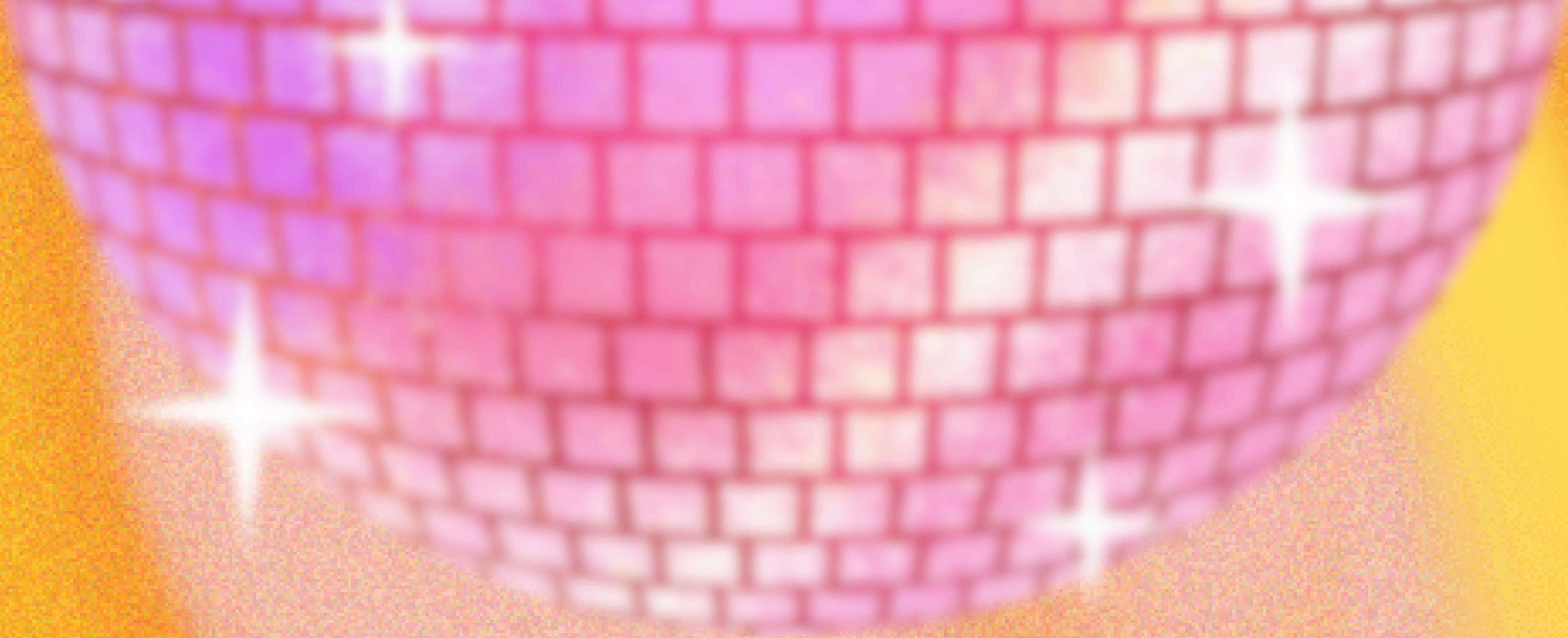

SPOTIFY RECOMMENDATION SYSTEM

A PROJECT BLOGG

12 JAN 2023

CLARA LENNARTSSON
SALMA GABOT
MOA BUREVIK

Background

There is a lot of research and ongoing studies on music's positive benefits and impact on people. Music has a lot of benefits on different levels and areas. For example there is research on the connection between musical activities and various aspects of health, both psychologically and physiologically. The role of music in emotional processing, positive aging and to fight or speed up the healing process after specific diseases like stroke or dementia.

Another way we can benefit from music is when exercising. Research in the field has consequently reported the enhancing effect of music on exercise performance. For example, listening to music before or during exercise may enhance handgrip strength, muscle endurance, velocity, and power. The factors that have been observed to be the most important for performance-enhancing are music tempo, volume and genre preference.
An enhancing-performance effect is commonly connected with faster tempo music (>120 bpm) and volume set to 70–80 decibels. Individual preference regarding music genre is also an important factor as well as lyrical content, harmony and melody.

