Music Similarity Analysis - Sprint2

Team (BS19-RO): Anton Buguev Mirna Alnoukari Elizaveta Kovanova

[Github repository]

Project Description

The algorithm will recommend a playlist based on a list of user's liked tracks using tracks' characteristics (chords, notes, rate, frequency, etc) and track's metadata (such as track's name, its artist, album, genre, etc.).

Implementation

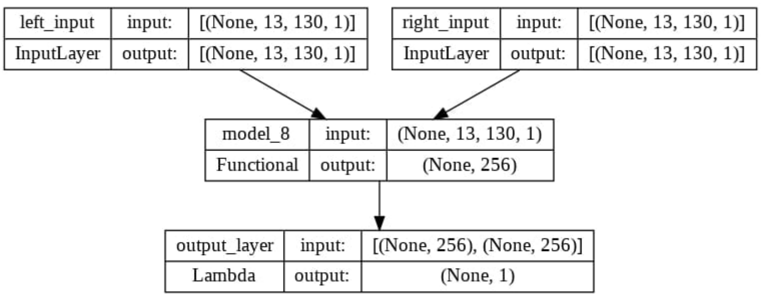
Dataset refactoring

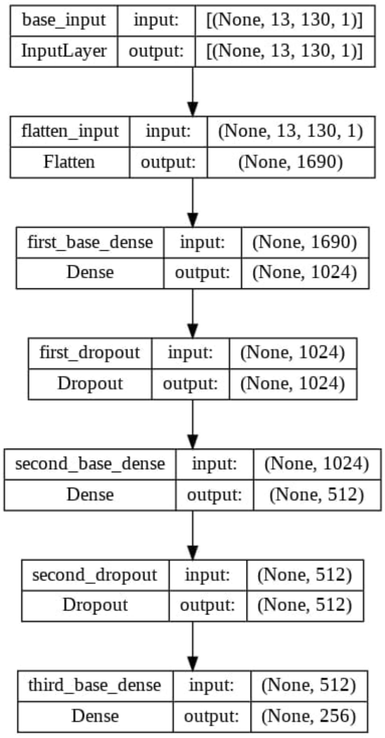
Since it is planned to create a UI, a need for quick and efficient filename getting arose. We have come to a conclusion that it would be easier and more time efficient to recreate and repeat the preprocessing of our dataset in the long run.

The new `mel_specs.json` dataset now also contains the corresponding filename for each MFCC segment. This is done so that when the program would suggest a most similar track to the some other track (chosen by the end user), the script would find the list of all similar segments and fetch them by name from the dataset directly, instead of wasting computational resources on finding the specific segment in the database to obtain its index and map it to the list of filenames.

Siamese Neural Network (SNN)

Siamese Neural Network model. This model takes 2 MFCC as inputs and as an output gives single value which represent the difference between inputs. The higher the value, the higher the difference between the inputs.

