Music Recommendation Systems

Zoh Tia, Tram Nguyen, Tsion Sherbeza, Maria Shiferaw

05/02/2024

Abstract

This project focuses on developing a content-based music recommender system based on acoustic similarity of musical compositions. Two approaches are considered: acoustic features analysis and machine learning methods. The aim is to improve the recommender system's results and enhance information retrieval tasks in music.

Introduction

In the digital era, music is one of the most popular entertainment sources, offering a platform for human creativity to convey ideas and emotions. The convenience of modern technology, particularly smartphones equipped with offline and online music, has made music easily accessible for the user. However, the abundance of digital music available today can overwhelm users, leading to information overload and fatigue when attempting to navigate through enormous libraries. Therefore, it is very useful for the audio streaming industry to develop a music recommender system that can search music libraries automatically and suggest songs that are suitable for users to improve the user experience and screen time. To build our recommendation system, our approach is a content-based music recommendation system based on the acoustic similarity of the music present in our database. For example, our model predicts the likelihood of a song being liked by the user and recommends songs with high predicted likelihood and cosine similarity to the user's input.

Data Preparation

Data Overview

The dataset used for this project is from Kaggle `https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs` which contains audio statistics of different tracks on Spotify. The data has various features of songs such as danceability, energy, loudness, speechiness, and more. Songs released from 1956 to 2019 are included from some notable and famous artists like Queen, The Beatles, Guns N' Roses, etc. This dataset is used for different music recommendation systems. In this project, this dataset is used for a user input based recommendation system.

Data Cleaning, Preprocessing, and feature selection

Data Cleaning

The Spotify data consistent of 32,833 entries, each representing a different music track on Spotify, with 24 different attributes covering details such as track ID, track name, track artist, as well as various music features like how danceable or loud the tracks are.
The first important step in preparing the data for analysis is inspect the dataset to find and fix any issues. During this examination, we found out some missing values in some column such as track_name, track_artist and track_album_name, with 5 missing entries each. To keep the data clean and reliable for further analysis, we decided to remove these entries with missing values. This cleanup helped prepare the dataset properly for the next stages of analysis.

The next important step in preparing the data was to find and remove any duplicate records. This is crucial to make sure the results of any analysis are not biased. In our case, we discovered many duplicates, specifically 6,599 entries based on the track_name and track_artist. To refine the dataset, we kept only the first occurrence of each duplicate. This make sure every entry on our dataset was unique, improving the dataset's quality and trustworthiness for more indept analysis.

After cleaning up duplicate and missing data, we made dataset more concise by removing unnecessary columns such as "Unnamed:0" and "track_album_id". This step simplified the daa and made processing faster for future tasks, letting us focus on the most important aspects of the data.

With a cleaner dataset, we then conducted detailed analysis of differ music features. It was important to identify and handle any outliers-data points that are much different from most of the data. We used boxplots and a technique called the Interquartile Range (IQR) method to find these outliers of music features.

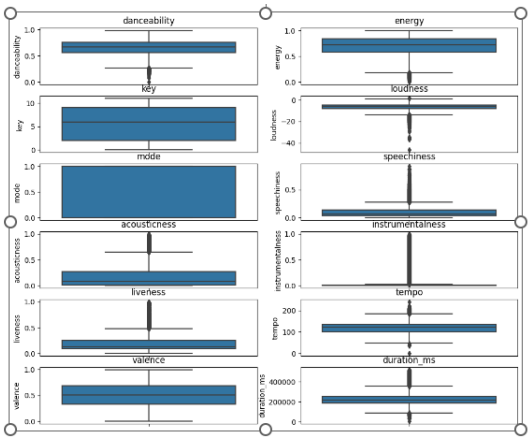

Figure 1: Outlier Distributions

The image presents a series of box plots that illustrate the variability and unique characteristics of various musical features, including danceability, energy, loudness, speechiness, instrumentalness, tempo, duration, acousticness, liveness, valence, key, and mode. These plots effectively demonstrate the distribution parameters such as the median, quartiles, and outliers, offering a visual representation of data spread.

