Group name: AJ Group

Group Members: Andrew Nakamura, Jess Yang

GitHub Repository Link: <https://github.com/andyjknakamura/finalproject/tree/master>

SI 206

Final Project Report

1. The goals for your project (10 points):

Music enjoyment is inherently subjective in nature, and as such the measurement of how “good” or “successful” an artist’s discography is valued is ill-defined. However, various streaming platforms and charts attempt to catalog the popularity of various artists to capture whose music is the most widely appreciated by a global audience of music listeners. Yet, music charts are not all determined by the same metrics, and we aimed to illustrate how the same set of artists can be given widely different places on different charts. We also wanted to see how many genres popular artists use.

2. The goals that were achieved (10 points):

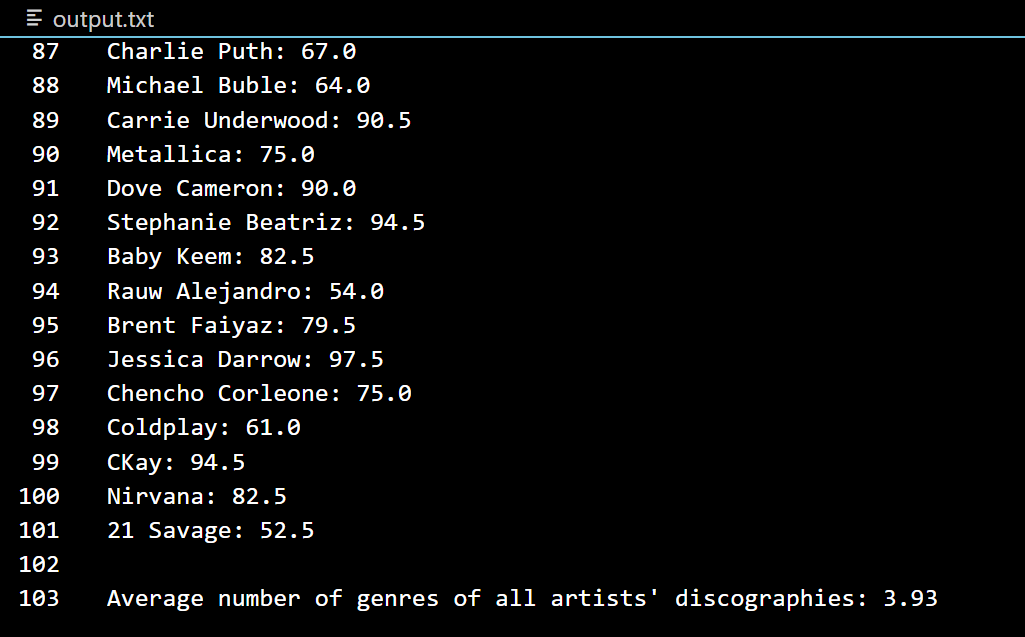
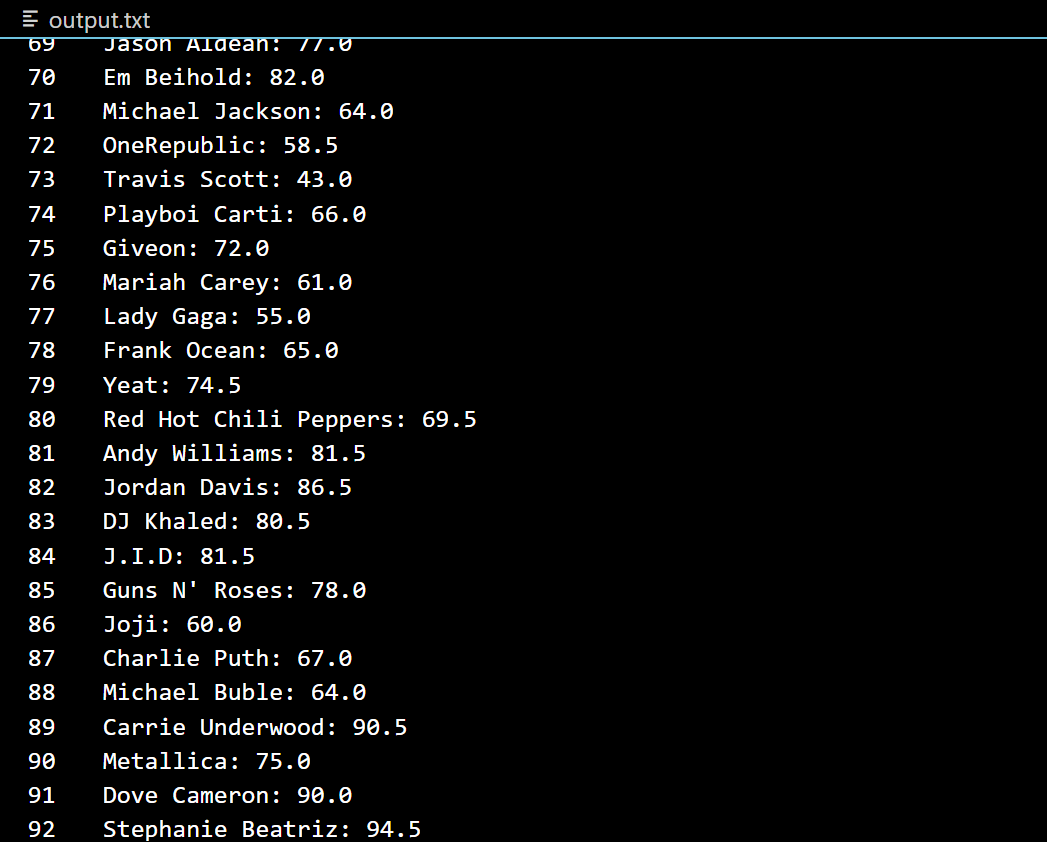
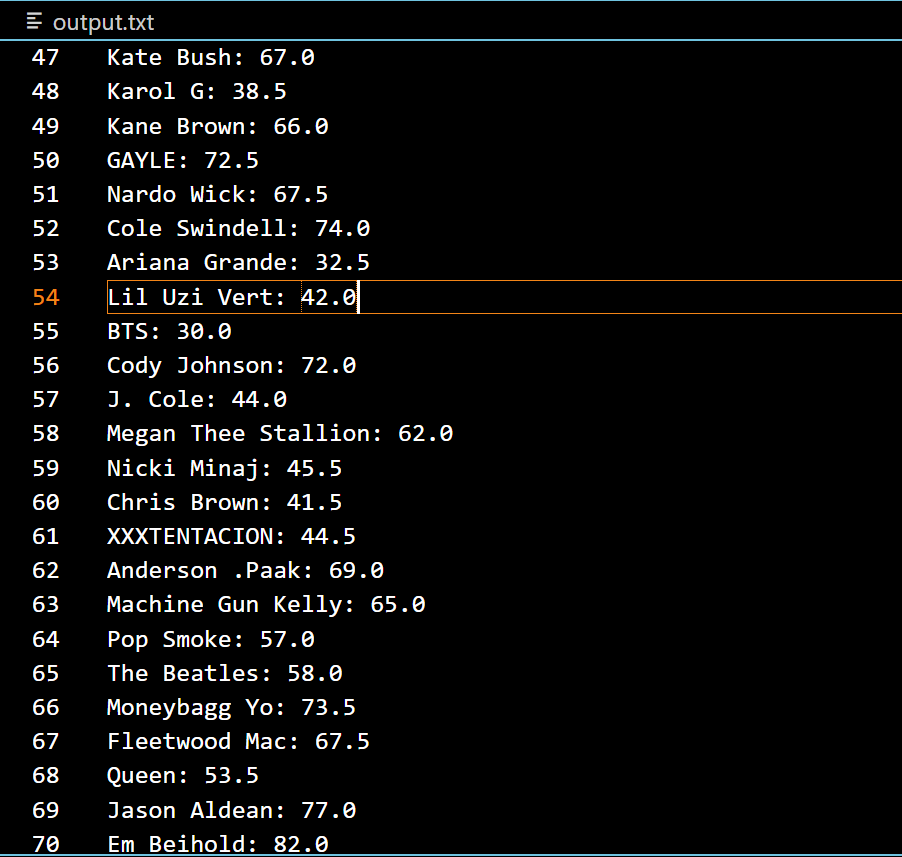
We compared Billboard’s Top 100 Artists of 2022 rankings to each artist’s popularity value stored in Spotify’s API, and created a graph that plotted three points for each artist based on their position in the Billboard Top 100, their position on spotify’s popularity ranks relative to the other artists on the Billboard chart, and the average position of each artist between Billboard’s and Spotify’s charts. We also created a pie chart that illustrates the percentages of different numbers of genres that make up all 100 artists’ discographies, showing how many genres popular artists usually have.

