Final Project Report

Youxi Zhao

1. Project Goal

Examine the differences in characteristics of popular songs between 1970 and 2020.

2. Achieved Goal

1. Using beautiful soup, collect the music name on billboard's top 100 list from billboard website.

2. Use the Spotify API to look up detailed audio features by music name.

3. Join the tables and study the characteristics of popular songs from different periods. (Using box plots and scatter plot closely examine two specific features: Instrumentalness and speechiness)

4. Achieved Goal:

1). find out that the spechiness in 1970 is greatly smaller than that of 2020, which indicates popular songs in 2020 contains more spoken words, and it may due to the rise of rap music.

2). There is no huge difference in Instrumentalness of two periods. But the the song in 2020 has slightly smaller instrumentalness, which means there is more vocal content (e.g. rap or spoken word).

3). For the loudness, the song in 2020 have greater loudness than that in 1970.

Reference:

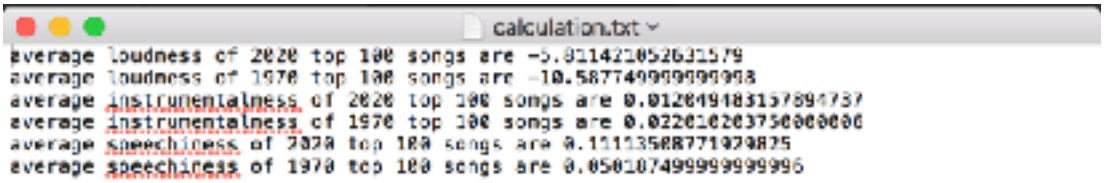
Instrumentalness: Instrumentalness predicts whether a track contains no vocals on a scale of 0 to 1. “Ooh” and “aah” sounds are treated as instrumentals as well. The closer the value is to 1, the more likely there is no vocal content (e.g. a soundtrack) and the closer it is to zero, the more vocal it is (e.g. rap or spoken word).

Loudness: The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude).
Speechiness: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

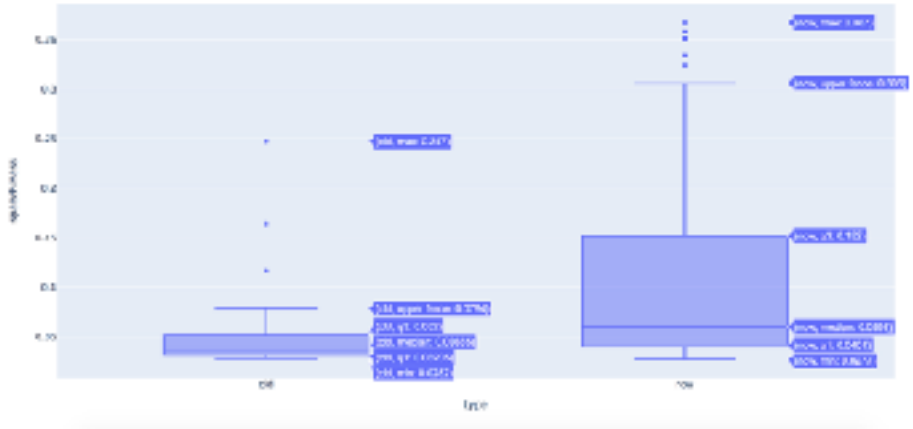
3. The problems that I faced

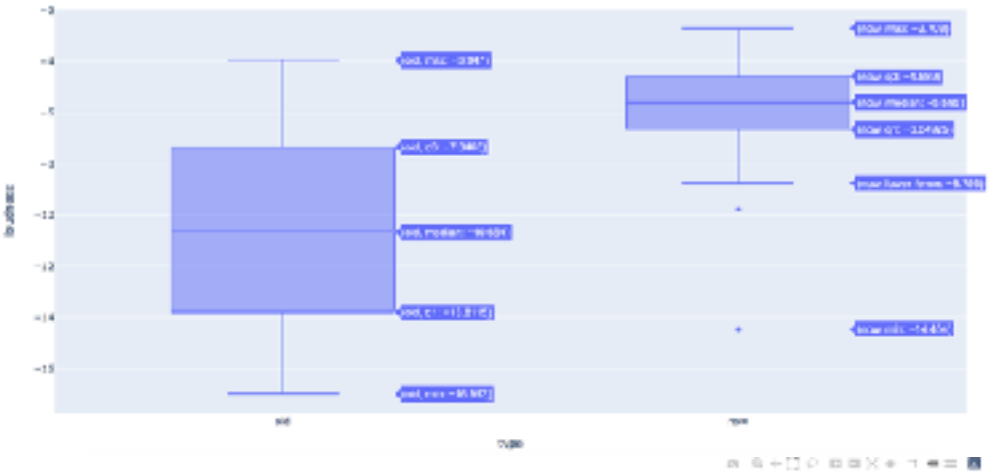
1. When getting track audio features from Spotify API using track name, it returns not only the track I'm looking for but also the related track with similar name. For example, when I search for "The Box", it also returns "The Boxer", "THE BOX", etc. and there is also different songs share the same name. So I failed to use the title as primary key. And then I found out each song in Spotify has its unique song_id. Therefore I used their id as primary key instead.
2. When I join the Billboard Table and the Spotify Table on track name and track artist. There still a lot of duplicates items left in the table. So I sort by the track's popularity by Group and left the song with highest popularity as the one to study at.