The plots for danceability and energy reveal that values for most songs are clustered around the median. However, outliers are present, particularly in songs with markedly low danceability and energy, such as those in ambient or certain jazz genres, highlighting their distinctiveness.

The loudness feature shows a broad range of values. Some tracks are significantly quieter than most, potentially due to differences in recording techniques or production processes used before release.

Speechiness in songs generally registers low, with few exceptions. Outliers in this category typically include tracks with a higher proportion of spoken words, like podcasts or rap music, indicating a deviation from the norm.

The box plots for acousticness and liveness display outliers that may represent tracks utilizing natural soundscapes or live recording methods. High liveness values could suggest recordings that capture audience noise and the ambiance of live performances.

Instrumentalness varies, with some tracks exhibiting high values, indicating no vocal content. These are often found in genres like classical music, electronic, or jazz.

Variations in tempo are also noted, with outliers indicating extremely fast or slow tracks. Fast tempos are often associated with genres such as speed metal or electronic dance music, while slower tempos might be characteristic of ambient or orchestral music.

The duration feature highlights outliers in song length, typically seen in tracks significantly longer than average, such as extended live performances or progressive rock songs.

Lastly, the features of valence, key, and mode do not exhibit any notable outliers in the box plots, suggesting a more uniform distribution across these musical aspects.

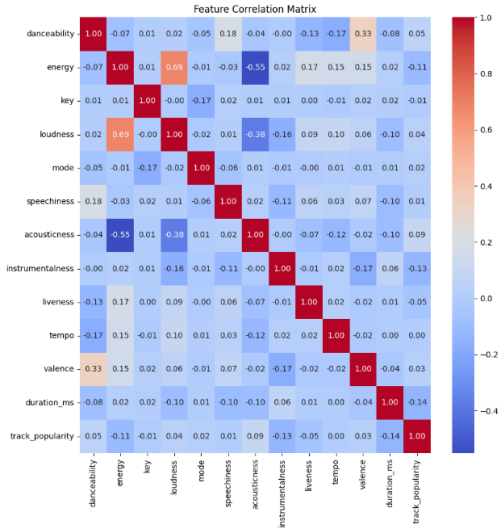

Figure 2: Correlation Graph

The image above illustrates a correlation matrix that quantifies the relationships among various musical features, with values ranging from -1 to 1. A correlation of 1 indicates a perfect positive relationship, -1 denotes a perfect negative relationship, and values near 0 suggest negligible or no relationship. This matrix is instrumental in analyzing music data effectively, enabling strategists to model data with greater precision.

The matrix reveals strong positive correlations between certain features, such as energy and loudness, where a correlation coefficient of 0.69 indicates that songs with higher energy typically exhibit increased loudness. This is intuitive, as louder music often feels more energetic. Another positive relationship exists between danceability and valence, suggesting that songs which are more danceable usually also feel happier. This trend is common in genres like pop and dance music, known for their lively beats. Additionally, the correlation between the release year and loudness of 0.28 reflects that newer songs tend to be louder, likely a result of advances in music technology that enhance the robustness and fullness of sound.

Conversely, the matrix also identifies significant negative relationships. For example, acousticness and energy have a correlation of -0.55, indicating that songs with a more acoustic and natural sound tend to be less energetic, aligning with genres such as folk or classical music. Acousticness and loudness show a negative correlation of -0.38, pointing to the tendency for acoustic songs to be quieter. Moreover, a correlation of -0.31 between song duration and release year suggests a trend towards shorter songs in recent releases, possibly because shorter tracks perform better on music streaming platforms, which favor quick and catchy songs.

