

Team: GoBlue!

Team Members: Doae Kim, Sohyun Lim, Luis Solano

APIs, SQL, and Visualizations

Link to repo: <https://github.com/lsohyun/SI206-Final-Project.git>

Original Goals

We initially planned to use APIs from Spotify, Apple Music and Billboard, calculating the change in streams throughout time. However, the Apple Music API is offered free only to participants in the Apple's Affiliate Program, which none of our team members is available to. Our team decided to switch our goal to use Twitter's API instead. We planned to determine the top music on Spotify and Billboard, and look for the number of times it has been searched on twitter. Using data retrieved from Billboard and spotify would allow us to get good insight on which songs are currently trending/popular. We were curious whether the top hits in the music chart are actually popular in the real world, and whether people talk about it in popular social media like Twitter.

Goals Achieved

APIs/websites used: Spotify, Twitter, and Billboard

During our project, we changed our scope to accommodate what was available through the APIs. We used three APIs which were a track list on Spotify, Twitter and Billboard. We were trying to see if the songs on Spotify's today's top hits

(<https://open.spotify.com/playlist/37i9dQZF1DXcBWIGoYBM5M?si=68c1aec4aba54c89>) list are frequently mentioned on Twitter. And we used Billboard ranking to compare with the Spotify popularity to give more credibility to the popularity which spit out one of 0-100 points for each song from Spotify. When the title-singer name of the songs in this track list was searched on Twitter by keyword in the latest order, the difference in Twitter upload time (the most recently mentioned time and the most distant time) was confirmed, and we wanted to guess the popularity through this. And we expected a negative correlation to appear here. In other words, it was expected that the Twitter interval would be short if the song on the track was popular. When 10 API calls are made with song title + singer's name as keywords, 10 tweets containing the keyword are displayed, starting with the latest one.

Problems Encountered:

Problem 1: Difficulty in issuing Twitter API & limiting Twitter API calls (only 750 calls are available per 15 minutes)

- Solution: 10 API's per song-singer were called on twitter

Problem 2: Spotify doesn't give any data for popularity or streams on any podcast.

- Solution: We used the API of ‘Today’s Top Hits’ playlist, which is combined with the popular songs being streamed today.

Problem 3: One issue we faced using beautifulsoup when retrieving data from the Billboard 100 site was that when collecting, for example, information on the artist's name, it also included additional, unwanted information in the beginning, middle and the end of the soup list.

- Solution: Because there was unwanted data ruining the pattern in the beginning, middle and end of the soup list, we used indexing and slicing to be left with a list of desired data. By doing this, we were able to avoid elements such as “NEW” and “RE-ENTRY”.

Calculation file

to calculate the difference between the latest time and the furthest time (the most recent time from the most recent time) among tweets for each keyword and convert it into time units