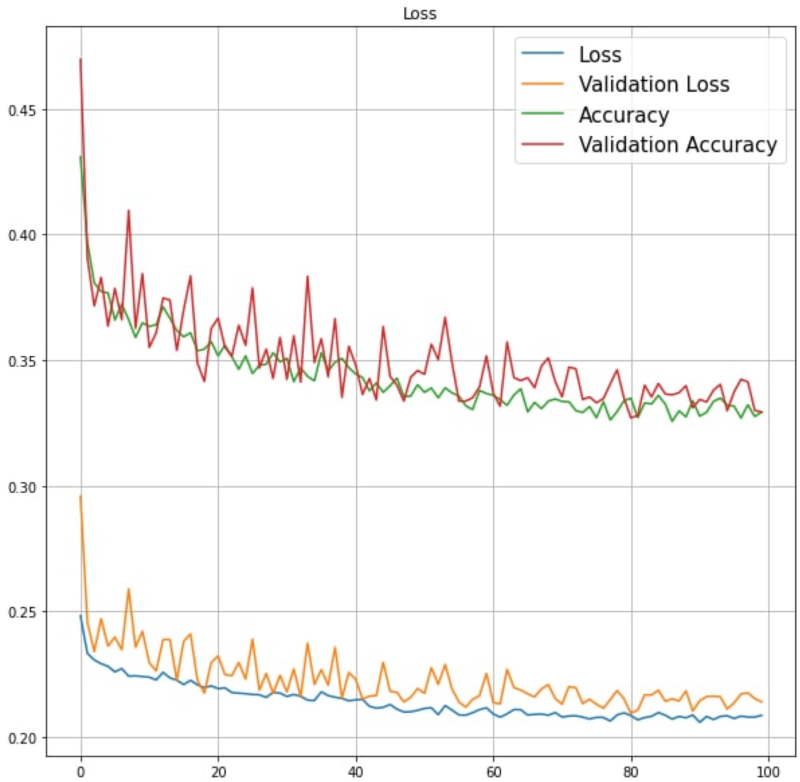




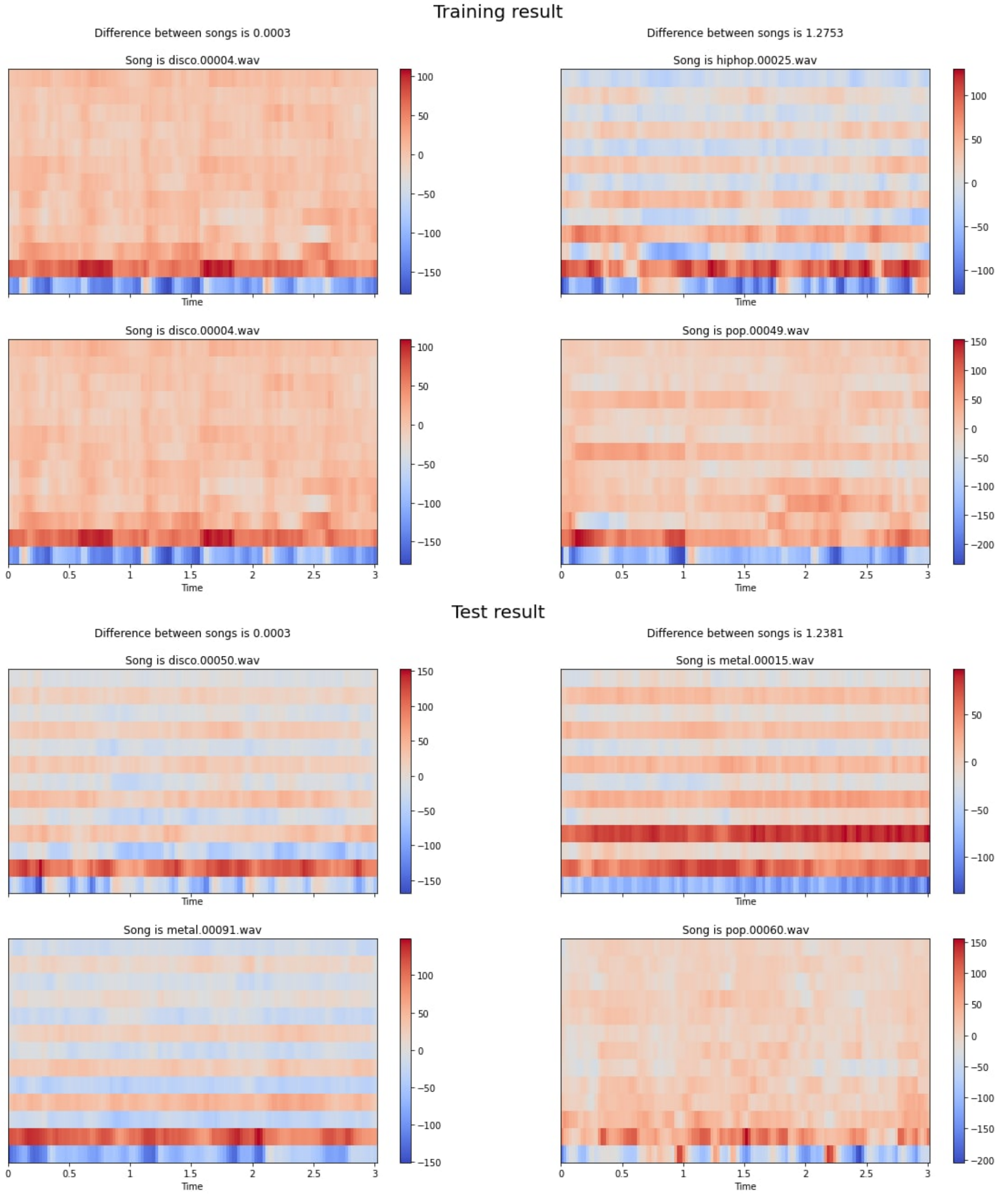
This is base Neural Network model that takes MFCC as an input and gives feature vector as an output.

