Project code:

GitHub Repo Link: https://github.com/luluzh77/507-finalproject
Install packages including matplotlib, flask, and requests.

To run the code, 1) install the required packages, including matplotlib, flask and requests.
2) boot up the flask app by running the command line: python app.py. 3) copy the link to the web browser and interact with the web app.

Data Sources:

Spotify data: I am using two API endpoints to get the data I needed for this project in Spotify:

- https://api.spotify.com/v1/me/following?type=artist

I can get all the artists related information from the response including name, popularity, artists profile image, etc.

- https://api.spotify.com/v1/artists/artist_internal_id/top-tracks?market=US

I can get all top tracks of a particular artist in the US. Normally, each artist has 10 or so top tracks. Each track would contain information including track name, popularity, album, etc.
I used Oauth 2.0 authentication method to get access the Spotify endpoint. Basically, it is needed to create a Spotify account and login the developer console to get client id and client secret in order to generate bearer token for authentication. Please note that the bearer token will be expired every 24 hours. If you need to replicate the data query process, you may need your own Spotify account to generate new bearer token for authentication. With the bearer token, I used Python "requests" package to send request to each API endpoint to get data and merged the results from each endpoints into the same file called "spotify.json."

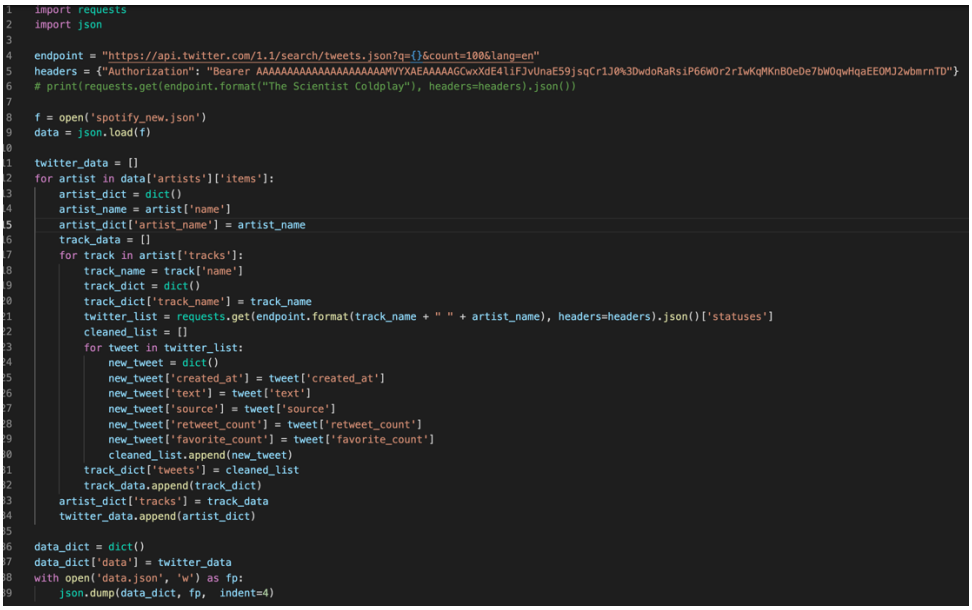
The number of records available on Spotify API are countless. I retrieved 17 artists and 10 top tracks of each artist, so the total number of records are 170. The important attributes are name (represents artist's name), popularity (represents artist's popularity) and url (represents the artist's image url).

Twitter data: For twitter data, I am using https://api.twitter.com/1.1/search/tweets.json?q=query_text&count=100&lang=en in Twitter standard API v1.1 to get the at most 100 tweets in English for each hot track of each artist from

Spotify data. The results would include the text content of the tweet, number of retweet, number of favorite, and etc. Same as Spotify API, I used Oauth 2.0 authentication method to get access the Twitter API. A twitter developer account is needed for accessing the standard v1.1 API. Twitter developer console would generate the bearer token for users. Then, I used Python "requests" package with the generate bearer token to get all twitter data needed for this project.

The number of records of the most popular tweets mentioned the artist or the track is 100 respectively, while the number of all the related tweets are numerous and each is different. The important attributes are artist_name, track_name, text (represents the tweet content), retweet_count and favorite_count.

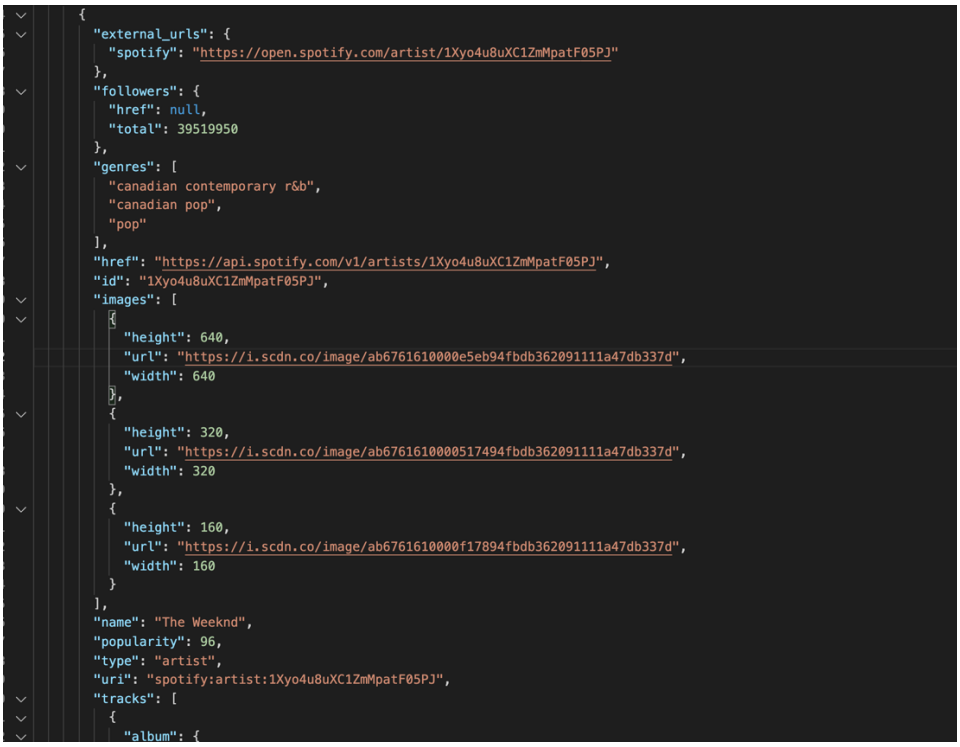
Below is the code I wrote to access the data from Spotify and Twitter APIs:



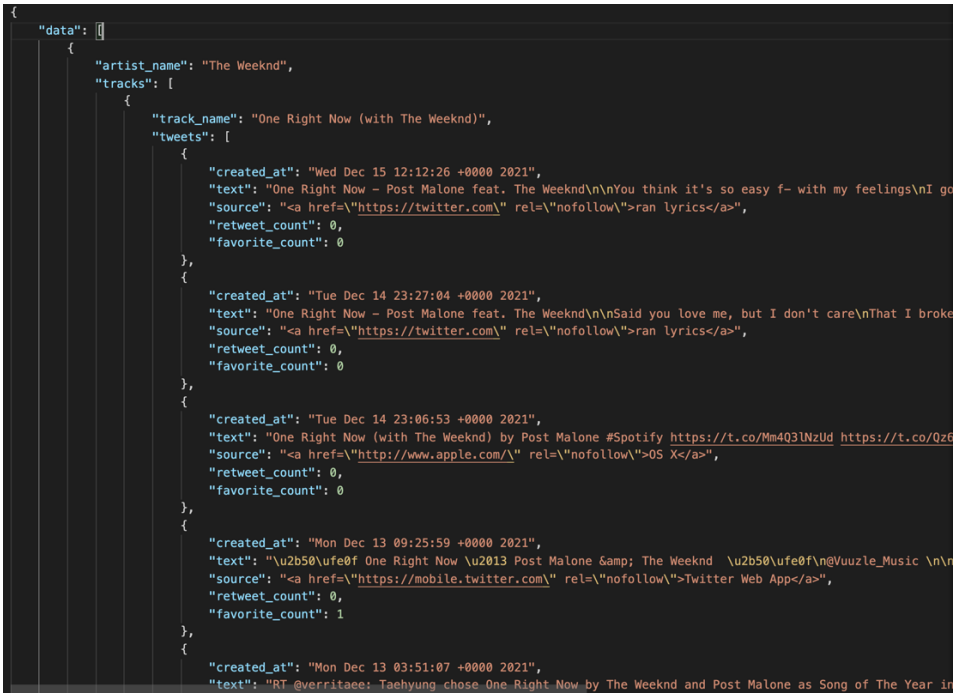
```python
import requests
import json

endpoint = "https://api.twitter.com/1.1/search/tweets.json?q={}&count=100&lang=en"
headers = {"Authorization": "Bearer AAAAAAAAAAAAAAAAAAAAAMVYXAEAAAAAGCwxXdE4liFJvUnaE59jsqCr1J0%3DwdoRaRsiP66WOr2rIwKqMKnBOeDe7bWOqwHqaEEOMJ2wbmrnTD"}
# print(requests.get(endpoint.format("The Scientist Coldplay"), headers=headers).json())

f = open('spotify_new.json')
data = json.load(f)

twitter_data = []
for artist in data['artists']['items']:
    artist_dict = dict()
    artist_name = artist['name']
    artist_dict['artist_name'] = artist_name
    track_data = []
    for track in artist['tracks']:
        track_name = track['name']
        track_dict = dict()
        track_dict['track_name'] = track_name
        twitter_list = requests.get(endpoint.format(track_name + " " + artist_name), headers=headers).json()['statuses']
        cleaned_list = []
        for tweet in twitter_list:
            new_tweet = dict()
            new_tweet['created_at'] = tweet['created_at']
            new_tweet['text'] = tweet['text']
            new_tweet['source'] = tweet['source']
            new_tweet['retweet_count'] = tweet['retweet_count']
            new_tweet['favorite_count'] = tweet['favorite_count']
            cleaned_list.append(new_tweet)
        track_dict['tweets'] = cleaned_list
        track_data.append(track_dict)
    artist_dict['tracks'] = track_data
    twitter_data.append(artist_dict)

data_dict = dict()
data_dict['data'] = twitter_data
with open('data.json', 'w') as fp:
    json.dump(data_dict, fp,  indent=4)
```

Data Structure:

Spotify data:



Twitter data:



Interaction and Presentation Options:

On my web application, users will see the 17 artists I'm following on Spotify. By hovering on each artist's photo, users can see their names. Scrolling down the artists page users will see two

bar graphs that demonstrates the popularity of each artist on Twitter and Spotify. Users can also click on any artist to see their top tracks on a new page where the popularity of each track on the two platforms is displayed by two bar graphs below the page. Users can view in-depth information about the popularity of each track by clicking on the track thumbnail and they will be directed to a new page where a table would appear that demonstrates the content of the most popular tweets, the retweet count and favorites count of each popular tweet. Below the table is a scatter plot that visualize the table information.

I use Flask to build my web application. The graphs shown on the web are made by matplotlib library.