550 Project Proposal - Spotify Songs

Haorui Li, haoruili@seas.upenn.edu, GitHub username: li002302
Yiting Li, ytingli@seas.upenn.edu,GitHub username: yitingliii
Jason Pan, jp2286@seas.upenn.edu, GitHub username: lilpannn
Kris Zhang, krisz@seas.upenn.edu, GitHub username: Kristian815

1. Website Idea Description

We want to make a music-focused search engine that uses the songs_df and track_df datasets, to create a fun website for music discovery. Users can search for songs by name, genre, or attributes like popularity and danceability. The site will feature intuitive filters, allowing users to refine their searches according to their preferences, such as finding upbeat dance tracks or soothing acoustic melodies.

2. Datasets Information

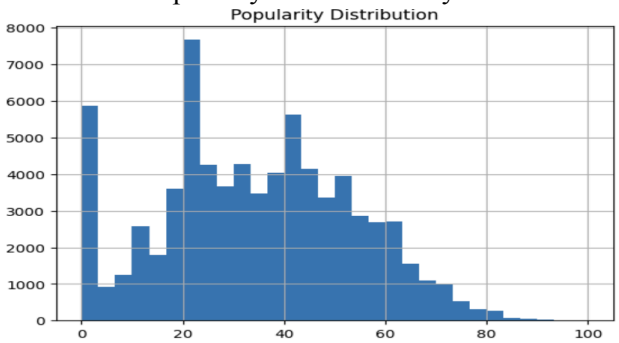
2.1 Dataset 114000 Spotify Songs (114k_songs)

a. Description: It contains song metadata, including attributes like track ID, artist, album, track name, popularity, and audio features such as danceability, energy, and tempo.
b. Link: https://www.kaggle.com/datasets/priyamchoksi/spotify-dataset-114k-songs?resource=download
c. As we're not scraping the data and using the tables:

i. relevant size statistics: MB/GB: 11.2+ MB memory usage, Size: 114000 rows, Number of attributes: 19 columns.

ii. summary statistics (after data cleaning):

- It includes 114,000 songs with a mean popularity of 33.2, showcasing a range from obscure to popular tracks. Songs average 3.8 minutes in length, with moderate danceability (0.57) and energy (0.64). The tempo averages 122 BPM, spanning all musical keys with a slight preference for major keys. Most tracks have low speech content (mean 0.085) and are minimally instrumental. Loudness averages -8.26 dB, typical of commercial music, and emotional tone is neutral (valence 0.47). It covers various musical features, reflecting diverse tempos, tones, and structures. Popularity seems normally distributed and right-skewed.



	popularity	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature
count	114000.000000	1.140000e+05	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000	114000.000000
mean	33.238535	2.280292e+05	0.566800	0.641383	5.309140	-8.258960	0.637553	0.084652	0.314910	0.156050	0.213553	0.474068	122.147837	3.904035
std	22.305078	1.072977e+05	0.173542	0.251529	3.559987	5.029337	0.480709	0.105732	0.332523	0.309555	0.190378	0.259261	29.978197	0.432621
min	0.000000	0.000000e+00	0.000000	0.000000	0.000000	-49.531000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	17.000000	1.740660e+05	0.456000	0.472000	2.000000	-10.013000	0.000000	0.035900	0.016900	0.000000	0.098000	0.260000	99.218750	4.000000
50%	35.000000	2.129060e+05	0.580000	0.685000	5.000000	-7.004000	1.000000	0.048900	0.169000	0.000042	0.132000	0.464000	122.017000	4.000000
75%	50.000000	2.615060e+05	0.695000	0.854000	8.000000	-5.003000	1.000000	0.084500	0.598000	0.049000	0.273000	0.683000	140.071000	4.000000
max	100.000000	5.237295e+06	0.985000	1.000000	11.000000	4.532000	1.000000	0.965000	0.996000	1.000000	1.000000	0.995000	243.372000	5.000000

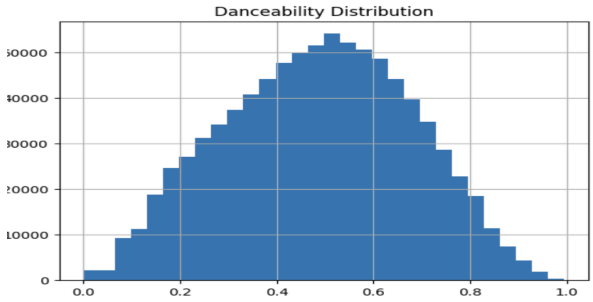
2.2 Dataset Spotify 1.2M+ Songs (1m_track_features)

a. Description: It has more detailed track-level data, including track IDs, album information, artists, track numbers, and audio features such as danceability, speechiness, acoustics, and tempo.
b. Link: https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs
c. As we're not scraping the data and using the tables:

i. relevant size statistics: MB/GB: 156.6+ MB memory usage, Size: 1204025 rows, Number of attributes: 24 columns

ii. summary statistics of several attributes (after data cleaning):

- The dataset contains 850,937 tracks with a mean track number of 7.83, indicating most appear mid-album. Most albums are single-disc (mean 1.06). Tracks show moderate danceability (0.49), energy (0.50), and an average tempo of 117 BPM. Songs are produced at a moderate loudness level (-12.2 dB) and favor major keys. The average track duration is 4.17 minutes, and most are in 4/4 time. With low speechiness (0.087) and a valence score of 0.21, the emotional tone is neutral to slightly negative. The dataset spans tracks from 1900 to 2020, with a median release year of 2008. Also found danceability almost has a normal distribution.



	track_number	disc_number	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	year
count	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937.00	850937
mean	7.84	1.06	0.49	0.50	5.20	-12.22	0.67	0.09	0.46	0.30	0.21	0.42	117.11	250461.81	3.82	2007.08
std	6.11	0.30	0.19	0.30	3.54	7.26	0.47	0.12	0.39	0.38	0.19	0.27	31.02	174548.91	0.58	11.71
min	1.00	1.00	0.00	0.00	0.00	-60.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1000.00	0.00	0.00
25%	3.00	1.00	0.34	0.23	2.00	-16.13	0.00	0.04	0.04	0.00	0.10	0.18	93.54	170945.00	4.00	2002.00
50%	7.00	1.00	0.49	0.51	5.00	-10.06	1.00	0.05	0.42	0.01	0.13	0.40	116.01	223973.00	4.00	2008.00
75%	11.00	1.00	0.63	0.77	8.00	-6.82	1.00	0.07	0.88	0.76	0.25	0.64	136.40	288947.00	4.00	2015.00
max	50.00	12.00	0.99	1.00	11.00	7.12	1.00	0.97	1.00	1.00	1.00	1.00	248.93	6061090.00	5.00	2020.00

2.3 How they overlap

- They overlap in song-level metadata like track IDs and audio features (e.g. danceability). ***114k_songs(rack_id)*** and ***1m_track_features(id)*** has 0…* vs. 0…* relationship.

3. Queries Ideas

- JOIN these two datasets together ON track_id/track_name to see the commons. (we have already cleaned the dataset so that they have unique track_names right now)
- Aggregate by COUNT() / MAX() to get how many songs / suggest the best song they want
- SELECT attributes FROM the 114k_songs, and ORDER BY danceability or genre WHERE danceability > 0.5 to recommend dancing songs
- Use WITH xxx AS to first SELECT songs FROM the merged dataset WHERE they are in a specific track_name(s) asked by the user, then with this to SLECT songs FROM xxx WHERE the loudness is lower than an amount yyy to recommend sleeping songs
- Use several WITH … AS & UNION ALL to find all kinds of isolated types of songs desired