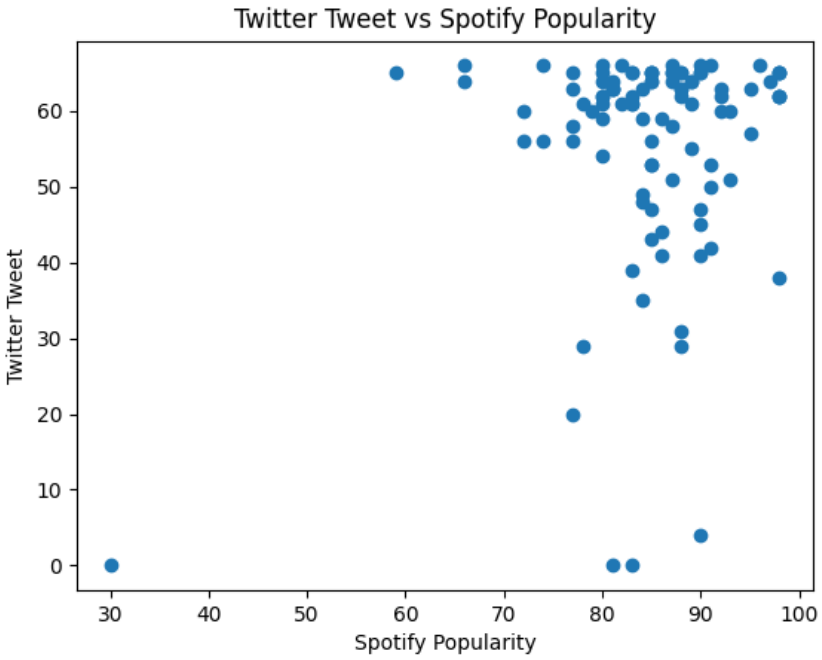
4. The calculations from the data in the database (i.e. a screen shot)

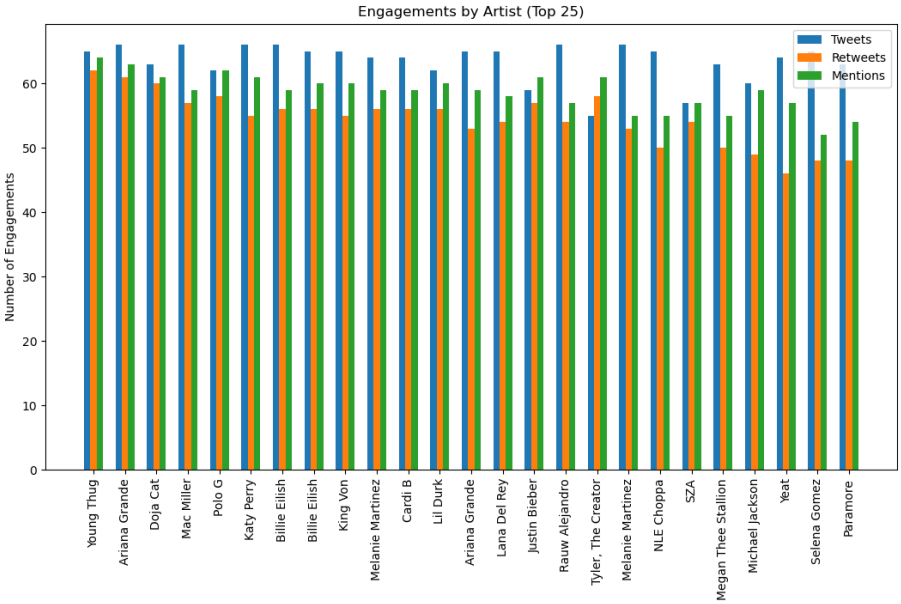
```
name,total engagements
Taylor Swift,161
Drake,158
SZA,168
Lil Baby,116
Juice WRLD,160
"Tyler, The Creator",174
Metro Boomin,159
$uicideboy$,0
Lil Uzi Vert,132
Peso Pluma,129
Zach Bryan,98
Yeat,161
XXXTENTACION,121
Ariana Grande,177
Melanie Martinez,174
KAROL G,162
Kali Uchis,141
Miley Cyrus,135
Brent Faiyaz,109
Billie Eilish,181
Trippie Redd,102
Bruno Mars,142
Playboi Carti,129
Beyoncé,145
JAY-Z,142
Lana Del Rey,177
Post Malone,162
Yeat,167
Nicki Minaj,159
XXXTENTACION,114
Ariana Grande,190
Melanie Martinez,179
Chris Brown,113
Selena Gomez,165
Kali Uchis,150
Mac Miller,182
Miley Cyrus,127
Young Thug,191
Lil Durk,178
Gunna,143
Doja Cat,184
NF,110
Justin Bieber,177
Polo G,182
Harry Styles,143
Brent Faiyaz,117
Billie Eilish,181
Trippie Redd,102
Bruno Mars,142
Playboi Carti,129
Fuerza Regida,130
Don Toliver,93
Kodak Black,101
Beyoncé,147
Rauw Alejandro,177
Rod Wave,127
Junior H,158
JAY-Z,142
Khalid,140
A Boogie Wit da Hoodie,150
Ed Sheeran,125
Lil Peep,141
Steve Lacy,162
NLE Choppa,170
BTS,164
Eslabon Armado,151
DaBaby,163
The Wild Earth,87
Linkin Park,148
Ice Spice,162
A$AP Rocky,2
Feid,157
Shakira,161
PinkPantheress,147
Imagine Dragons,157
Deftones,136
Katy Perry,182
Adele,133
Dua Lipa,136
Megan Thee Stallion,168
Luis R Conriquez,87
Cardi B,179
50 Cent,154
Nirvana,131
Olivia Rodrigo,128
TV Girl,123
sped up nightcore,59
Big Sean,117
Carin Leon,61
Red Hot Chili Peppers,131
Russ,150
Logic,138
King Von,180
Elton John,133
Machine Gun Kelly,138
Queen,156
2Pac,152
Paramore,165
Pitbull,161
Michael Jackson,168
```