Updates on SNN

1. Tuned the SNN, which decreased the loss value, however, accuracy is still only 0.5 and hopefully can be improved
2. Using updated dataset now we are able to display the most and the least similar songs and not just MFCC

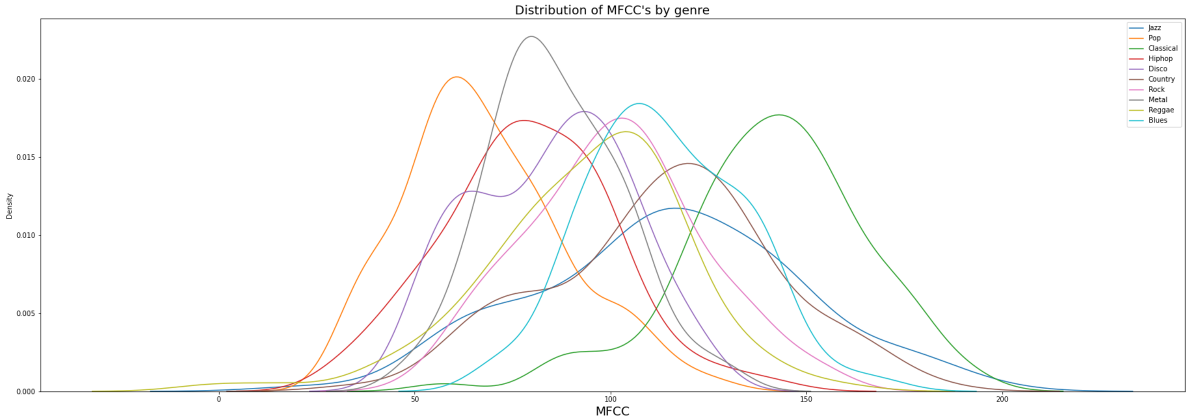


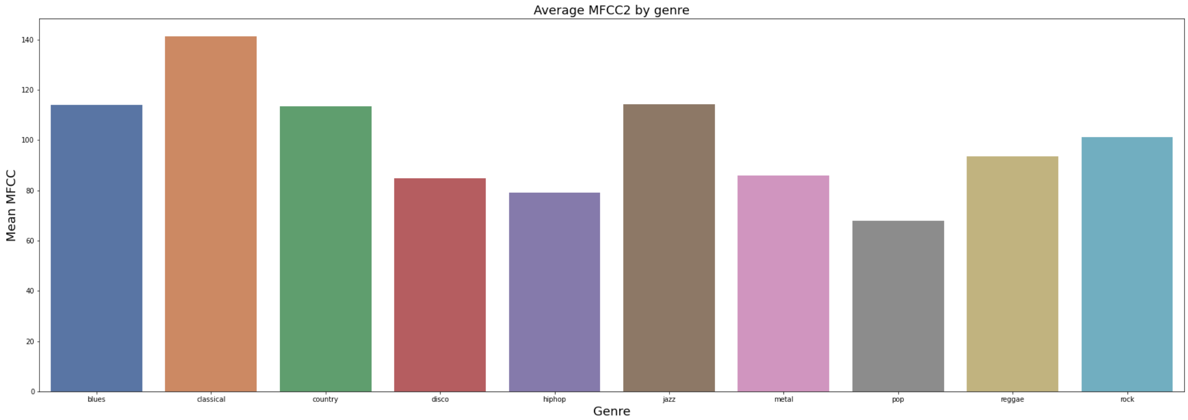
The following are Mel-frequency cepstral coefficients (MFCC) which represent a song fragment. During training the model analyzes and finds similarity between them. After that we can check that songs with small difference indeed have similar MFCC plots, and songs wit high difference have different MFCC plots.