3. The problems that you faced (10 points)

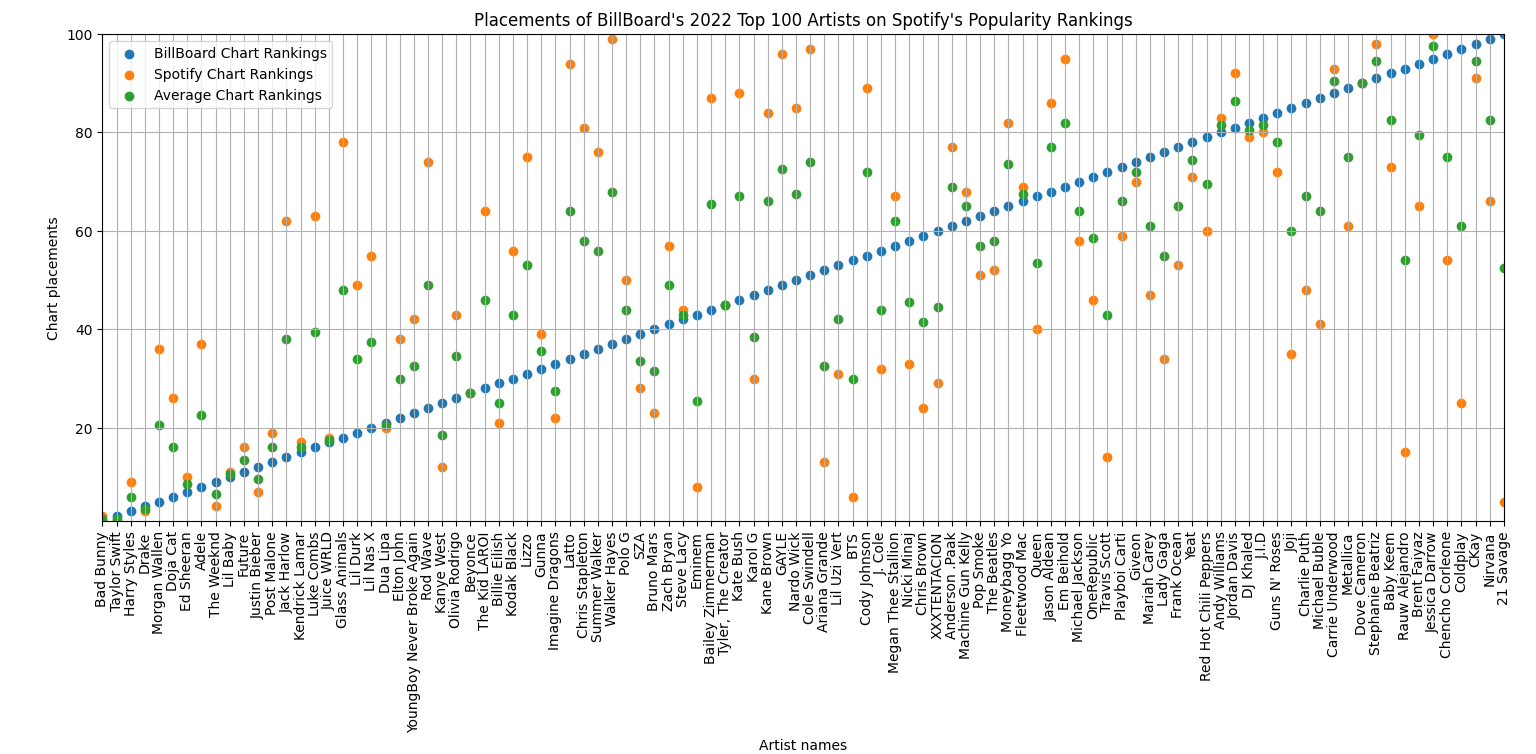
Spotify API does not allow users to get artists’ info by entering their names. Instead, users have to enter their IDs to obtain the data. Since we need to get Billboard’s Top 100 artists’ popularity on Spotify, we had to search their IDs using their names one by one on Spotify, which was time-consuming. Also, some artists do not have info about genres. We found this problem and added use a if statement to fix this problem.

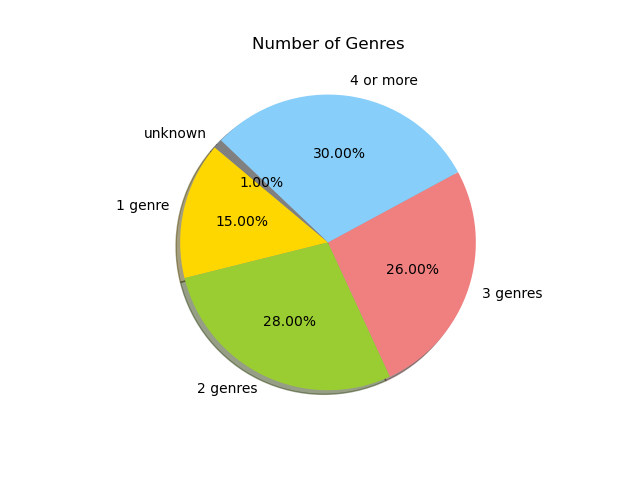
4. Your file that contains the calculations from the data in the database (10 points)