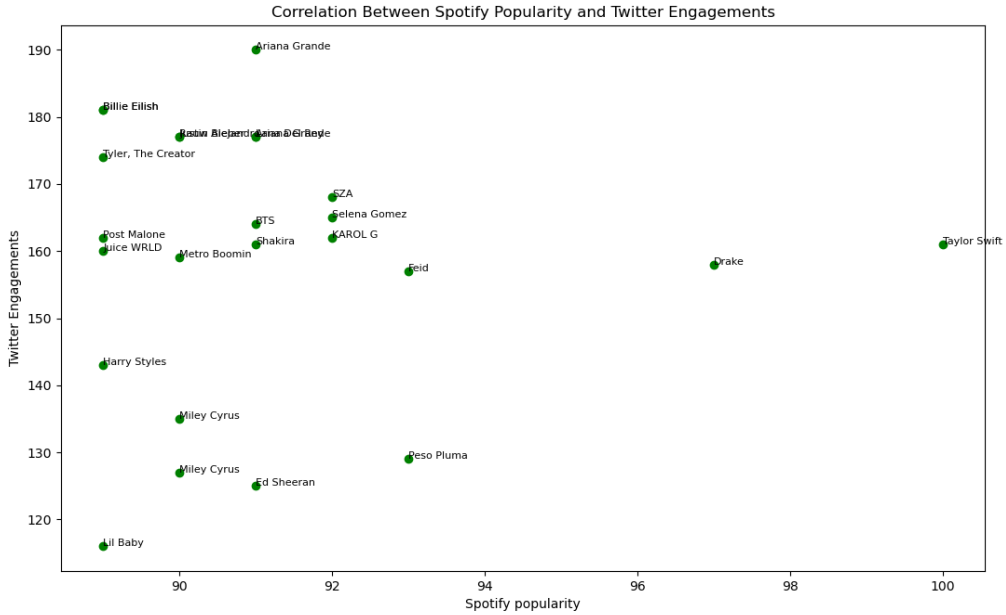
csv file

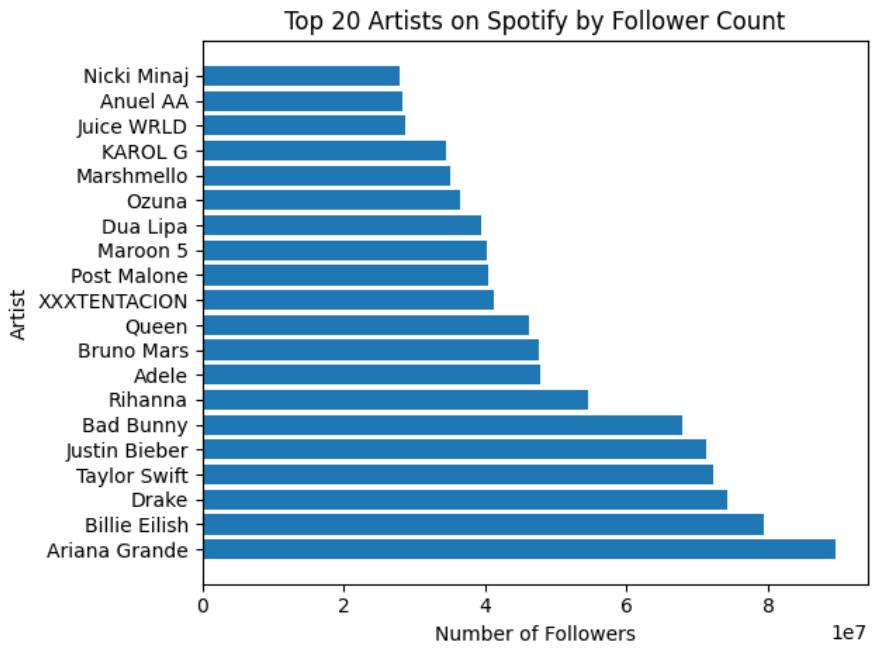
5. The visualization that you created (i.e. screen shot or image file) (10 points)






