SPOTIFY

Is this artist appropriate for your children?

Grace Badagliacca

GOALS/GOALS ACHIEVED

My initial goal for my API final project was to utilize the Spotify API to take in user input and return related tracks. To do this, I wanted the program to prompt a user to enter a keyword which would return a list of tracks that were relevant to this word (i.e. dance, study, etc.). Unfortunately, this was difficult due to the large amount of tracks available for access, along with the issue of matching tracks with keywords, which cannot efficiently be done with the public Spotify developer API.

I decided to update my project after working with the previous idea, but was determined to maintain the user input aspect of it. Admittedly, I wanted my project to be something acceptable for class, and cool enough to demonstrate with my friends (who don't really believe that I am surviving a coding course). I altered my project to take in an artist name, return the artist's number of followers, genre, and popularity score. I also wanted the program to return some useful data about the individual tracks, and decided that the "explicit" or "not explicit" rating of a song would be interesting.

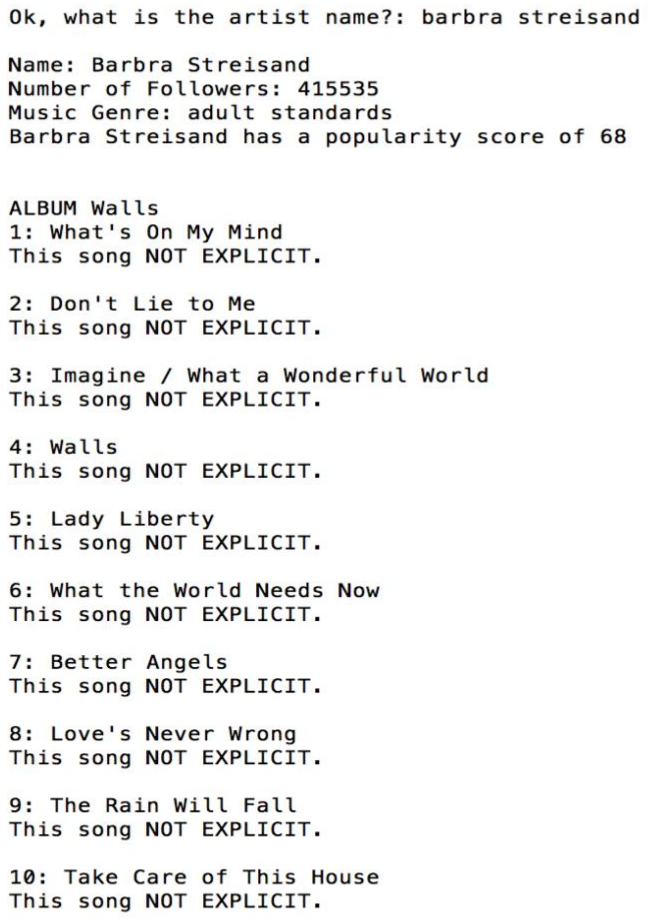
PROBLEMS

The biggest difficulty that I faced throughout this project was understanding the authentication process with Spotify. Because Spotify is a public website, but also a private desktop browser for subscribed users, the authentication process can vary based on the user. In order to understand the authentication process, I utilized resources from YouTube and GitHub, which helped to implement the 'Spotipy' documentation. I discovered that the most effective way to authenticate my access to the developer API was to export the Client ID, Client Secret, and Redirect URI. The Redirect URI almost acts as a deeper level of authentication in proving that you received access and were redirected to the given site (in this case: 'https://google.com')

My second most difficult problem was implementing SQL commands that created new tables for each user-input, rather than one large table with several different artists. I wanted the program to reiterate for each input so the visualization would also represent the songs of one artist. I fixed this problem by placing the commands within a specific for loop at a proper indentation to reiterate each time, but continue to input each track of one artist into the same database. I also wanted the SQL tables to iterate with the user input as a parameter for the title of the table. By inputting the 'artistName' parameter into the SQL command (as well as the matplotlib visualization), the output table and visualization are unique for each user's input. Any artist available on Spotify can be input and analyzed with this program.

