Samantha Hicks and Andie Carroll

Professor Barbara Ericson

SI 206

Github repo: https://github.com/sjhicks/final-project.git

Original and Achieved Goals:

The original goal of our project was to take 5-10 artists and rate their popularity based off of their most popular songs through the Spotify API. We then wanted to compare these artists and popularity to their rating on their music videos through the iTunes API to see if it was comparable. When we started the project we realized that we wanted to change our end goal and scope of the project, but wanted to continue to work with data concerning music. We decided to utilize Beautifulsoup to gather data from Spotify's website of the Spotify top 200 charts. From here, we found how many times an artist appeared on the charts, what their total amount of streams were on the charts, and we calculated their average amount of streams. We then realized we needed to use an API apart from iTunes and switched to the streaming service Deezer. With this API we were able to access the Top 100 playlist on their platform and compare it with the Spotify website. Our new goal was to find a correlation between the streaming service of Spotify and Deever. We achieved our goals of creating bar graphs of average streams and average rankings and ended with a combined scatterplot of our data on both platforms in the end. We were also able to find the times that the artists appeared in the charts and create a visualization for that as well.

Problems We Faced:

At first we planned on using the iTunes API to create a database and look at the top charts on iTunes compared to Spotify. However, we soon found that this API was mainly for searching basic information about songs, podcasts, among other types of media, and it actually didn't show charts or rankings of any kind. We then had to find another API that would give us the type of data we wanted to use. The one we chose was this streaming service called Deezer. Although we were not familiar with this online music streaming service as it is actually from France, it was one of the few music APIs that was free and easy to use. After switching to this API it was easy to find and gather the data in our original goals.

We also ran into an issue when we tried to pull data from the spotify website. The first day the code was running smoothly, but abruptly stopped working due to hitting a quota or being blocked. We went to office hours to solve this issue and found a specific solution and line of code that helped overcome our problem.

Since we changed the scope of our project, we had to change what our end goal was. It took us some time to figure out what we wanted to calculate and find through our data, but we felt that finding a correlation between spotify streams and deezer ranks would make the most sense. Combining this data from two tables though was difficult for us and required us to make a total of 5 tables. Even though it took longer than expected, we were able to create a common key so that we could properly compare the two streaming services.

Functions:

File containing our calculations for the spotify.py:

```
We found the amount of times an artist appeared on the top 200 charts and their total
amount of streams on the top 200 charts. Using those two data points, we found each
artist's average amount of streams on the top 200 charts and organized it by the average
streams.
Doja Cat has an average streams of 1893526.5
Masked Wolf has an average streams of 1730062.0
Giveon has an average streams of 1679914.0
Bruno Mars, Anderson .Paak, Silk Sonic has an average streams of 1572820.0
Lil Nas X has an average streams of 1499937.0
Olivia Rodrigo has an average streams of 1386640.0
Glass Animals has an average streams of 1338614.0
Kali Uchis has an average streams of 1314746.0
Lil Tjay, 6LACK has an average streams of 1314680.0
SZA has an average streams of 1255208.0
Saweetie has an average streams of 1205684.0
Polo G has an average streams of 1103960.0
Bad Bunny, Jhay Cortez has an average streams of 1068040.0
Pooh Shiesty has an average streams of 1038682.0
Maroon 5 has an average streams of 941600.0
Cardi B has an average streams of 915549.0
Dua Lipa has an average streams of 907108.3333333334
AURORA has an average streams of 885996.0
Machine Gun Kelly has an average streams of 865744.0
The Weeknd has an average streams of 816205.25
Travis Scott, HVME has an average streams of 788900.0
Internet Money has an average streams of 737542.0
42 Dugg has an average streams of 727604.0
Lil Tjay has an average streams of 725725.5
Shotgun Shane, Young Buck, Bubba Sparxxx has an average streams of 724388.0
Kevin Gates has an average streams of 720714.0
Los Legendarios, Wisin, Jhay Cortez has an average streams of 701238.0
Beach Bunny has an average streams of 685414.0
Young Stoner Life, Young Thug, Gunna has an average streams of 671117.6666666666
```