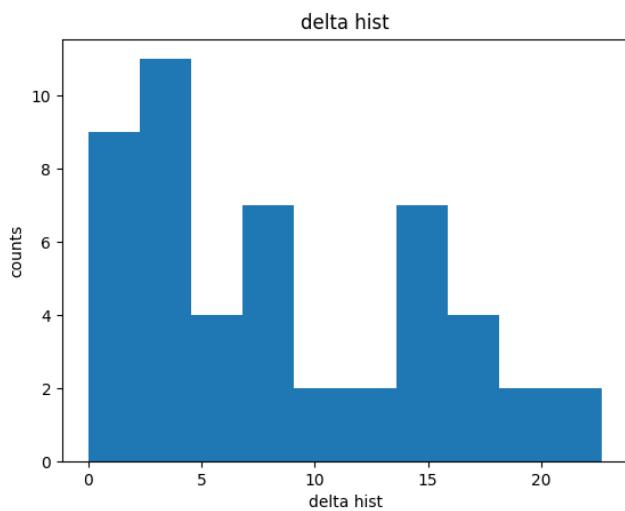
```
duration_ms,id,popularity,player,name
0,168601,4fouWK6VHhzl78KzQ1UjL,95,GAYLE,abcdefu
1,141805,5HCyWLXZPP0y6Gqg8tgaZ0,98,The Kid LAROI,STAY (with Justin Bieber)
2,174000,5dXWFMwD7I7zXsInONV10H,85,SZA,I Hate U
3,224694,0gpll1WMoJ6iYaPgMCL0gX,100,Adele,Easy On Me
4,207853,6b0fniqyCX7UaQSVvVG04I,58,Ed Sheeran,Shivers
5,226149,7DF81lvLdV3htIbuTwpc7lr,0,Juice WRLD,Girl Of My Dreams (with SUGA of BTS)
6,212000,27NovPIUIRrOzoChxAbJwK,97,Lil Nas X,INDUSTRY BABY (feat. Jack Harlow)
7,238805,3USxtqRwSYz57Ewm6wWRMp,89,Glass Animals,Heat Waves
8,202735,6zSpb8dQRaw0M1dK8PBwQz,96,Elton John,Cold Heart - PNAU Remix
9,145800,2Xr1dTzJee307rmrk8c0g,95,CKay,love nwantiti (ah ah ah)
10,172626,6Uj1ctrB0j0as8xZXGgkK4,96,Doja Cat,Woman
11,157890,20on25jryn53hWgthhWW3,93,ACRAZE,Do It To It
12,189727,5J6rTmMjF9DVIAF8G3M9n4,83,Juice WRLD,Wandered To LA (with Justin Bieber)
13,193506,00B1m7zeNggYLptW6zg8cj,93,Post Malone,One Right Now (with The Weeknd)
14,197442,4iN16F8JtVxG2UTzp3avG1,90,Bruno Mars,Smokin Out The Window
15,225148,3Kkj03cT83cw09VJyrlLnWx,94,Adele,Oh My God
16,231041,6PQ88X9TkUIAUIZJHw2upE,90,Ed Sheeran,Bad Habits
17,153190,5dPz35akJjpqb17yeqNwqH,85,Justin Bieber,Ghost
18,173381,1r9xUip0qqNwgBpENDsvJ,96,Imagine Dragons,Enemy (with JID) - from the series Arcane League of Legends
19,228000,3FeVmId7tL5YN8B7R3imoM,94,Coldplay,My Universe
20,222800,2KnLkZ3z7P03kgVGHGdpD,82,Shawn Mendes,It'll Be Okay
21,613026,5enwxA8aAbwZbf5qCHORxi,96,Taylor Swift,All Too Well (10 Minute Version) (Taylor's Version) (From The Vault)
22,160656,6f5ExP43esnvdkPddwKXJH,91,NEIKED,Better Days (NEIKED x Mae Muller x Polo G)
23,287120,5fwSHlTEwpluwOM0Sxnh5k,95,Farruko,Pepas
24,148197,3uUuGVFu1V7jTQL60S1r8z,93,Lost Frequencies,Where Are You Now
25,237720,15OC1UWh018vhZYmZ3Gluk,88,Jaymes Young,Infinity
26,143901,0e8nrVls4Qqv5Rfa2Uhqm0,94,Lil Nas X,THATS WHAT I WANT
27,210384,6w8ZPYdnGajyfPddTwdthN,90,Adele,Can I Get It
28,210560,3Vi5XqYrmQg0VByajMWSvCi,93,Doja Cat,Need to Know
29,151444,32BeYxKPrig1LefhCs0Xuo,88,Billie Eilish,Happier Than Ever - Edit
30,188102,6dmXZ9B5HdFAyzHeTneYBK,89,ROSALÍA,LA FAMA (with The Weeknd)
31,204763,15HMh4yxdF4wyxSZSl0gZ,91,Amaarae,SAD GIRLZ LUV MONEY Remix (feat. Kali Uchis and Moliy)
32,232813,3n021U41SZZ0N3fuWr9nb,84,Emmy Meli,I AM WOMAN
33,168020,2gQPv5jvVPqU2a9HhMN01v,91,BoyWithUke,Toxic
34,211560,3Wrjm47oTz2sjIgc115e,95,Måneskin,Beggin'
35,234000,2gpWfyu7eZ01zzncHpx0ta,93,Swedish House Mafia,Moth To A Flame (with The Weeknd)
36,201667,3QPBoCwfIC0CdFFvmqn60F,78,Ariana Grande,Just Look Up (From Don't Look Up)
37,140500,0bIT7K9Becu2dtXK1Q3cZNb,91,Tiësto,Don't Be Shy
38,168227,7hU3IHwjX150XLotVmjd0q,92,LISA,MONEY
39,161803,0eu4C55hL6x29mmeAjytzC,93,Oliver Tree,Life Goes On
40,193733,135Lf400CzlMMf0xbEUslH,88,Benson Boone,GHOST TOWN
41,178147,6PERP62TejQjgHu810HxgM,42,Olivia Rodrigo,good 4 u
42,162680,07MDkzWARZaLEdKxo6yArG,92,THE ANXIETY,Meet Me At Our Spot
43,204314,2EBnvom1dTybtm75a3Xh1T,85,Tate McRae,feel like shit
```