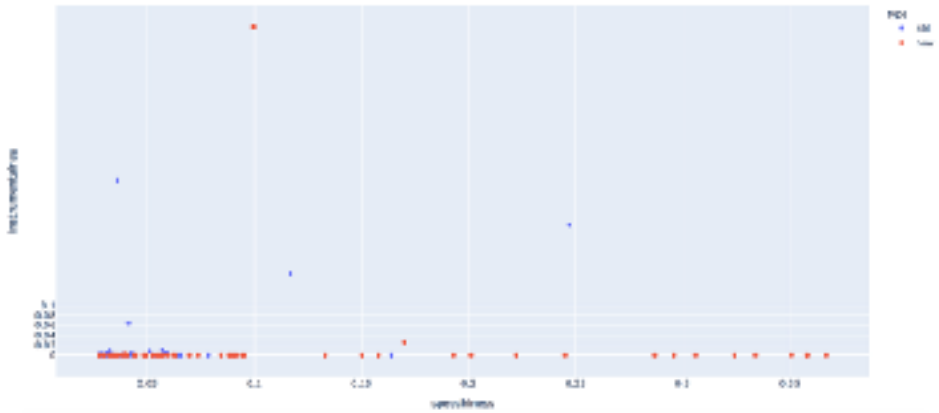
4.file that contains the calculations



5. The visualizations







•Speechiness: http://127.0.0.1:49863

•Instrumentalness: http://127.0.0.1:49708

•Loudeness: http://127.0.0.1:49801

•Scatter Plot: http://127.0.0.1:49584

6. Instructions for running code

1. Run get_billboarddata.py, it create mymusic.db database and set up Billboard Table which contains track title and artists from 2020 top 100 list and 1970 top 100 list.
2. Run create_tracks_table. It set up old_Spotify_Data and now_Spotify_Data which contain tracks' audio feature from Spotify API.

   It saves 20 items into the table each time, and there are 844 rows in old_Spotify_Data and

   962 rows in now_Spotify_Data. Therefore, it needs to run at least 49 times to fully store the

   data.
3. Run join and visualize.py, it create join_table_old and join_table_now, which are the table

   join on Billboard and Spotify. Then, it creates table final_table. And use the data in final_table to do the calculation and write the result into the file. Also use the data to do

   visualization and create 3 box plots and 1 scatterplot.

7. Documentation for each function

1. get_billboarddata.py
   - ① def setupBillboardTable(cur, conn)

     Using Beautiful Soup, collecting 1970 and 20020's track name and track artist from Top 100 list on BillBoard website.
2. create_tracks_table.py
   1. def setupPresentSpotifyTable(cur,conn)

      set up now_Spotify_Data table, Use the Spotify API to look up detailed audio features by music name, the music name is extracted from old_name column of BillBoard table. save the audio features: popularity, loudness, speechiness, instrumental ness into the table

2. def setupOldSpotifyTable(cur,conn)

   set up old_Spotify_Data table, Use the Spotify API to look up detailed audio features by music name, the music name is extracted from now_name column of BillBoard table. save the audio features: popularity, loudness, speechiness, instrumental ness into the table

3. join and visualize.py
   1. def setup_table(cur,conn)

      Respectively join old_Spotify_Data table and now_Spotify_Data table with BillBoard table, and get rid of the duplicate songs and save them into join_table_now and join_table_old

      Union two join table together and get rid of the Null value in the table. Save it into final_table

   2. def calculation(cur,conn)

      Using the data in final_table, calculate the average value of loudness, speechiness, instrumentalness of the songs from different period

   3. def visualize(curr, conn)

      Using the data from final_table, use Plotly to visualize the difference between two periods, and make up 3 box plots and 1 scatterplot

8. Document all resources

Date	Issue Description	Location of Resource	Result
4.20	Spotify API — how to get audio features and what is audio features contain	https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/	use: sp.search(“track name”,type="track")

Date	Issue Description	Location of Resource	Result
4.20	how to get authorization from Spotify	https://spotipy.readthedocs.io/en/2.12.0/	use: client_credentials_manager = SpotifyClientCredentials(client_id=cid, client_secret=secret) sp =spotipy.Spotify(client_credentials_manager=client_credentials_manager)
4.20	get top 100 song from BillBoard website	https://www.billboard.com/charts/hot-100	use BeautifulSoup to extract track name and track artist
4.20	what to compare among all the features provided by Spotify API	https://towardsdatascience.com/billboard-hot-100-analytics-using-data-to-understand-the-shift-in-popular-music-in-the-last-60-ac3919d39b49	Choose to compare loudness, speechiness, instrumentalness