File containing our calculations for deezer:

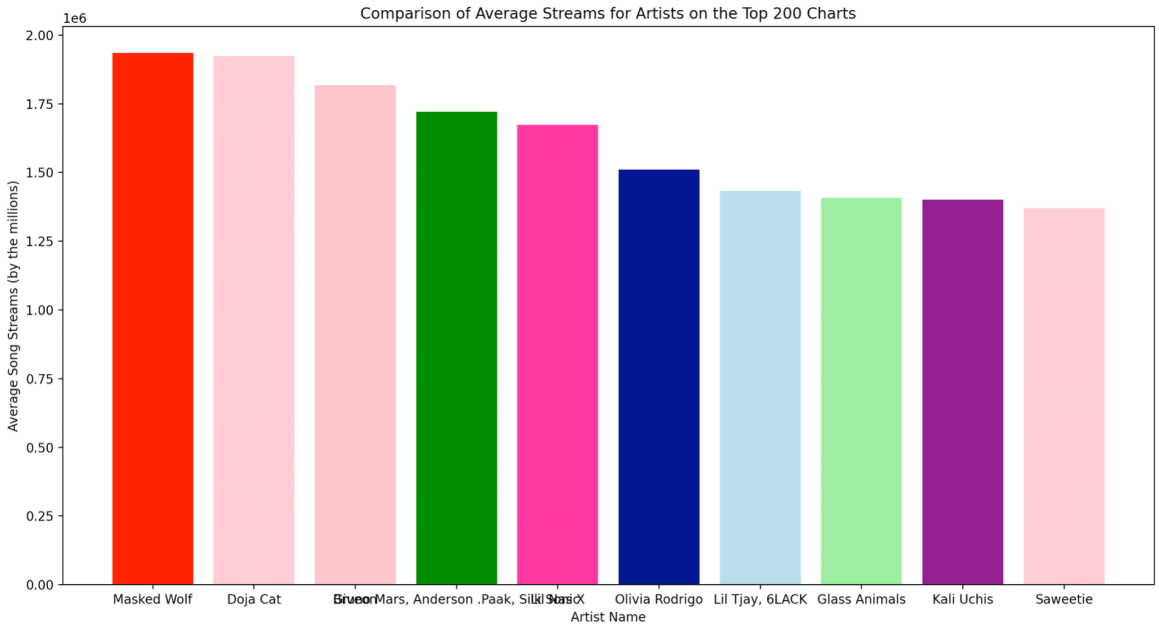
deezerfile.txt — Edited

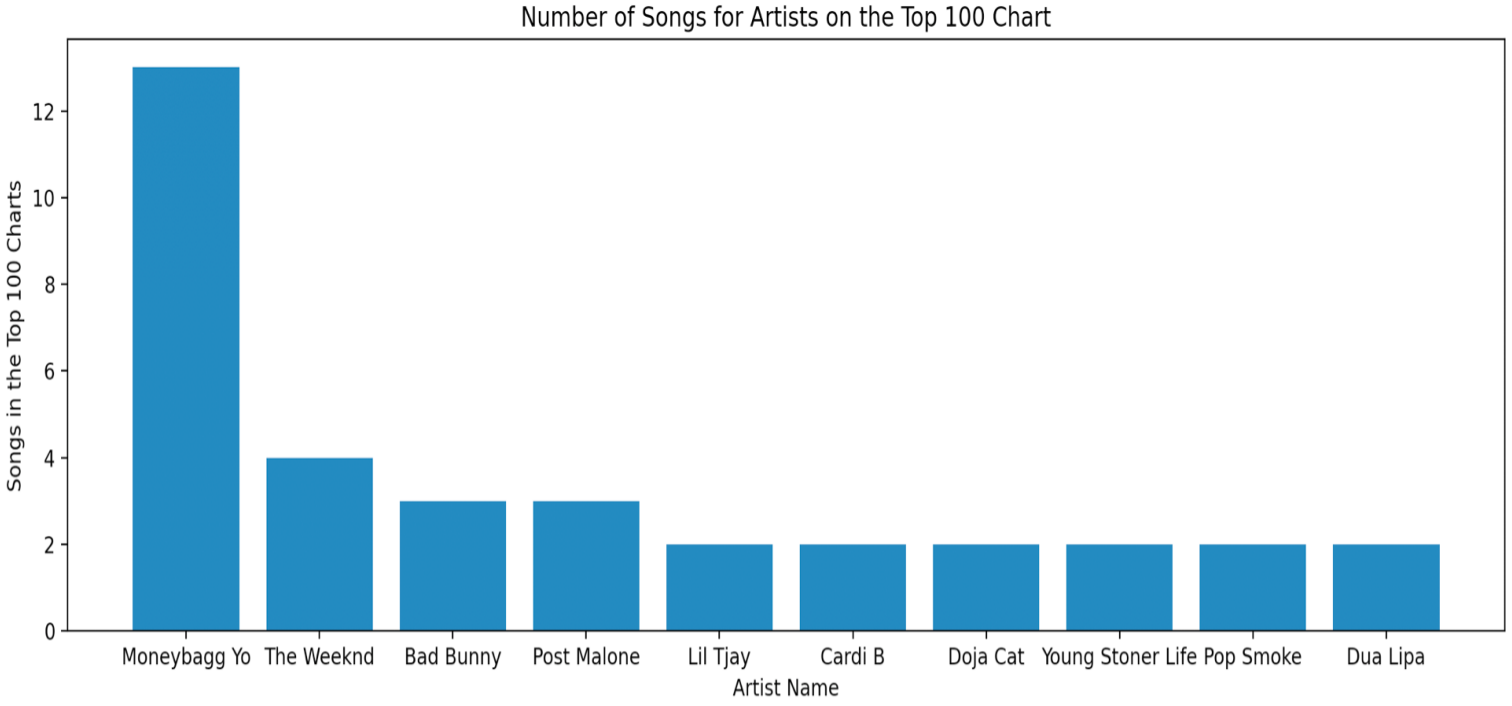
```
We found the average rank of each artist's songs in the top 100 for the US. Each rank is out of a million, with those being closest to a
million ranked the highest based on total number of streams.
Gera MX has an average ranking of 999858.0 out of a million based on songs in the top 100 charts
Tones and I has an average ranking of 998443.0 out of a million based on songs in the top 100 charts
Jawsh 685 has an average ranking of 997408.0 out of a million based on songs in the top 100 charts
Lewis Capaldi has an average ranking of 997181.0 out of a million based on songs in the top 100 charts
Billie Eilish has an average ranking of 995391.0 out of a million based on songs in the top 100 charts
Harry Styles has an average ranking of 995196.0 out of a million based on songs in the top 100 charts
Bruno Mars has an average ranking of 995120.0 out of a million based on songs in the top 100 charts
Topic has an average ranking of 994963.0 out of a million based on songs in the top 100 charts
Doja Cat has an average ranking of 993578.0 out of a million based on songs in the top 100 charts
KAROL G has an average ranking of 993458.0 out of a million based on songs in the top 100 charts
Justin Bieber has an average ranking of 992645.0 out of a million based on songs in the top 100 charts
Lil Nas X has an average ranking of 991146.0 out of a million based on songs in the top 100 charts
Masked Wolf has an average ranking of 990613.0 out of a million based on songs in the top 100 charts
Dua Lipa has an average ranking of 990234.5 out of a million based on songs in the top 100 charts