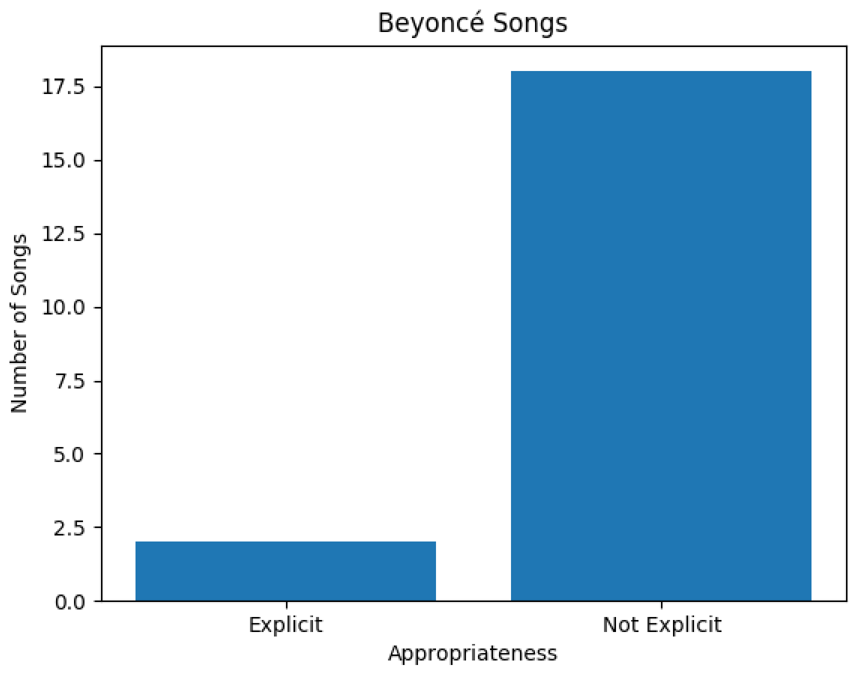
SOCIAL MEDIA REPORT

The social media "report" can be seen when a user inputs an artist name and data is returned in the form of terminal output. The output is clear and easy to parse, beginning with the artist's name, number of followers, popularity score, and genre. Next are the track results, beginning with the name of the album, followed by the songs that are in that specific album. This will run results for all of the artists' albums and songs (in numbered order). Each track will also have text below it specifying whether or not the track is explicit.



```
Ok, what is the artist name?: barbra streisand

Name: Barbra Streisand
Number of Followers: 415535
Music Genre: adult standards
Barbra Streisand has a popularity score of 68


ALBUM Walls
1: What's On My Mind
This song NOT EXPLICIT.

2: Don't Lie to Me
This song NOT EXPLICIT.

3: Imagine / What a Wonderful World
This song NOT EXPLICIT.

4: Walls
This song NOT EXPLICIT.

5: Lady Liberty
This song NOT EXPLICIT.

6: What the World Needs Now
This song NOT EXPLICIT.

7: Better Angels
This song NOT EXPLICIT.

8: Love's Never Wrong
This song NOT EXPLICIT.

9: The Rain Will Fall
This song NOT EXPLICIT.

10: Take Care of This House
This song NOT EXPLICIT.
```

The results are then processed into an SQLite database, creating a new table for each new artist input. The results of explicit/not explicit are then processed with matplotlib to create a bar chart displaying the amount of Not Explicit/Explicit songs in an easy to compare side-by-side plot. The bar chart is also recreated each time a new input is processed, with the specified artist name as the parameter for the title. (See below)



INSTRUCTIONS

The Spotify developer API requires authentication through login and redirect URI. In order to run the program, the user must first export their Client ID and Client Secret, as well as a Redirect URI into the terminal. The format for doing so is like such:

```
export SPOTIPY_CLIENT_ID='your-spotify-client-id'
export SPOTIPY_CLIENT_SECRET='your-spotify-client-secret'
export SPOTIPY_REDIRECT_URI='your-app-redirect-url'
```

My personal Client ID and Client Secret are provided within the documentation of my program, which are required for the program to function properly. Once these have been imported, the program can be run using a basic "python" or "python3" command in terminal, followed by the user's Spotify username.

To find one's Spotify username, locate the circle with three white dots within a Spotify profile. Once the user chooses "Copy Spotify URI", the username is the numbers or letters that follow "user:". (spotify:user:gottabegracieb) My request in the terminal looks as follows:

```
python3 SpotifyFP.py gottabegracieb
```

The user will then be redirected to the Spotify authentication page, click "Okay" and then be directed to a basic Google homepage. This link will be copy and pasted to the terminal that will be prompting the following:

```
Enter the URL you were directed to:
```

The user will then be asked to make a choice of ‘0’ or ‘x’ (0: Enter an artist name; x: exit). If ‘0’ is chosen, the user can enter any artist available on Spotify to return the statistics of their tracks and explicit/not explicit values.

DOCUMENTATION

(included as comments in SpotifyFP.py)

RESOURCES

DATE	ISSUE	LOCATION	RESULT
12/02	Needed to understand the ‘spotipy’ package to parse the Spotify API and find data	Spotipy Read the Docs	Learned how to implement ‘spotipy’ package to find specific indexes of the API for data collection
12/04	Could not determine the proper way to authenticate access to Spotify	YouTube Tutorial	Learned the requirement of exporting Client ID, Client Secret, and Redirect URI into terminal
12/06	SQL database needed to input data into new table for each new user input	Office hours with Sonal!!	Placed SQL commands within a for loop in the correct location to reiterate for each new artist

** I received lots of help from the lecture slides to implement the data visualizations as well