Other interesting points emerge regarding features like speechiness and instrumentalness, which exhibit minor negative relationships with several other music features. For instance, tracks with more spoken words, such as podcasts or spoken word tracks, tend to diverge from typical music tracks in terms of loudness and overall sound. A slight negative correlation between instrumentalness and acousticness implies that many instrumental tracks might include genres like electronic music, which often do not use acoustic elements.

Regarding track popularity, small negative correlations with features like acousticness and energy suggest that very acoustic or mellow songs might not align well with mainstream musical tastes. Conversely, slight positive correlations with loudness and danceability indicate a preference for louder and more danceable songs. The negative correlation with song duration reinforces the preference among listeners for tracks that are concise and easy to enjoy.

This correlation matrix is crucial for enhancing music recommendation systems, particularly those focusing on song popularity. By examining key features linked to a track's popularity, such as its loudness, danceability, and duration, the system can better align recommendations with current listener preferences and the latest trends in the music industry. Understanding which features are less liked allows the system to avoid suggesting those tracks, thus enhancing user satisfaction. By integrating details such as release dates and social media performance, the listening experience becomes more personalized and engaging, potentially increasing user retention and loyalty. This

data-focused approach ensures that the recommendation system remains flexible and effective in matching individual tastes with popular music.

Preprocessing

Following the data cleaning, we had to deal with the outliers. We choose the winsorization method to deal with the outliers and replace the outliers. The Winsorization method uses a wisorized mean to replace the outliers. This means replacing the smallest and largest values of the dataset with the observations closest to them. In our case, we chose the domain to be within the 5th percentile to the 95th percentile. Based on that, anything above the 95 percentile will be replaced with the largest value and anything below the 5th percentile will be replaced with the smallest value.

We also converted the date of release to only the year of release because that is more relevant to the recommendation system rather than the month and date the songs were released.

Feature Selection

In the feature selection section of this technical report, specific musical features were chosen for analysis based on their distinct contributions to the understanding of a song's characteristics and their potential impact on its popularity. These features are grouped based on their relevance to various aspects of musical theory and listener perception, which are crucial for building a sophisticated music recommendation system.

Danceability, Energy, and Valence: These features are critical indicators of a track's overall feel and how it is enjoyed in various settings. Danceability reflects the ease with which a listener can dance to a track, which is pivotal in settings where movement to music is encouraged, such as parties or personal workouts. Energy denotes the intensity and activity level of a song, influencing how a track can invigorate or relax the listener, directly affecting its usage in different social and personal contexts. Valence measures the musical positiveness conveyed by a track; songs with high valence typically sound more positive and are often preferred in happy or uplifting scenarios.

Loudness, Tempo, and Duration: These features directly influence the energy and dynamics of a song, shaping the listener's emotional and physiological response. Loudness impacts the perceived energy level and can determine the suitability of a track for various listening environments, from intimate settings to large venues. Tempo dictates the speed at which the music is played, influencing how listeners experience rhythm and pacing—faster tempos generally energize listeners, while slower tempos are more calming. Duration affects the listening experience by defining the length of exposure to a single musical idea; longer tracks might be featured in more contemplative settings, whereas shorter tracks might cater to commercial radio or playlists designed for brief listening sessions.

Speechiness, Acousticness, and Instrumentalness: These features are instrumental in differentiating between types of songs. Speechiness detects the presence of spoken words in a track, distinguishing between music that contains significant speech, like rap or spoken word, and more melodic or instrumental tracks. Acousticness measures how much of a track sounds acoustic, as opposed to electronically produced; this quality can influence a listener's perception of authenticity and warmth. Instrumentalness identifies tracks that do not contain vocals, which are crucial for contexts where lyrics might be a distraction, such as during studying or in certain public environments.

Key and Mode: These features relate to the emotional aspects of a song. The key in which a song is played can affect the sonic quality and mood, while the mode (major or minor) can significantly influence the emotional tone of the music. Major modes are typically associated with happier, brighter sounds, whereas minor modes often evoke sadness or melancholy. Understanding these elements helps in predicting emotional responses to music, which is crucial for tailoring recommendations based on mood or emotional state.

