Mood Based Song Recommender

Bas Goossens **Fatemah Iqbal** **Kushagra**

Abstract

This paper assesses the music recommender system that we made for our final project. The data comes from a Kaggle dataset and is first filtered and then clustered using the K-means approach and visualized using PCA and t-SNE. User evaluations on the recommender can be found at the end of the paper.

1 Introduction

Recommender systems serve to to produce meaningful recommendations that may interest users. Song recommending algorithms are often already built into streaming services, such as Spotify, Apple Music, Deezer and YouTube Music. However, we've found these recommendations lacking when it comes to mood based recommendations. We wanted to create a recommender system that would give song recommendations based on the mood of the user and the specific vibe of the song(s) they input.

2 Related Works

Song, Dixon, and Pearce (2012) present a **content** based approach to music recommendation rooted in information retrieval and information filtering. That is, the model leverages the audio features and other key attributes from the items in the dataset to make similar recommendations. A suggested method was to use K-means clustering with acoustic features. An emotional model using *valence* and *arousal* was proposed to use acoustic cues to calculate the distances between songs and make recommendations based on similarity. (Kim, Kim, Park, Lee, & Lee, 2007) also propose an algorithm with a dynamic k-means clustering approach. Our project incorporates both of these methods, however we were unable to find **arousal** (how exciting or calming a song is) tags and instead used other features such as danceability and energy.

3 EDA

3.1 Dataset

The dataset was the most critical component in this recommender system. An extensive set was required, labelled with relevant attributes to provide essential information about each song, such as track ID, song name, genre, artist, etc. The following sections explore the procedure followed to curate the dataset used in this project.

3.1.1 Selection Criteria

In order to minimize processing time, a selection criteria was created to filter the dataset and reduce its dimensionality. As per the selection criteria, most non-English songs were filtered out so that the model could mainly be trained on English songs. This allowed us to train the recommender system on more popular and well known songs, as well as reduce the size of the dataset.

As songs would be recommended according to clusters, every song in the dataset had to be tagged with certain audio features (as described by the Spotify API). The Spotify API tags songs from the Spotify dataset with 13 attributes, namely: acousticness, danceability, duration-ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time-signature, valence.

The attributes for the dataset were narrowed down to energy, valence, danceability, accousticness, liveliness, loudness, mode, speechiness and tempo as these features can provide information related to emotions or mood. Some of the features were dropped as the correlation matrix of all the chosen features indicated that tempo was not significant as it was negatively correlated to other features.

A description of the few of the main features as provided by "Spotify for Developers":

energy : Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity

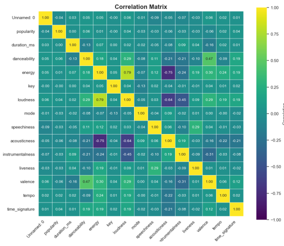

Figure 1: Correlation Heatmap

and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

danceability : 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. A value of 0.0 is least danceable and 1.0 is most danceable.

valence : A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)

3.1.2 Data Cleaning and Filtering

A dataset with all 13 attributes was acquired from Kaggle. It initially contained 114,000 songs with 114 genres, which we filtered to 69,000. 38 genres were dropped on the basis of our selection critera, i.e if any of the songs were missing the relevant attributes or if the genre was from a different language. For example: kpop, jpop, mandopop.

Columns like 'track_id', 'album_id', and 'album_ release_date' were deemed unnecessary for the core analysis and were subsequently removed. Missing values were identified and handled appropriately. Since the dataset did not contain significant missing data, simple imputation methods or row deletions were sufficient.

3.1.3 Summary Statistics

Category	**Value**
Songs: Database	68,999
Genres	69
Audio Features	3
Dataset Attributes	8

3.2 Correlation Analysis

We conducted a correlation analysis to examine these relationships. A correlation matrix was generated to visualize the relationships between the numerical features. This matrix helps in identifying which features are positively or negatively correlated with each other. As discussed earlier in the selection criteria section insignificant ones were dropped.

Few of the things considered:

Energy and Loudness: There is a strong positive correlation between energy and loudness, suggesting that louder tracks are generally more energetic.

Speechiness: This feature shows relatively low correlations with other audio features, highlighting its distinctiveness in capturing the spoken word content of tracks.

3.3 Standardization:

To ensure the audio features are on the same scale and have a mean of 0 and a standard deviation of 1, we applied StandardScaler from sklearn. The scaled features included valence, acousticness, danceability, energy, liveness, loudness, mode, speechiness, and tempo.

4 Clustering

To identify patterns and group similar songs, we applied K-Means clustering. The optimal number of clusters was determined using the elbow method, plotting the inertia against the number of clusters. This method involved plotting the within-cluster sum of squares (WCSS) against the number of clusters.

The plot suggested an elbow point at 7 clusters, which we used for the K-Means algorithm. By observing the point where the rate of decrease in WCSS slows down, we were able to identify the optimal number of clusters, which signifies the point where adding more clusters does not significantly reduce the WCSS.

Subsequently, the K-Means clustering algorithm was chosen for its simplicity and efficiency, particularly suitable for handling large datasets. The model training involved testing various values of K to find the optimal number of clusters that best represent the underlying structure of the data. Notably, features with high within-cluster variance and low between-cluster variance were deemed crucial for cluster interpretation.