Visualizations

1 . Title: Histogram of first-last hours difference mentioned on Twitter.

X-axis: (time difference between first tweet and last tweet of 10 search)

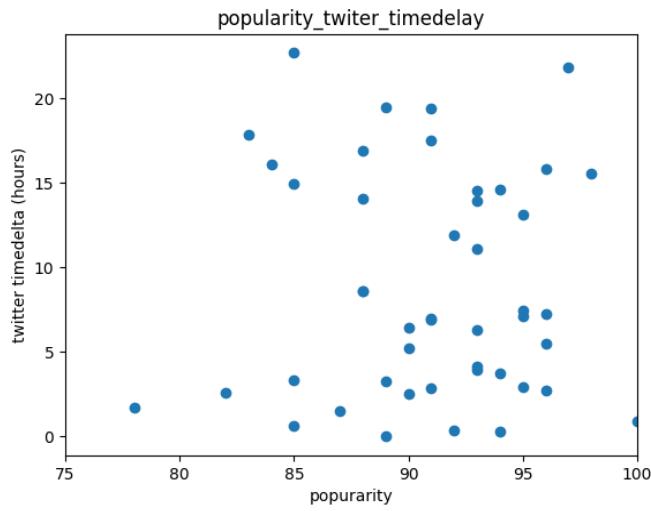
Y-axis: number of data in corresponding section (row number in database)



2. Title: Song popularity provided by Spotify / Twitter time difference

X-axis: Song popularity (a number between 0-100)

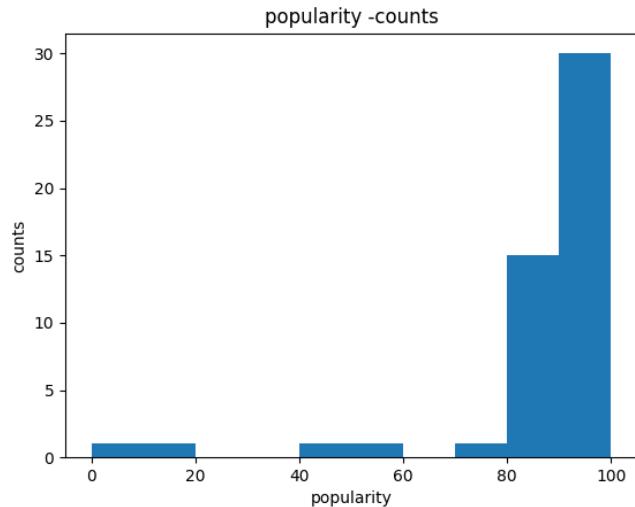
Y-axis: (time difference between first tweet and last tweet of 10 search)



3. Title: popularity -counts

X-axis: Song popularity (a given number between 0-100)

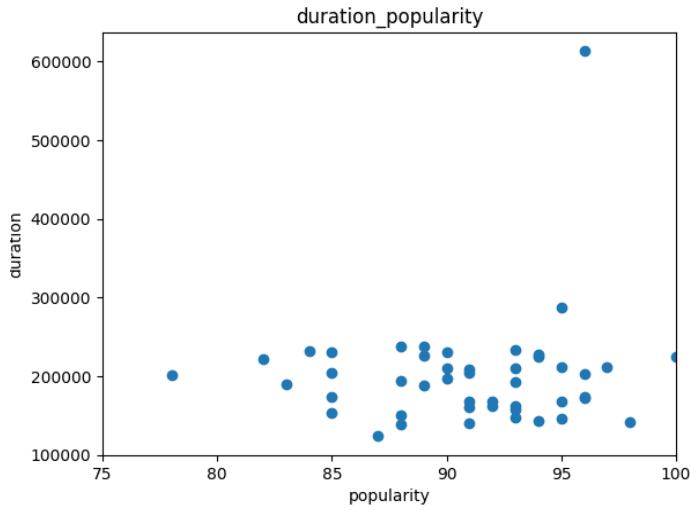
Y-axis: number of data in corresponding section (db row number)



