Project Goals

Our final project is based on a dataset compiled by data scientist Yamac Eren Ay (link in our team contract). This dataset contains tracks on Spotify across several decades (1921-2020). Alongside the tracks, are characteristics including: ID, acousticness, danceability, energy, duration, instrumentalness, valence, popularity, tempo, liveness, loudness, speechiness, year, mode, explicit, key, artists, release date, and name. We found this to be a very fitting dataset given the quantity of data available at our disposable as well as the variety of different specs per piece of data. Furthermore, two of our team members are CS + Music majors, and have a solid understanding of the data and characteristics. A big picture idea for this project is to create a modular program where a user can input a song (that is already in this dataset), and receive a list of songs that they would enjoy based on how close their shared characteristics are.

Our ultimate goal with this final project is to store a cleaner version of this dataset within a structure that we have learned to apply this semester. We intend on using the differentiating factor of the dataset to create a K-D tree, which seems to be the most appropriate choice for such data. The factors will act as different dimensions which will allow us to implement a Nearest Neighbor Search Algorithm to find songs similar to one input by the user. Initially, the raw data set will have to be scraped and refined such that it only includes relevant results (ie: eliminating audiobooks). Since the data in our dataset is quantized, we will be able to give each song a unique ID based upon each of the relevant characteristics that will eventually be used to partition each plane at every level of the tree.

Although we are implementing a K-D tree, in order to break down songs based upon relevant features, the user inputted song will have to be part of the database already so that similar songs can be found and returned. This means that we will have to employ a secondary condition on top of the Search to make sure the program does not return the same value but

finds five songs with similar features within the threshold radius of a pre-defined hypersphere, thereby adapting the algorithm we used in the MP to better serve our needs.