**Content of ‘output.txt’, the file we wrote our calculations to:**



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





6. Instructions for running your code (10 points)

1. Unzip the file to whichever file folder on your computer you would have the easiest time accessing.
2. In the event that you must create new tables in a new database other than the one attached, replace ‘project.db’ in API.py with your database name and run the entire python file in your terminal. This will also write calculations to a text file and generate our graphs
3. The data visualizations only open one at a time, so when you are done viewing or saving the first graph, close the graph’s window to allow the second one to appear

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

**Original sheet**: <https://docs.google.com/spreadsheets/d/1s7bJIGTl7Ghxjzs61jXsbHXg61UAUrSTC8Hl89DqYAE/edit?usp=sharing>

* authentication()
* get\_artist\_info(access\_token, ID)
* open\_database(db\_name)
* spotify\_table1(artist\_list, cur, conn)
* spotify\_table2(artist\_list, cur, conn)
* billboard\_table(link, cur, conn)
* get\_genre\_average(cur, conn)
* get\_rank\_averages(cur, conn)
* write\_averages(avg\_genre\_num, diff\_dict, file)
* graph\_ranks(cur, conn)
* genre\_percentage(cur, conn)

| **Function** | **Description** | **Input** | **Output** |
| --- | --- | --- | --- |
| authentication() : | To get the access token to Spotify API | None | str: access\_token |
| get\_artist\_info(access\_token,  ID) | To get certain artist's name, genres and popularity info on Spotify by entering their ID. | access\_token: string ; ID: str | tuple: (ID, name, genres, popularity) |
| open\_database(db\_name) | To open the database named db\_name and return cur, conn. | db\_name: string, database name. | cur, conn |
| spotify\_table1(artist\_list, cur, conn) | To create a table of genres of each artist. The table contains 3 columns: id(the order of artists), genres and the number of genres. | artist\_list: a list of artists' information tuples including genres and the number of genres;conn: connect; cur: cursor | None |
| spotify\_table2(artist\_list,  cur, conn) | To create a table of artists' popularity. The table includes id (the order of artists), popularity, popularity ranking | artist\_list: a list of artists' information tuples including genres, the number of genres & popularity; conn: connect; cur: cursor | None |
| billboard\_table(link, cur,  conn) | To create a table of artists' billboard ranking. The table includes id, name and rank. | link: the link of billboard website; conn: connect; cur: cursor | None |
| get\_genre\_average(cur, conn) | To calculate the average number of genres 100 popular artists have | cur, conn | the average number of genres |
| get\_rank\_averages(cur, conn) | To calculate the average of billboard rank and spotify rank for each artist | cur, conn | diff\_dict: dict, keys: each artist's name; values: each artists' average rank |
| write\_averages(avg\_genre\_num,  diff\_dict, file) | To write the genre and the rank average data into a file | avg\_genre\_num: the average number of genres; diff\_dict: the dict of average ranks; file: string, a filename. | None |
| graph\_ranks(cur, conn) | To plot average ranking data as a scatterpoint graph to show the potential relationship between two rankings. | cur, conn | None |
| genre\_percentage(cur, conn) | To plot the number of genres as a pie chart to show the distribution of genre numbers. | cur, conn | None |

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points): (<https://piazza.com/class/l754aa3ib8z43s/post/493>)

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue Description** | **Location of Resource** | **Result(did it solve the issue?)** |
| Dec 7 | Obtaining data from Spotify API | Spotify API: https://developer.spotify.com/documentation/web-api/ | Yes |
| Dec 7 | Obtaining data from Billboard Top 100 Artists of 2022 | https://www.billboard.com/charts/year-end/top-artists/ | Yes |
| Dec 8 | Could not get access to Spotify API without a token. Checked a tutorial. | https://stmorse.github.io/journal/spotify-api.html | Yes |
| Dec 10 | Needed to find Spotify URIs for each artist to get their information from the API | Spotify Desktop Client | Yes |
| Dec 11 | Checked how to create 2 graphs using matplotlib | Lecture 21 Slides | Yes |
| Dec 11 | Wanted to rotate x-axis tick labels and reduce the margin sizes on the scatterplot | https://matplotlib.org/3.4.3/gallery/ticks\_and\_spines/ticklabels\_rotation.html | Yes |
| Dec 11 | Needed to resolve a warning regarding the rotation of the x-axis labels on the rank graph | https://stackoverflow.com/questions/63723514/userwarning-fixedformatter-should-only-be-used-together-with-fixedlocator | Yes |