4. Title: duration_popularity

X-axis: Song popularity (a given number between 0-100)

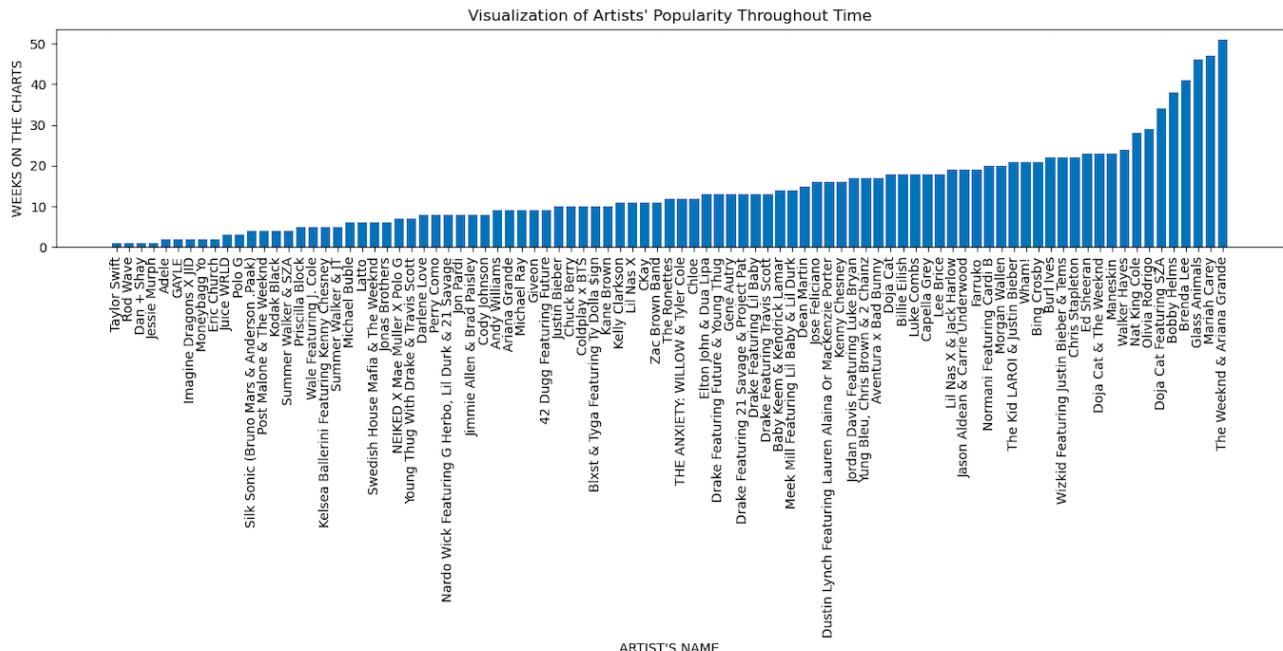
Y-axis: (Duration of the song played in total)



5. Title: Visualization of Artists' Popularity Throughout Time

X-axis: Artist's Name

Y-axis: Weeks on chart



Instructions for running code

sp.py must be run first. Next, twit.py can be run. Billboard-Hits.py can be run after (after running, it will update the database and show visualizations).

sp.py

Collect playlist's song information, singer name, popularity rating, running time of each song and uploads to track table. Playlist has 50 songs in total. One time data extraction and its iteration would occur 50 database table insert.

