CSC111 Project Proposal: Algorithmic Music Recommender (A.R.M.)

Johnson Qin, Olivier D'Aragon Flores & Hongyu Zhu

Wednesday, March 5, 2025

Problem Description and Research Question

We have chosen to create a program that suggests songs based on a user's interests because when listening to music, we often find ourselves listening to the same songs over and over again, without leaving our comfort zone and exploring new songs. We want to develop a tool that will suggest songs similar to our interests to add variety in our playlist and ensure that new songs align with our preferences, all without having to take the time to find new songs ourselves. As such, we decided to make a recommendation algorithm to help us find similar music more efficiently, and spice up our playlists a little bit, all without wasting our time!

Project Goal: Recommend songs to users by taking in songs they like and finding similar songs. The algorithm will compare the characteristics of the song from a Spotify song database to generate predictions. The experience should be customizable, allowing users to tweak recommendation parameters in order to obtain different results.

The user should be aware of some different variables associated with each song (some examples below):

- `Danceability`: "describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable."
- `Energy`: "a measure from 0.0 to 1.0 [that] represents a perceptual measure of intensity and activity."
- `Speechiness`: "detects the presence of spoken words in a track. The more [...] speech-like the recording [...], the closer to 1.0 the attribute value."

Computational Plan

The program should take in (valid) user input songs and personalized search parameters, perform calculations (exact algorithm TBD), and return recommended similar songs. Personalized search parameters could include:

- What song characteristics to use (ex: key: low importance, danceability: high importance
- How many songs to input (ex: 1 song only)
- How many songs to receive (ex: 5 similar songs)
- How much of the database to parse through (ex: shallow = 500 songs, maximum = 30,000)
- Element of randomness to add spice (ex: lower randomness means new songs are common)

Our program works interactively by asking for user input and customization of search parameters. Finally, the program generates a graph of similar songs, linking those who more closely resemble each other, grouping similar ones together in order to show the relationship they have to each other. The way in which elements in the graph are grouped is customizable.

The Numpy library allows us to manipulate `np.arrays`, which are considerably more computationally efficient than python lists. Numpy also has very fast built-in mathematical operations on these arrays, optimized in C/C++, perfect for large-scale data parsing, such as `diff()` or `cross()`.

We will mostly use Pandas to store, visualize and debug the data we need at every step of the coding process. The main functionality we will be using is the `DataFrame` class. Pandas also meshes well with Numpy, with useful functions like `DataFrame.to_numpy()`.
We will probably use NetworkX to visualize the graph, but we might need to use matplotlib to show other statistics, such as a visual representation of the song characteristic weight chosen by the user. The exact methods we use will be determined during coding.

References

Arvidsson, Joakim. "30000 Spotify Songs." Kaggle, 1 Nov. 2023, www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs.
https://pandas.pydata.org/
https://numpy.org/doc/stable/index.html
https://networkx.org/
https://matplotlib.org/