Release Year: The inclusion of the release year as a feature is based on its relationship with nostalgia and the evolving trends in music over time. Songs from specific years or decades can evoke memories and feel-

ings of nostalgia, significantly affecting their popularity. Furthermore, musical styles and production techniques have evolved, so the release year can also provide context about the sound quality and stylistic trends of the period.
Collectively, these features form a comprehensive dataset that allows for nuanced analysis and prediction of song popularity and listener preferences. By selecting these specific features for the model, the recommendation system can leverage detailed insights into how various attributes of a song interact with listener experiences and preferences, ultimately leading to more accurate and personalized music recommendations. This tailored approach ensures that the system remains adaptable to individual tastes and responsive to the dynamic nature of music consumption.

Machine Learning Algorithms

For our recommendation system, we chose different machine learning algorithms. The algorithms we chose and the reasons we chose them are:

Decision Tree:
In the development of our music recommendation system, we employ a predictive model known as decision trees. This model utilizes musical features such as 'danceability', 'energy', and 'loudness' to forecast the potential popularity of various songs. The advantage of decision trees lies in their robustness; they perform efficiently without requiring normalization of feature scales. Unlike many algorithms that compute distances between data points, decision trees operate by comparing feature values and determining splits based on specific criteria, thereby ensuring that the scaling of features does not influence the outcome. Furthermore, decision trees exhibit exceptional resilience in managing outliers within the dataset, operating on straightforward "if-then" logic that remains unaffected by extreme data values.

To enhance the predictive capability of our decision tree model, we implement a technique known as hyperparameter tuning through grid search. This process involves the meticulous adjustment of parameters such as 'max-depth', which controls the tree's maximum depth; 'min-sample-leaf', which sets the minimum number of samples each leaf must contain; and 'min-sample-split', which defines the minimum number of samples required to initiate a new branch. These adjustments are critical for optimizing model accuracy, preventing overfitting, and ensuring reliable predictions concerning song preferences, thereby significantly improving the effectiveness of our music recommendation system.

Random Forest:

In this sophisticated implementation of our music recommendation system, we employ a Random Forest classifier to predict user preferences based on a range of musical features. Initially, the 'track_album_release_date' within our song dataset is converted to datetime format to extract the 'release_year', which is then included as a feature to augment our model's predictive capability.
The features selected for modeling exclude 'track_popularity' to ensure unbiased predictions, forming a dataset designated as 'X_r'. A binary target is generated from 'track_popularity', classifying songs as popular or not based on whether they exceed the 75th popularity percentile. The dataset is split into training and testing subsets with a 20% test allocation, ensuring a stratified distribution of the binary target.
Prior to model training, missing values within the dataset are handled through mean imputation, preparing the data for a robust training process. The Random Forest model, configured with 100 estimators, is trained on this preprocessed data. Following the training, the model's efficacy in classification is assessed through standard metrics including accuracy, precision, recall, and F1-score, with detailed results provided in a classification report.
For user interaction, the system prompts for the title and artist of a song. If the song is located within the dataset, the system first re-applies the imputation to ensure data integrity. It then uses the trained model to predict the likelihood of user preference across the entire dataset, appending these probabilities back to

the main dataset.
To offer personalized recommendations, cosine similarity between the user-specified song and all songs in the dataset is calculated. Songs are then ranked according to both their predicted likelihood of preference and their similarity score, with the top 20 songs being selected. These recommendations are presented to the user, showcasing songs that align closely with their expressed preferences based on both predictive likelihood and feature similarity.
In addition, these top recommendations are saved to a CSV file, providing a tangible output for further analysis or external use. If the user's song is not found, the system provides a prompt notification, maintaining clear and effective communication.