Twitter.py

From the track table, call the spotify playlist data. For each song, call recent 10 tweets with keyword "song name + singer". Saves tweet main contents, date, location and id and saves into twit table. Data from api draws graphs for visualization. There is total 50 song in the track so with the for loop it calls 50 apis. For each api, 10 twit returns, so total of 500 data are acquired.

Env.py

Saves key value for api access

Billboard-Hits.py

This file will utilize BeautifulSoup to scrape desired information from the billboard.com/charts/hot-100 site. This will create a table Billboard100, collect the artist name, song name, and weeks on the chart and insert it into the table (Billboard100). With this, a visualization portraying popularity of the artist throughout time will be generated for each one on the list. The file only needs to be run once.

Code Documentation: (what each function does including its input and output)

Sp.py: Using Spotify API, extracts the title, popularity, singer name, play time, and the like of the tracks listed in a specific playlist (hot's 50, id: 37i9dQZF1DXcBWIGoYBM5M) and store them in Spotify.sqlite (track table).

```

def get_play_track(playlist_id):
    # Takes in the playlist_id as an input to get data from it, and output is data frame
    # of track of playlist

def dbUpdate(df):
    # Takes dataframe as an input and updates it to DataBase
    [output: none]

def visualization(df):
    # scatter and histogram

```

Twitter.py: From the previous track table, call the spotify playlist data. Using Twitter api, search for ‘song titles + singers’ as keywords for each song, search 10 from recent data. Saves main twit content, date, location, and ID and saves into twit table. Data from api draws the visualization.

```

def get_data_from_db():
    # get data from spotify table to dataframe
    [input: none, output: dataframe of Spotify database]

def get_data(df):
    # twitter data get by api (creates twit table and then get data and insert from each
    # song)
    [input: dataframe of Spotify db, output: none]

def visualization():
    # gets data from twit, calculates time delta needs for visualization, converts second
    # to hours, get data from spotify database, merge two dataframe to draw scatter plot and
    # histogram

```

env.py: saves key for api access

```

# spotify
spcid = 'b48d6e811f9b43f1b2ff4705ff134d90'
spsecret = '875be44f97e449af996c9cf829cfa7ab'

# twitter
twitter_consumer_key = "DQiJMJdiwaBUQTRMLN5aYzhAt"
twitter_consumer_secret = "81jrNK5s0rxJ50QLRAREg8DNsPPwZjzwOy8hhFpjuKkjmrta0V"
twitter_access_token = "1461639316317171715-hp5pyBwTVEdA9hcd41Sz7iF7iRX0Ft"
twitter_access_secret = "QuxzK52AMRnbRpswKXdrAQ0nUGyFkmNE7nQjpIYMtbnUC"

```

Bilboard-Hits.py: using Billboard website, collected data using Beautiful soup.

```
def collect_data():
    #gathers the ranks, artists, and song from the site -- returns a list of tuples with
    artist, sng, weeks in chart

def set_up_db(db_name):
    #set up database

def create_table(cur, conn):
    #creates table Billboard100 that will have artist, song, and weeks on chart

def main():
    #call the above function and creates visualization and inserts data into table
```

Resources

Date	Issue Description	Location of Resource	Result
11/22	Needed to use Twitter API to pull search information	https://developer.twitter.com/en/docs/api-reference-index	API reference index
11/22	Needed to set up Spotify and Access Spotify's Web API	https://stackoverflow.com/questions/54494889/how-to-set-up-spotipy-and-access-spotifys-web-api https://spotipy.readthedocs.io/en/latest/	Made requests to Spotify's API with spotipy : Figured out how to use Spotify API
11/25	Needed to use Spotify Web API to pull playlist information	https://developer.spotify.com/documentation/web-api/	returned JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue
11/23	Needed to get familiar to tags and classes used in the billboard.com site	billboard.com	Found the tags needed to access data
11/25	Needed to add the sqlite3 module to Python	https://stackoverflow.com/questions/19530974/how-can-i-add-the-sqlite3-module-to-python	Used (pip install pysqlite3) for Python version

		e3-module-to-python	3
11/22	Needed to install tweepy module	https://stackoverflow.com/questions/34463574/tweepy-installation	Used (pip install tweepy)
12/1	Needed to visualize histogram and scatter plot using Matplotlib	https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.html	I was able to create a more complex diagram