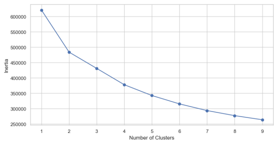

Figure 2: Elbow Graph

For visualization purposes, techniques like Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) were utilized to reduce the dimensionality of the dataset. This reduction allowed for the creation of scatter plots and other visualizations to represent the clusters in the reduced feature space, aiding in the interpretation and understanding of the clustering results.

4.1 Recommendation

The provided code in the notebook facilitates song recommendations through a multi-step process. Initially, it retrieves and preprocesses song data from the Spotify API, extracting relevant audio features. Subsequently, it employs KMeans clustering to group songs based on their audio feature similarity, using a pipeline that scales the features and performs clustering. Following the clustering, it calculates a mean vector representing the average audio features of input songs provided by the user. Then, using cosine similarity, it identifies songs with similar audio features to the mean vector, thus generating a list of recommended songs. Finally, it excludes the input songs from the recommendations and presents the recommended songs to the user. This approach leverages both clustering and similarity techniques to provide personalized song recommendations tailored to the user's preferences, offering a diverse selection of songs that align with their musical tastes.

5 Evaluation and Analysis

The evaluation was conducted through a structured approach consisting of two rounds, aimed at assessing the system's performance in providing personalized music recommendations.

Round 1: Similarity Rating Participants were presented with pre-selected songs tailored to two mood profiles: happy and sad. These profiles were alternated among participants. Participants listened to the recommended songs and then rated their similarity to the initial song provided.

Ratings ranged from 6 to 9, indicating varying degrees of satisfaction with the input selection. The average rating across all participants was 8.2, suggesting a generally positive perception of the songs provided.

Round 2: User Input Recommendations Participants were asked to provide 2-3 songs that they considered similar. The system then generated recommendations based on the provided input, and a random recommendation was played for the participant to assess. Participants were asked to rate the overall similarity of the recommended song to their input selections.

Participants rated the system's performance on a scale from 5 to 9. The average test performance rating was 7.8, indicating that the system effectively provided recommendations aligned with the user's mood.

Feedback and Assessment Following the two rounds of evaluation, participants were asked to provide overall feedback on their experience with the evaluation procedure and the music recommendation system.

Participants rated their satisfaction with the recommended songs and score varied from 6 to 10. The average user satisfaction rating was 7.7, indicating moderate to high levels of satisfaction with the music recommendations provided by the system.

Overall Evaluation:

Participants provided feedback and comments regarding their experience with the music recommendation system. The basic music recommendation system demonstrated promising performance in aligning recommendations with user mood and eliciting moderate to high levels of user satisfaction. However, there are areas for improvement highlighted by participant feedback, such as refining the recommendation algorithm to better address the intended mood and reducing repetitiveness in song selections.

6 Future Implematations

In a future implementation, with additional time and resources, we would like to explore broader array of datasets to create a tailored dataset focused specifically on analyzing the evolution of music preferences for individual users. Using this information, we could builda a recommendation systems that suggest music based on a user's evolving profile.

Furthermore, for evaluation purposes, we propose extracting lyrical text from the music data and performing sentiment analysis to identify the mood conveyed by the songs. By doing that we could segment songs into mood-based clusters, offering personalized recommendations that align with their current emotional state.

Additionally, an idea suggested by a participant involves offering users the option to fine-tune specific features such as 'danceability' or 'acousticness'. By allowing users to adjust these features according to their preferences, we can offer a more tailored music listening experience that aligns with individual tastes and preferences.

7 Conclusion

In conclusion, the music recommendation project was an exciting journey of data exploration, preprocessing, and clustering techniques. Through exploratory data analysis, we gained insights into the structure of the music dataset, enabling effective data preprocessing and feature engineering. Leveraging clustering algorithms such as K-Means, we partitioned the dataset into clusters, enabling us to offer tailored music recommendations based on users' preferences.

In addition to the technical achievements, it's crucial to acknowledge lessons learned throughout this project journey. Foremost among these was the collaborative spirit within the team. Working together, we seamlessly divided tasks, and engaged in fruitful discussions. Moreover, the process of brainstorming and considering various ideas was important in shaping our project, many ideas were considered and dropped due to time constraints.

Looking ahead, there are numerous opportunities for further refinement and enhancement of the recommendation system. By incorporating sentiment analysis of lyrical text and fine-tuning features based on user preferences, we can offer more nuanced and personalized recommendations that resonate with users on an emotional level.

References

Song, Yading, et al.

1 Jan. 2012. *A Survey of Music Recommendation Systems and Future Perspectives.*

Kim, Dongmoon, et al.

Dec. 2007 *A Music Recommendation System with a Dynamic K-Means Clustering Algorithm.*International Conference on Machine Learning and Applications (ICMLA 2007), Dec. 2007, https://doi.org/10.1109/icmla.2007.97. Accessed 2 May 2022.

Spotify. "Web API Reference — Spotify for Developers.

Developer.spotify.com, developer.spotify.com/documentation/web-api/reference/get-audio-features.