Music Feature Extraction

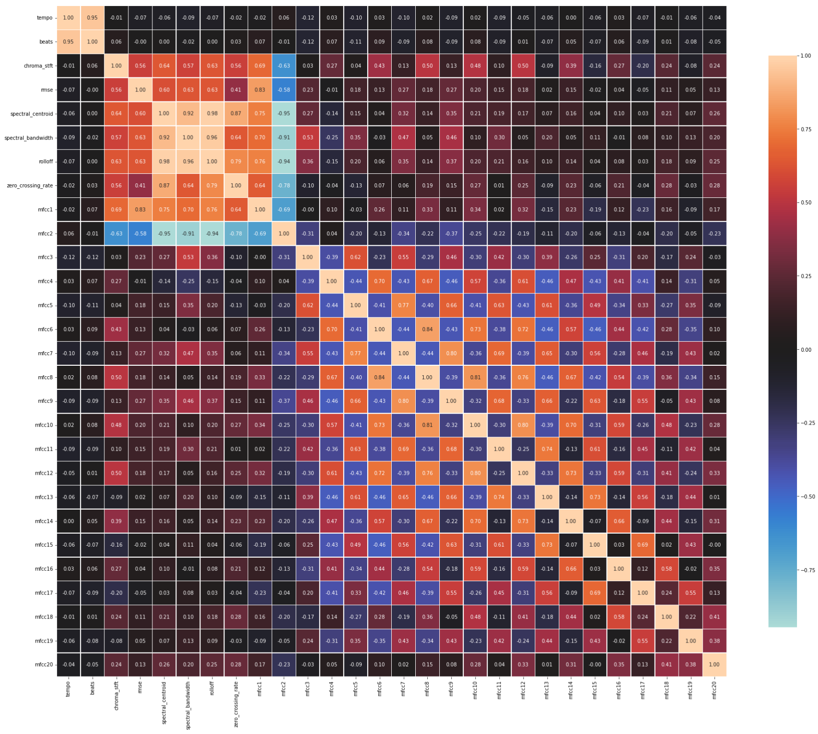
Since every song have many different features (tempo, spectral centroid, spectral bandwidth, rolloff, zero crossing rate, mfccs, etc.) it is important to understand how those features can be used in recommending system. The following figures show the distribution of tempos between different genres, and their means. Notice how the Disco tempo is distinctly different, with a peak close to 150 BPM. While Classical and Jazz are the most diverse, having less prominent peaks and a wider spread of tempos. The average tempo for all genres are very similar. The same can be done for the other features, we chose MFCC as they are more representative and different for each genre. Specifically MFCC2, the second coefficient of the Mel-frequency spectrum.





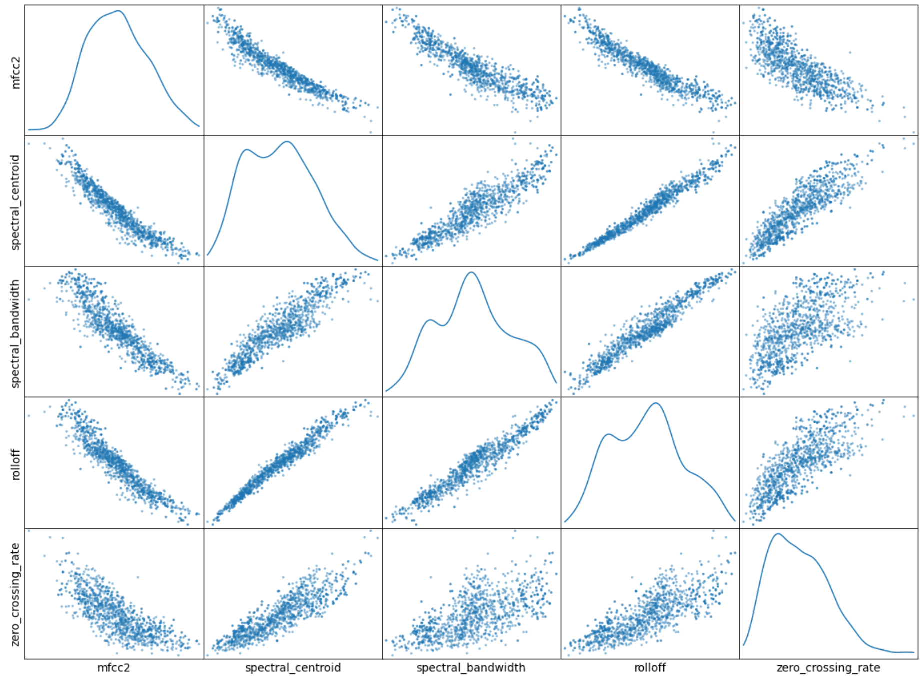
Features Correlations

Quantifies how close they are to a perfect linear relationship. This filtered list more clearly shows the features with the strongest positive correlations. Here we can see that rolloff is strongly correlated with both spectral centroid and spectral bandwidth. We can also see that there is also a strong correlation between spectral bandwidth and centroid.



Scattered Plots

These scatter plots effectively visualize the relationships between the highly correlated variables. Notice that some variables have negative, non-linear relationships with mfcc2.



Future plans

We plan to create a UI, where a song similar to the current one will be suggested. This system should work as follows:

- Obtain MFCCs for all segments of the current track
- Using the Neural Network obtain MFCCs of segments similar to all segments of the current track
- Compute Euler distance of each pair and define the error as mean value of Euler distances
- From these MFCCs extract filenames of corresponding tracks and make a list of all unique tracks
- Remove from list if the filename is the same as current one's
- Suggest the track (filename) with minimal error

Tasks Distribution

Dataset refactoring part was done by Elizaveta Kovanova, SNN updates were done by Anton Buguev, and Music features extraction was done by Mirna Alnoukari. All in all, planning, analyzing the results, and decision making was done by the whole team together.