Week 2 Writeup – Ingest and Explore the Dataset
Ryan

Project Recap:

Our group's problem statement is that with the vast amount of music available via streaming services such as Spotify, exploring and finding new and enjoyable music is harder than ever. There are so many options, we often suffer from choice overload bias, reverting back to what is comfortable (old, mainstream music). Current streaming services such as Spotify have rolled out "DJs" to help users explore new music and get a mix of music they already listen to. The problem with these models is that they are largely based on user listening history. What a user liked 5 years ago may not hold true today. In addition, these algorithms almost "punish" a user from exploring music on their own as these algorithms take into account listening habits, assuming all song listens carry a positive sentiment for the user which is not always the case.

Our model aims to be a user input model, where the user can input specific songs, they enjoy to get the top 5 recommendations from every era of modern music. We believe this type of model gives the user more customization on their recommendations and can streamline exploring newer and less mainstream artists on Spotify which sees over 100,000 audio uploads per day.

Target Variable

The target variable in our project is the set of 5 similar songs recommended for a given input track. The recommendations will be generated using a KNN model using cosine distance to pull the 5 songs closest to the input song. The target variable here is not a traditional single numeric label, the "target" in our case is the outcome of the recommender, the system's best guess at what songs feel alike. We would like to evaluate the output by checking if similar songs share musical qualities (e.g., genre, BPM, danceability) and thematic content.

For our recommendations the target variable will be the recommended song and artist pair.

Predictors

The predictors that were pulled for our dataset can be split into different sections like audio, lyrics (text data) and metadata.

The metadata features include:

- Popularity
- BPM
- Dance
- Energy
- Acoustic
- Instrumental
- Happy
- Live
- Loud (Db)
- Key
- Camelot

These categories are referred to as “song metadata”. Some features follow a 0-100 grading scale (popularity, BPM, dance, energy, acoustic, instrumental, happy, and live). Loud follows an interesting scale with a minimum of -26 to -1 with a mean of -7.64 Dbs.

Upon further investigation, Key and Camelot represent the same metric in different ways. We will drop the “Key” column and later code Camelot into a unit circle in two columns (sin and cos) in a later week.

Once EDA is performed some variables such as BPM and Acoustic (with heavy right skewed distributions) may benefit from a transformation such as taking the square root. All variables will have to be normalized after transformations and EDA to fit a 0-1 scale as to not weigh/overrepresent certain variables on higher magnitude scales.

We’ll also use lyrics as text data, which we plan to tokenize and embed to capture mood, theme, or emotion behind a song. These embeddings will be another layer of similarity in our latent space. We will be using the Twitter Glove embedding API as we believe Twitter language best mimics lyrical language (slang and vulgar language is prevalent in music)

Finally, genre data will have to be cleaned up as there are over 250 current genres, many with less than 5 songs. We plan to work to combine underrepresented and obscure genres into a

“parent genre” to reduce dimensionality and overfitting risks in the model. In addition, we only plan on using genre tags as about 10% of our cosine distance due to concerns about overfitting the model (model could recommend only other songs from the specific genre of the input). We want our model to focus on music metadata and lyrics while mixing in genre tags slightly.

Initial Cleaning Steps

We drew our data by downloading song data (title, artist, time, etc.) from 17 different public Spotify playlists comprising of about 9,000 songs for our initial dataset. Since this dataset was formed by combining multiple playlists, we expect (and determined) that duplicate songs are fairly common. Cleaning was required to delete duplicate rows and to keep the song with the highest popularity score in the event of remastered songs.

Checking for duplicates based on song title and artist quickly removed about 1,900 songs, and removing remixes knocked out another 200 songs. Then, after this light cleaning was done, we were able to scrape music genre tags from musicbrainz.org, an open-source database of music metadata. We were able to collect track-level genre tags for about 5,500 of the 6,600 songs. This is different from artist level genre tags which are the same for all songs in an artist's discography, giving us a more accurate genre for each specific song.

Now we were able to scrape lyric data from genius.com, which has a massive collection of lyrics. Of the 5,500 scraped, about 4,800 were able to pull full lyrics. Of those, another 300 could be removed that were not English language songs. At this point before any deep cleaning was done, our dataset went from 8,685 songs to 4,488 songs, nearly halved in size.

At this point a few more deep cleaning steps were taken to ensure a quality dataset with no duplicates. Some more songs were removed that had slightly different names, were remastered versions, or included features by other artists but were otherwise the same exact song. Also, ~300 songs were discovered that didn’t have a time feature, so those were also removed. In all, after the full dataset was cleaned, we had a sample size of about 3,600 songs.

