Project proposal

1. Choice of dataset

We will make our own dataset inspired by this existing Kaggle Dataset, which contains a spreadsheet of many songs and their features from Spotify. We will select a playlist created by Spotify that contains at least 50 songs from each genre. Ideally, the playlists contain some of the currently most listened songs. They will be read using the spotify web API in order to obtain all of the features of each song. The main data set will be an input playlist from which we identify the songs the client likes the most.

2. Methodology

a. Data Preprocessing:

The data will include several details relating to the music.
We will use spotify's web API to extract the data from users spotify account and will put the information into python dictionaries from json format. This will allow us to analyze the data using python.

We will use python and matplotlib/seaborn to visualize the data, namely all the features that the songs have. After we identify the quantity of the songs with the **closest** features to the input playlist, we will mark them to choose a 50 song playlist that has the closest relation to the favorite songs inputted.

b. Machine learning model:

The goal of our model is to produce a list of song recommendations based on an input playlist. Thus, our model will use features of those songs to find and recommend similar songs that would match a user's taste in music.

We will use the k-means algorithm to mark the input data based on their features. We chose this machine learning model because it will allow us to detect patterns in a user's playlist by clustering. The recommendations will be based on the closest neighbors of the cluster of the input playlist. The number of the clusters in the initial part of the algorithm will be a hyperparameter. Hyperparameter search must be done in an accurate way to have an accurate model. One way to select a good value of k is to use the Elbow method. In this method, you evaluate the summed square error for k=1, k=2, etc. until the SSE starts to flatten out (the SSE for the k values are about the same). The point when this happens gives us a good k value.

One advantage of using this model is that this model is often used in systems that generate recommendations. It is quite a simple machine learning and it is easy to visualize the clusters (our dataset) and to see if the model is performing well on input data. Also, the algorithm will always give a result (even though it may not be optimal).

One potential problem with using this method is that the clusters can have different sizes and densities, which is something that may influence the recommendations. If the input is very close to a cluster with few songs, then the generated playlist may not be as accurate because it has to take songs from a different and further cluster that doesn't match the features of the input playlist as well. Furthermore, this model works best for clusters that have a more spherical shape.

Sources: K Means Clustering | K Means Clustering Algorithm in Python (analyticsvidhya.com)
K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks | by Imad Dabbura | Towards Data Science

c. Evaluation Metric:

To evaluate the algorithm we will ask the user to listen to the output playlist and let the model know if they liked it or not. They can rate out of 10 and over 70% result will be an ideal rate.

Moreover, we can run a quick survey gathering data about the user experience of the app while it is in its prototype form to evaluate the satisfaction with the generated playlist. We can do this using Google Forms.

3. Application:

What does the user input? How does the user provide inputs? (Is there a webcam? A way for users to submit images? text?)

The user will ideally type a playlist of 62 or more of their favorite songs into our webapp, but there is no specific number of inputs. However since it's a machine learning model, the more, the better.

What does the user receive as output, how will the output be displayed?
The output is a list of 50 recommended songs which will be generated by the model based on the input playlist. The output will be displayed as a playlist format that they can import into their spotify account as a new playlist.

We are planning to use the spotify Web API to read the data and ask the user to input their data, thus we will have a web application that handles all these difficulties with a good UI.