Cardi B has an average ranking of 987868.5 out of a million based on songs in the top 100 charts
Eminem has an average ranking of 985755.0 out of a million based on songs in the top 100 charts
Bad Bunny has an average ranking of 984886.6666666666 out of a million based on songs in the top 100 charts
The Weeknd has an average ranking of 984801.25 out of a million based on songs in the top 100 charts
Los Legendarios has an average ranking of 983716.0 out of a million based on songs in the top 100 charts
24kgoldn has an average ranking of 982179.0 out of a million based on songs in the top 100 charts
Olivia Rodrigo has an average ranking of 982163.5 out of a million based on songs in the top 100 charts
Regard has an average ranking of 980164.0 out of a million based on songs in the top 100 charts
Sebastian Yatra has an average ranking of 977759.0 out of a million based on songs in the top 100 charts
Giveon has an average ranking of 975375.0 out of a million based on songs in the top 100 charts
POLO G has an average ranking of 975084.0 out of a million based on songs in the top 100 charts
J Balvin has an average ranking of 973895.0 out of a million based on songs in the top 100 charts
R.E.M. has an average ranking of 973774.0 out of a million based on songs in the top 100 charts
DaBaby has an average ranking of 971835.0 out of a million based on songs in the top 100 charts
Kali Uchis has an average ranking of 971604.0 out of a million based on songs in the top 100 charts
Sech has an average ranking of 971525.0 out of a million based on songs in the top 100 charts
AC/DC has an average ranking of 968113.0 out of a million based on songs in the top 100 charts
```