Engagements by Artist (Top 25)
Tweets
Retweets
Mentions
Number of Engagements
0
10
20
30
40
50
60
Young Thug
Ariana Grande
Doja Cat
Mac Miller
Polo G
Katy Perry
Billie Eilish
Billie Eilish
King Von
Melanie Martinez
Cardi B
Lil Durk
Ariana Grande
Lana Del Rey
Justin Bieber
Rauw Alejandro
Tyler, The Creator
Melanie Martinez
NLE Choppa
SZA
Megan Thee Stallion
Michael Jackson
Yeat
Selena Gomez
Paramore


Correlation Between Spotify Popularity and Twitter Engagements
Twitter Engagements
Spotify popularity
120
130
140
150
160
170
180
190
90
92
94
96
98
100
Ariana Grande
Billie Eilish
Tyler, The Creator
SZA
Selena Gomez
BTS
Shakira
KAROL G
Post Malone
Juice WRLD
Metro Boomin
Feid
Drake
Taylor Swift
Harry Styles
Miley Cyrus
Miley Cyrus
Peso Pluma
Ed Sheeran
Lil Baby



6. **Instructions for running your code (10 points)**

Spotify:

A. Install the Spotipy library by running pip install spotipy in your terminal or command prompt
B. Run the script spotify.py 4 times to collect the data for the top 100 most streamed artists in 2021. Database will be saved in the spotify.db database in two tables, top_artists and song_info

Twitter:

A. Install the required packages: tweepy and sqlite3.
B. Update the Twitter API credentials with your own credentials.
C. Ensure that the top_artists.db file exists in the same directory as the Python file.
D. Run the Python file in your IDE or from the command line
E. Run 4 times, as this will make 25 requests at a time. If time is out, wait 15 minutes to run again.
F. Saves id, name, popularity, tweets, retweets, mentions in a new file ‘artist_twt’

calc_total:

A. Make sure that you have SQLite and the "csv" module installed in your Python environment.
B. Run the script
C. The script will connect to the "artist_twt.db" SQLite database, retrieve all the rows from the "top_artists" table, calculate the total engagements for each artist, update the

"total_engagements" column in the database, and write the data to a CSV file called "total_engagements.csv" in the same directory as the Python script.

D. After the script has finished running, you should be able to find the "total_engagements.csv" file in the same directory as the Python script, which contains the artist names and their corresponding total engagements.

Top_engage_chart, streams_sp, engage_sp, tweets_vs_popularity

A. Make sure that artist_twt.db SQLite database is available before running in the command line

7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)

spotify.py

- Authenticate with Spotify using client ID and secret using the SpotifyClientCredentials class from Spotipy.
- Connect to an SQLite database using the sqlite3 module in Python.
- Create two tables in the database: "top_artists" and "song_info".
- If the number of rows in the "top_artists" table is less than 100, retrieve the top 25 most streamed artists of 2021 using the search() method from Spotipy and insert each artist into the "top_artists" table using the execute() method from the sqlite3 module.
- For each artist, retrieve their top track using the artist_top_tracks() method from Spotipy and insert the top track information into the "song_info" table using the execute() method from the sqlite3 module.
- Commit the changes to the database using the commit() method from the sqlite3 module and close the connection using the close() method from the sqlite3 module.
- Wait for five seconds before running the code again using the time.sleep() method from the time module.
- Select all rows from the "top_artists" table using the execute() method from the sqlite3 module and print the results using a for loop.
- Close the connection to the database using the close() method from the sqlite3 module.

twt.py

- Uses the Tweepy library to connect to the Twitter API using the API credentials provided. Connects to the spotify.db SQLite database using the sqlite3 library. Creates artist_twt.db
- Enters a loop that will continue until all artists in the database have been searched. Each iteration of the loop retrieves the next 25 distinct artist names from the database, starting from the last artist searched, and then loops through each artist.
- For each artist, the script constructs a query string using the artist name and searches for tweets containing the query string using the api.search_tweets method from the Tweepy library.