So by knowing how music can affect and improve exercise performance, this project aims to create the optimal exercise playlist based on the most important factors for performance-enhancing (tempo (bpm) and individual preferences). With these features together with cosine similarity and euclidean distance for classifying music, a playlist based on recommended songs for the user and the exercise purpose will be created.

Reasearch question

This project aims to create a recommendation system that creates a customized playlist for the user based on recommended songs for the user and his/her exercise purpose.

Data collection

The data set used in this project was gathered from Spotify API. In order to use the Spotify Web API an app was created using Spotify for Developers. This allowed for Web API calls. Spotify Web API provides a wide range of functionality where one can request access to a given resource in order to retrieve data. Detailed profile information about the user such as tracks, artists and albums were gathered to get recommendations and create and manage playlists.

The dataset contains of the following features for each track:

- ID
- Name
- Album
- Album ID
- Artists
- Artists ID:s
- Track number
- Disc number
- Explicit
- Danceability
- Energy
- Key
- Loudness
- Mode
- Speechiness
- Acousticness
- Instrumentalness
- Valence
- Tempo
- Duration ms
- Time signature
- Year
- Release date
- Liveness

Training data

The training data was constructed by merging all songs from all playlists from one user's Spotify account. The merged playlist had about 3000 songs. This allowed for the system to be trained on a wide range of music and alo is a part of the users music preference. However, some problems occurred with the size and the amount of API calls when gathering the songs, therefore a limit on 100 songs was used and the songs were saved in a csv file.

Relying on a small dataset of only 100 songs most certainly limited the diversity of genres, artists and other features which could be the reason why the system makes narrow recommendations. The system might also have developed biases based on the limited data available by skewing the recommendations towards the specific genres etc, ignoring the user's potential interest in other genres or styles. This could also have made the system more susceptible to overfitting. By tailoring the recommendations to the limited training data the system could struggle when presented with new or unseen songs.

Test data

The test data was gathered from Kaggle and called Spotify 1.2M+ Songs. This data set was already a csv-file so the previous problem with the training data was easily avoided. The Spotify 1.2M+ Songs data set contained 1.2 million songs with different artists and genres and even though it was a big collection of data, the set was limited. This data set was last updated in 2020 and therefore lacked a large part of newer music. It was also limited to certain artists and songs and lacked some of the more popular artists, songs and genres. The data set however, contained all of the necessary features needed in the project research.

When the test set does not represent the full spectrum of user preferences, the evaluation results might not accurately reflect how well the system performs in real-world scenarios. Similar to biased training data, a biased test dataset can lead to skewed evaluation results where the system's performance might appear better or worse than it actually is because it's being evaluated on a limited subset of preferences.

Data processing

Principal component analysis (PCA) and important features

To be able to perform principal component analysis (PCA) the data was first scaled. By then applying PCA to the data, the number of features were reduced to the most important ones. The features with the highest weights according to the PCA were:

- Tempo
- Key
- Loudness

These features matched the factors that had been observed to be the most important for performance-enhancing, which was mentioned in the background. The exception was the “key” feature which appeared instead of musical preference. Individual preference regarding music genre was an important factor as well as lyrical content, harmony and melody. Unfortunately the “genre” feature did not exist as an audio feature in the Spotify API. Therefore a set of three other features were collected to compare with the result from using the PCA features. The other set of features (hereafter called DVS features) contained:

- Danceability
- Valence
- Speechiness

Features

The features used in this project describe different ways of categorizing a song. In short:

- Tempo describes the overall estimated tempo of a track in beats per minute.
- Loudness describes the overall loudness of a track in decibels.
- Key describes the key the track is in.
- Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
- Valence is a measure describing the musical positiveness conveyed by a track. A high value indicates that a track sounds more positive (e.g happy, cheerful, euphoric) while tracks with low value indicate that the track sounds more negative (e.g sad, depressed, angry).
- Speechiness detects the presence of spoken words in a track. A high value can be an example of rap music.