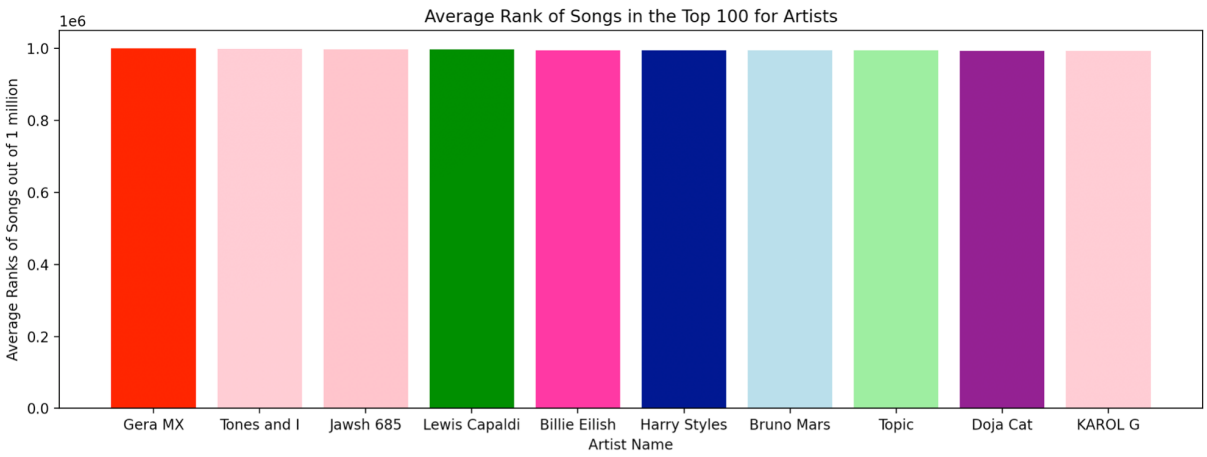
Database Joined Table:

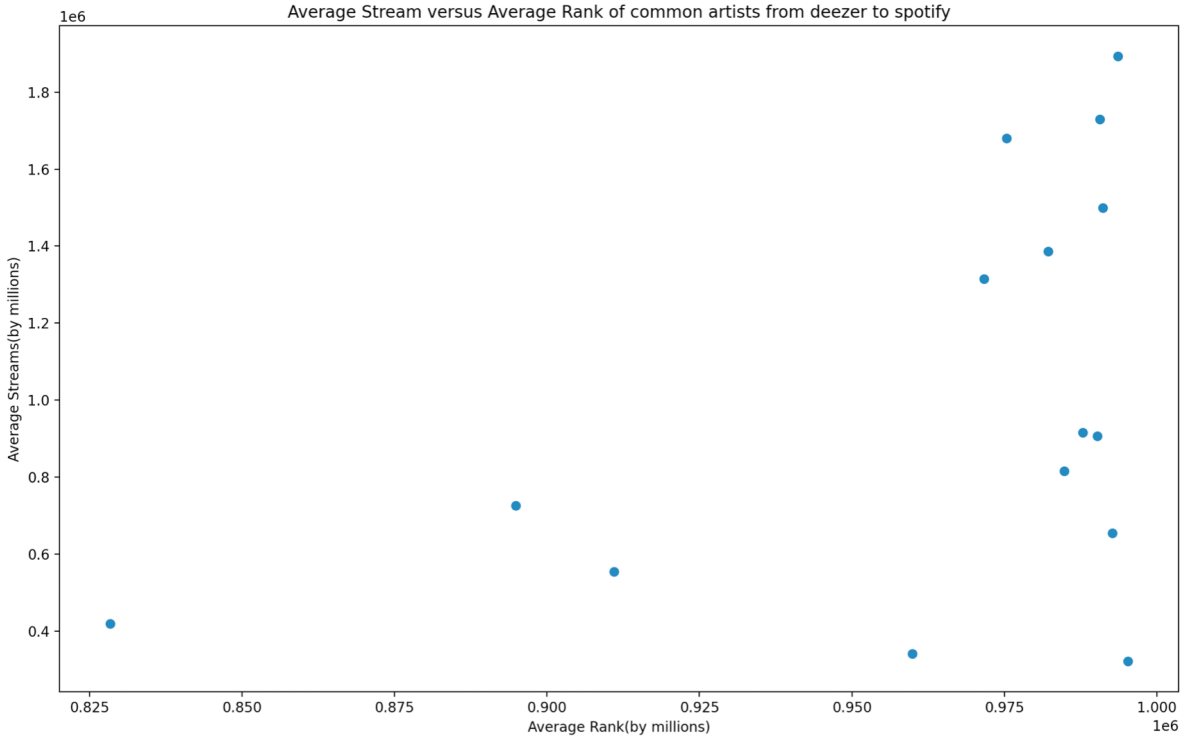
	Artist	average_streams	average_rank
	Filter	Filter	Filter
1	Doja Cat	1893526.5	993578
2	Masked Wolf	1730062	990613
3	Giveon	1679914	975375
4	Lil Nas X	1499937	991146
5	Olivia Rodrigo	1386640	982163.5
6	Kali Uchis	1314746	971604
7	Cardi B	915549	987868.5
8	Dua Lipa	907108.333333333	990234.5
9	The Weeknd	816205.25	984801.25
10	Lil Tjay	725725.5	894901

Visualizations






Average Rank of Songs in the Top 100 for Artists
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Average Ranks of Songs out of 1 million
Gera MX
Tones and I
Jawsh 685
Lewis Capaldi
Billie Eilish
Harry Styles
Bruno Mars
Topic
Doja Cat
KAROL G
Artist Name


Average Stream versus Average Rank of common artists from deezer to spotify
1e6
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
Average Streams(by millions)
0.825
0.850
0.875
0.900
0.925
0.950
0.975
1.000
1e6
Average Rank(by millions)

Instructions to run code

1. Locate the folder “final-project”. This folder contains all the files necessary to run the code, such as spotify_infol.py and deezer.py
2. First run deezer.py and it will create two visualizations
3. Then run spotify_final.py to create the visualizations using the data in the database
    a. Spotify_info.py makes a barchart of the average amount of streams on Spotify’s top 200 charts for the top ten most average streamed artists.
    b. Spotify_infol.py also makes a scatter plot of the average streams versus the average rank for artists that are in common for both the Spotify charts and Deezer Charts.
    c. Deezer.py makes a barchart of the average ranks on Deezer’s top 100 charts for the top ten most average ranked artists.
    d. Deezer.py also makes a barchart of how many times an artist appears on the top 100 charts.

Documentation

Below is a list of our three files that contain code for our project. Each file is broken down into its respective functions. We will go through each function and explain their input, functionality, and output, if applicable.