- If the search encounters a rate limit error, the script will pause for 15 minutes before trying again.
- Counts the number of tweets, retweets and mentions for the artist and updates the corresponding record in the artist_twt database using an SQL statement.
- Sleeps for 1 second to avoid hitting the Twitter API rate limits and moves on to the next artist.
- Script closes the database connection and starts the next iteration of the loop to retrieve the next batch of artists from the database.

calc_total.py
- Connects to the "artist_twt.db" database and selects all the rows
- Loops through each row, calculates the total engagements for the artist, and updates the "total_engagements" column of that row. Selects the artist name and their total engagements from the "top_artists" table and stores them in the "rows" variable.
- Writes the "rows" variable to a CSV file named "total_engagements.csv". The CSV file includes a header row that consists of the "name" and "total engagements" columns.

top_engage_chart.py
- Connects to the "artist_twt.db" SQLite database
- Retrieves data on the top 25 artists by total engagements.
- Creates a bar chart with three sets of bars representing the number of tweets, retweets, and mentions for each artist.
- X-axis tick locations and labels are set based on the artist names. The chart title and axis labels are added, and a legend is included.
- Chart is displayed using the plt.show() function.

streams_sp.py
- Creates a scatter plot to visualize the correlation between the number of Spotify streams and the total engagements (tweets, retweets, and mentions) on Twitter for the top 25 artists in the "artist_twt" table
- Connects to the database and selects the required data using an SQL query.
- Extracts the required data into separate lists (names, streams, and engagements)
- Creates a scatter plot using matplotlib
- Sets the x and y-axis labels and title, adds text labels for each point, and adjusts the y-axis tick increment.
- Shows the plot and closes the database connection.

engage_sp.py
- Connects to a SQLite database called "artist_twt.db"
- Selects the top 25 artists with the highest number of Twitter engagements.
- Extracts the name, Spotify streams, and Twitter engagements for each of these artists
- Creates a scatter plot of the data.
- X-axis represents the number of Twitter engagements Y-axis represents the number of Spotify streams.

- Plot is displayed and the database connection is closed.

top_artist_chart.py
- Connects to a SQLite database named 'spotify.db' and retrieves data from the 'top_artists' table.
- It extracts the names and follower counts of the top 20 artists based on follower count and creates two lists.
- These lists are then used to create a horizontal bar chart using Matplotlib, with artist names on the y-axis and follower counts on the x-axis.
- The chart is then customized with a title, axis labels, and layout. Finally, the chart is displayed using plt.show().

tweets_vs_popularity.py
- Connects to the spotify.db database then attaches the artist_twt.db database to i
- Queries the combined databases using JOIN and retrieves the results
- Uses Matplotlib to create a scatter plot of the results with Spotify Popularity on the x-axis and Twitter Tweet on the y-axis, with a title and axis labels.
- Create a scatter plot visualization of the relationship between Spotify Popularity and Twitter Tweet for each artist, with each data point representing an artist.

8. **You must also clearly document all resources you used. The documentation should be of the following form (20 points)**

Date	**Issue Description**	**Location of Resource**	**Result (did it solve the issue?)**
4/20	How to make a bar chart in matplotlib	https://datatofish.com/bar-chart-python-matplotlib/	Yes
4/19	Having trouble getting started with the spotipy API	https://medium.com/@maxtingle/getting-started-with-spotifys-api-spotipy-197c3dc6353b	Yes, the article provided sample code as well as detailed instructions and graphics to aid the process
4/17	What is the Twitter API limit?	https://developer.twitter.com/en/docs/twitter-api/rate-limits	Yes, it helped me better to figure out how many tweets to pull per artist, per run
4/20	Intro to matplotlib (types of graphs and customization)	https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/#	Yes, it was a good, efficient description of how to utilize

			matplotlib in different ways
4/19	Error Output: from spotipy.oauth2 import SpotifyClientCredentials ModuleNotFoundError: No module named 'spotipy.oauth2'; 'spotipy' is not a package	https://stackoverflow.com/questions/62196197/why-dont-find-spotipy-oauth2-module-in-spotipy	Yes, I realized that I named the scripts as 'spotipy', therefore python tries to look for 'oauth2' in the script. Error was solved after renaming the file