**Project Report for Objective 2**

Objective 2: Using pandas dataframe, recommend a list of genres with similar characteristics to a genre chosen by the user. Additionally, recommend a list of genres based on the user’s chosen artist.

Results:

The ‘genre recommendations’ section successfully returns a list of chosen length of similar genres to the genre input by the user. On the whole, the genres returned appear very similar to the input genre. For example, when ‘techno’ is input, the output is (for 10 genres):

\*Output here showing ‘techno’ output in Notebook\*

This was achieved by calculating a similarity index (where 0 is exactly the same and the further away from 0, the less similar) for each element in the genre column of the dataframe (using normalised vectors for each row and using Euclidean distance to represent the similarity to the chosen genre over all the characteristics). Characteristics include aspects such as ‘danceability’, ‘tempo’ and ‘acousticness.’ Therefore, genres with similar characteristics should be similar.

The genres were then sorted by similarity value (to the chosen genre) ascending and the first n elements retrieved to give the most similar n genres (where n is the user’s desired number of genres).

The limitation of this is that the recommended genres probably won’t be exactly the most similar genres, as only the characteristics present in the data.csv file were used to compare genres, when in fact, genres could be associated with many more characteristics (for example, artists weren’t included in this data file and all the genres associated with one artist are often similar). Nonetheless, it is sufficiently successful in achieving the objective.

Similarly, for the ‘genre recommendations by artist’ section, the user is recommended all the genres associated with their chosen artist. This is done simply by using a pandas dataframe and retrieving the ‘genres’ element at the same index as the given artist. For example, when ‘Radiohead’ is input, the output is as follows:

\*Output here showing genres associated with Radiohead\*

Visualisation:

After the user had been recommended a series of genres, the user was then prompted to select a metric by which to compare the genres from the available metrics in the dataframe. Matplotlib was used to display the comparison between the metric values for each of the genres as a bar chart (with each bar indicating a separate genre) including the original genre so that the user could see how that compares with the recommended genres. They are ordered by value ascending in terms of the selected metric so that the user can visualise their ranking.

So, for example, choosing ‘tempo’ for the genres similar to ‘techno’ produces the following:

\*Output here showing bar graph for genres similar to ‘techno’ based on their tempo\*

Conclusion Notes:

Achievements:

As we had anticipated, we were able to successfully return a curated Spotify playlist based on the user’s desired level of a certain characteristic (such as ‘danceability’). When we retrieved genre recommendations based on a selected genre by comparing over the set of characteristics that were available, the genres did appear very similar to that of the input genre (even if they were perhaps not the most similar genres). We were able to display a visual comparison of the similarity of these genres in terms of the user’s chosen metric.

We also successfully retrieved a set of genres associated with a user’s favourite artist. Finally, we added additional functionality to check the validity of user input and make it more error-proof (for example, by restricting the type and magnitude of the user input), and made the output layout more aesthetically pleasing to enhance user experience.

Future work:

In the future, we would expand the functionality of the playlist, artist and genre recommendations by recommending artists based on genres, and allowing the user to personally select what kind of recommendations they would like.

Once we were able to give artist recommendations, we could then return a playlist of a handful of songs by those artists, so that the user would only have to input a single genre or artist and would receive a specially curated playlist. This would be done by matching up the indexes of the dataframe in which the genre or artist appears and then retrieving songs with the same indexes. Alternatively, we would allow multiple inputs of genres or artists as a list so that the user could receive a recommendation which is even more specific to their tastes. This would be achieved by retrieving multiple sets of recommendations (as performed individually in objective 2) and prioritising those which are duplicates. We would also like to extend the usability to enhance user experience. We would do this by adding more interactive elements (such as clickable buttons and drop-down lists), making the design more aesthetic with colourful graphics relating to the user’s musical taste, and making the input handling even more robust to errors.

Limitations:

We were limited in that we didn’t have access to one single database with all the necessary information in: for example, the database that displayed information by genre didn’t have any associated songs or artists with that genre, and so we were limited in what we could select easily based on a genre. In the database with artist data, every artist was associated with multiple genres and so it would not have been trivial to select a genre and return a list of associated artists. Given the time limitations, we did not achieve this.