Basic Data Stats

The project's dataset may include a diverse and carefully chosen collection of songs highlighting various contemporary musical genres and production qualities. We compiled the dataset using

publicly available playlists and metadata, ensuring that scraping methods removed duplicate entries, remixed versions, and language-specific music other than English. We decided to only use English songs as finding a tokenizer that integrates multiple languages is hard and would add too much complexity to our model. However, our model can be used for other languages by changing the tokenizer, but it is recommended only one language be modeled at a time. Only distinct, excellent tracks fit for lyrical and audio analysis are included in the final dataset.

Although softer or quieter types are not excluded, the dataset has a modest bias toward lively and joyful music. The distribution of danceability and energy levels guarantees that both slower, moodier picks and highly rhythmic, movement-friendly tracks are included. Because of this balance, the dataset is especially well-suited for developing systems that accommodate a broad range of listener preferences, from casual pop lovers to those looking for more experimental or ambient music.

The data collection demonstrates current trends in audio production. Since most recordings are expertly recorded and mastered in a studio, the collection provides accurate data and a listening experience. There is a focus on digitally created content, consistent with streaming platform standards and contemporary music consumption patterns.

The recordings in this dataset show a wide range of emotional tones, from joyful to introspective and melancholy. This variation enhances the latent space's richness and the recommendation engine's ability to recognize similar sounds and emotional or thematic alignment. Furthermore, poetic data improves the aural features and makes mood- or story-based grouping possible by introducing an additional layer of interpretive complexity. The metadata in particular will be helpful to differentiate mood (such as happiness score) as lyrical data alone may not be enough. For example, there are many happy and sad songs about love, so identifying it is a song about love may be insufficient by itself to recommend songs.

We decided to drop time from our modeling plans. We feel as if time of a song is irrelevant in our lives when finding songs we enjoy. We like songs that are both short and long and we wanted to avoid the model from restraining from recommending songs with extreme times (either short or very long). We feared songs like “Bohemian Rapsody” or “American Pie” may not have many close neighbors

in KNN as they are both 8+ minute songs, more than double the average. We also did not want to delete longer songs from the playlist for being outliers as some of the most popular songs of all time fall on the longer end of the time spectrum.

This dataset effectively balances variation and consistency. While maintaining sufficient flexibility to enable strong similarity identification and suggestions, the dataset captures the essential components of contemporary music. Its structure facilitates an adaptable, user-focused design for a music recommendation system that can function well in various listening situations, moods, and genres.

Data Dictionary + Types

Variable	Description	Type
Song	Title of the song	object
Artists	The name of the person or group who released the song	object
Popularity	Popularity score (0-100, 100 being most popular) based on total streams and how recent those have been	int64
BPM	The estimated speed of a track measured in beats per minute (BPM). How fast or slow a piece is, based on the average duration between beats.	int64
Time	Duration of the song from start to finish	int64
Dance	This feature measures how	int64

	well-suited a track is for dancing, taking into account elements such as tempo, rhythm consistency, beat strength, and overall flow. A higher danceability score means the track is easier and more enjoyable to dance to. A value of 0 is least danceable and 100 is most danceable.	
Energy	Energy is a measure from 0 to 100 and represents a perceptual measure of intensity and activity. Tracks with higher energy scores are typically more lively.	int64
Acoustic	A confidence score ranging from 0 to 100 indicating how likely the track is acoustic, with 100 meaning very high confidence that the track is acoustic.	int64
Happy	A score between 0 and 100 that reflects the musical positivity of a track. Higher valence values indicate a more positive mood (such as happy, cheerful, or euphoric), while lower values suggest a more negative mood (like	int64

	sad, depressed, or angry).	
Live	Liveness indicates the presence of an audience in the recording. Higher values suggest a greater chance the track was recorded live. A score above 80 strongly suggests the track is a live performance.	int64
Loud (Db)	The average loudness of a track measured in decibels (dB), calculated across its full duration. Loudness reflects the perceived intensity of sound, corresponding to its amplitude.	int64
Camelot	The musical key of the track, represented as an integer using standard Pitch Class notation (e.g., 0 = C, 1 = C♯/D♭, 2 = D, etc.). A value of -1 indicates that no key was detected.	object
Genre Tags	List of 4-5 genre types the song falls under	object
Lyrics	Lyrics to the song stored as text data	object

New Dataset : Spotify Playlist Analyzer - Chosic

References : https://help.spotontrack.com/article/what-do-the-audio-features-mean

https://onlyoneaman.medium.com/unleashing-the-power-of-audio-features-with-the-spotify-api-c544fda1af40

https://developer.spotify.com/documentation/web-api/reference/get-track