Method

PCA

Principal component analysis is a technique used to simplify a dataset. Since our songs contain a lot of features for each song the dataset needed to be compressed. By finding the eigenvalues and eigenvectors of the covariance matrix, it was possible to find the dimensions that had the strongest correlations in the dataset. The eigenvectors with the largest eigenvalues corresponded to the dimensions with the strongest correlations to the data.

To model the similarity between songs two different methods were used:

Method

Cosine similarity

The cosine similarity is a measure that calculates the cosine angle between two vectors. The similarity ranges between 1 and 0. The smaller the angle (near 0°), the higher the similarity. Since the essential thing is the angle and not the magnitude of the vectors, two vectors pointing to points far away from each other could still have a small angle and therefore a high similarity .

The formula for the cosine similarity:

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \, \|\vec{b}\|}$$

Method

Euclidean distance

The Euclidean distance calculates the distance between two points in the feature space, meaning measuring the distance between the feature vectors of the training data and all other songs in the data set. A short distance between vectors indicates a similar song.

The formula for the euclidean distance:

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Result

Euclidean distance

Using the Euclidean distance method with the PCA features (tempo, loudness and key) a playlist with 100 recommended songs in the specific bpm range was created. Using the Euclidean distance method with the DVS features (speechiness, danceability and valence) the system could find 24 suitable songs to recommend.
With the PCA features the songs gave an average value in euclidean distance with 1.67. With the DVS features the average value was 0.33.

Result

Cosine SImilarity

The cosine result with the PCA features (tempo, loudness and key) could create a playlist with 53 recommended songs in the specific bpm range while the cosine result with the other set of features (speechiness, danceability and valence) could only find 19 suitable songs to recommend.
With the PCA features the songs gave an average value in cosine similarity with 0.996. With the DVS features the average value was 0.8994.

The different range of recommended songs can be a consequence of test data limited selection of songs of popular artists.

Evaluation

As mentioned above the result when using Euclidean distance with PCA and DVS features differ. To objectively decide which was better then the other the similarity between the selected and the recommended songs were calculated using the features.

When using the Euclidean distance with the DVS features the distance was 0.33 which is low and indicates a good value. While with the PCA feature the distance was 1.67 which is very high and not a good value for the songs.
Therefore the result for the Euclidean distance had the best result when using the DVS features rather than the PCA features.

The result when using Cosine similarity and the two sets of features also differed and the same method was used for deciding which was better.

When using the Cosine similarity with the DVS features the similarity was 0.8994 which is a high value but the PCA features had a value at 0.996 which is better and higher.
Therefore the result for the Cosine similarity had the best result when using the PCA features rather than the DVS features.

Both Euclidean distance and Cosine similarity were good at recommending songs in the specific bpm range (150-170) and the containing features. However the songs did not match our individual preferences of genre and artist. This was likely due to the limited range of songs in the test data.

Conclusions

Using the DVS feature in the Euclidean distance method gave the shortest distance and therefore the best result. When using the Cosine similarity the PCA features gave a better similarity and therefore the best result.

Both methods were good at recommending songs in the specific bpm range and the containing features but the resulting songs did not match our individual preferences in music regarding genre and artist. This likely had to do with the limited data. By training the system on a larger set of training data the insights would be broader, aiding models to better understand complex patterns and generalize accurately. It would reduce overfitting, ensure robustness by covering diverse scenarios, and improve the models adaptability, resulting in more accurate and reliable predictions.

Broader test data would also ensure a more realistic evaluation of the models performance across different scenarios and validate its ability to generalize and adapt. It would help uncover weaknesses, ensure fairness, and comprehensively assess the model's reliability across various user preferences and situations. If this project were to be recreated, a larger set of training data and broader set of test data should be used.

Overall, the system created can recommend songs in the specified bpm range, and other features using both the Euclidean distance and Cosine similarity methods, resulting in a playlist suited for the type of training intended.