AI Development Prep

The main features of the algorithms that we will use that will be difficult are Euclidian distance, data organisation and profile updating.

Euclidian distance will be computed in a similar way to the following pseudocode:

- Let n be the number of attributes per song.
- Iterate through all x songs.
- Square the first songs first attribute value.
- Square the user's corresponding attribute value.
- Square root the sum and add to an array.
- Do this for every attribute and every song.
- Return the k-lowest-values of that array.

Data organisation includes taking the returned songs from the Spotify API using the following query format:

https://developer.spotify.com/documentation/web-api/reference/#/operations/search

Then use the appropriate API requests to get the n details about all the songs. This data will most likely be stored in a .csv file with the ID of the song being first and the attributes after. As the artist of a song won't be used until collaborative-filtering is implemented, both the artist, name and other non-comparable attributes will not be stored in this file.

Before the Euclidian algorithm is performed the .csv file will be appropriately split into the rows for each song, ready for use.

Each song will be treated as an ID right up until it is to be played as this is the most efficient way of working with them.

The profile updating will be done using a set of rules that won't be changed by any algorithm. The reason for this is because this would lead to a 2-variable situation where the profile updating and the song recommendation could end up counteracting each other.

To update the profile, each attribute of a song is first divided by X times by the corresponding attribute of the user's profile. That value is then added to the profile attribute. In this case X is the number of songs already used to train the algorithm. This is done to stop drastic changes in the user's profile after one song choice.