Name of file: Spotify_final.py

This file finds information on artists that are on Spotify’s top 200 charts. This information includes the artist’s name, total amount of streams, and how often they are on the charts.

- **Def find_artist_info**: this function utilizes beautiful soup to find information of the artists on Spotify’s top 200 list. It finds the amount of streams an artist has on a track and then how many times they are on the charts. We used dictionary accumulation to find the total amount of streams across the charts an artist has because they could appear more than once.
- **Def get_averages(lst)**: this function takes a list that contains how many total streams an artist has and the amount of times they appear on the charts. It then finds the average amount of streams each artist has (by dividing their total streams by their amount of appearances on the charts) and returns a sorted list of artists by the amount of average streams they have.
- **Def setUpDatabase(db_name):** this function creates a database on SQLite.

- Def setUpSongTable(info, average, cur, conn): in this function we pass in the tuple lists returned from each of the functions above. We do this so we are able to enter it into a database. We iterate through each tuple and grab the artist, total streams, amount of occurrences on the chart, and their average amount of streams. Then we set up the table.
- **Def setUpComparison(cur, conn):** this function utilizes the JOIN function to create a new table with datapoints from both average_streams and average_ranks which are two tables we created. We wanted to compare artists that were on both the Spotify charts and Deezer charts and see how their ranks and streamed correlated.
- **Def barchart_averages():** this function grabs all the top 10 most streamed artists based on their average amount of streams on the top 200 charts. It goes through each row to find the artist's name and then their average streams. Once we have the average streams, we create a bar graph. This bar graph has the name of each artist on the x-axis and the average streams on the y axis. It returns a list of tuples with the name of the artist and the average streams for each. This list of tuples is used to create the calculation file.

Name of file: Deezer.py

- **Def setUpSongsTable(tracks, cur, conn):** This function creates a table using the Deezer API to select the top 100 songs in the US from a playlist. This table inclue the songs in ranking with the song title as the primary key, the name of the artist who made the song, the album, the song is in, the duration of the song in seconds, and the ranking of the song out of a million based on streams.
- **Def findArtistTopSongsCount(cur, conn):** This function selects all the names of the artists from the database and results in a list. It then iterates through the list of artists to find how many times their songs are in the top 100 to creat a dictionary with the key being the name of the artist and the value being the count of songs. After that the dictionary is sorted by highest count to lowest and the result of the function is a list of tuples (artist_name, count_songs).
- **Def createBarGraph(tuple_list):** This function takes the list of tuples from the **findArtistTopSongsCount(cur, conn)** function and splits the tuples up into lists of artists and song counts. It then takes these lists to create a bar graph that demonstrates the top 10 artists with the most songs on the top 100 charts and the amount of songs they have on the charts.
- **Def findAverageRank(cur, conn):** This function finds the average rank of the songs each artist has in the top 100 charts. After selecting artists and ranks from the database, I created a dictionary where each artist was the key to a list of ranks that each of their songs in the top 100 had. After creating this dictionary, I iterated through each value of the dict to find the average of the ranks each artist has and create a sorted list of tuples from highest rankings to lowest.
- **Def createBarGraph2(tuple_list):** This function takes the list of tuples from the **findAverageRank(cur, conn)** function and splits the tuples up into lists of artists and

average rankings. It then takes these lists to create a bar graph that demonstrates the rankings the top 10 artists had out of a million.

Resources:

Date	Issue Description	Location of Resource	Result
4/21/21	Spotify blocked me from using their data	Office Hours	AJ helped me find a line of code that could be implemented in my file to allow me to use data from Spotify
4/22/21	I was having trouble creating a dictionary with the value being a list	StackOverflow - https://stackoverflow.com/questions/960733/python-creating-a-dictionary-of-lists	I was able to create a dictionary with the value being a list and use both to find my average rank for each artist
4/26/21	I didn't know how to utilize the JOIN function to create a new table	https://www.sqlitetutorial.net/sqlite-inner-join/	I was able to take datapoints from two tables and combine them with a shared key to create a new table