K-Nearest Neighbors (KNN):
In the outlined framework for our music recommendation system, a set of musical attributes named 'correlated_features' is established, encompassing characteristics such as 'danceability', 'valence', 'energy', etc., while omitting 'track_popularity'. The dataset is bifurcated into features ('X_knn') and binary labels ('y'), where the labels indicate whether a song's popularity surpasses the 75th percentile within the distribution of 'track_popularity'.
The dataset is subsequently partitioned into training ('X_train', 'y_train') and testing ('X_test', 'y_test') subsets, adhering to an 80/20 split ratio. A K-nearest neighbor (KNN) classifier is then calibrated using the normalized training data, with the number of neighbors ('k') configured at 181. Post-training, this model is deployed to predict the classification labels of the test set, and these predictions are leveraged to compute key performance metrics such as accuracy, precision, recall, and F1 score. These metrics are integral in assessing the classifier's efficacy in distinguishing between songs categorized as popular or not, according to the pre-established popularity threshold.
Interactive user engagement is facilitated through a prompt requesting the title and artist of a song. Upon locating the song within the dataset, the system estimates the likelihood of the song being favored (popularity prediction) employing the KNN model. Cosine similarity metrics are computed to gauge the resemblance between the feature set of the selected song and all other songs in the dataset. Songs are ranked based on their similarity scores, with the top 10 most akin songs curated as recommendations.
These recommendations are subsequently presented to the user, offering insights into similar songs that might align with their musical preferences based on the initially inputted song. Should the user's song be absent from the dataset, an error message is displayed, indicating the non-availability of the song within the system.

Long-short Term Memory (LSTM):
In this advanced framework for our song recommendation system, we employ a Long Short-Term Memory (LSTM) neural network model to predict user preferences based on musical features. The model is constructed using the Keras Sequential API, incorporating two LSTM layers with 100 units each, interspersed with Dropout layers set at 30% to prevent overfitting. The output layer consists of a Dense layer with a softmax activation function tailored for binary classification. Optimization is performed using the Adam optimizer with a learning rate of 0.001, and the model is compiled with categorical crossentropy as the loss function, focusing on accuracy metrics.
The LSTM model undergoes training over 50 epochs with a batch size of 32, using the training dataset. Following training, predictions are made on the test dataset to ascertain the model's effectiveness in classifying songs based on user-defined popularity thresholds. The model's performance is quantitatively evaluated through metrics such as accuracy, precision, recall, and F1-score, providing a comprehensive overview of its classification prowess.
For user interaction, the system prompts the user to enter the title and artist of a song, which it then attempts to match within the dataset by converting all entries to lowercase for case-insensitive matching. If the song is found, the system calculates the probability of the song being liked using the trained LSTM model. Additionally, it computes the cosine similarity between the features of the user-selected song and all songs in the dataset to identify those with similar

characteristics.
If the song exists in the dataset, the system filters out the user's input from the recommendations to ensure relevance and uniqueness. Songs are ranked based on their predicted likelihood of being liked, and the top 10 unique songs are then displayed to the user, providing tailored recommendations that enhance user experience. In the event the song is not found in the dataset, the system promptly informs the user of this, ensuring clear communication.
After selecting the algorithms, we decided to focus on the accuracy, precision, recall, and F1 score of the algorithms and their results for the dataset we chose to conduct this project. Based on the results of the algorithms, we choose the one with the best results and build a recommendation system for the dataset. The results will be shown in the next section of the paper.

Results

In this section of the paper, we will discuss the results we obtained from the algorithms we chose earlier and build a recommendation system using the model with the best results. We will discuss the results of each algorithm separately in different subsections.

Decision Tree

As stated previously, one of the models we chose is the Decision Tree for our model.

Datasets	Accuracy	Precision	Recall	F1-score
Before Cleaning	73.08%	42.55%	16.47%	23.75%
After Cleaning	73.10%	42.63%	16.47%	23.77%

Table 1: Decision Tree results for the recommendation system

From what we can see from the table above, Decision tree does not have that much of a change when it comes to the results before or after dealing with the outliers because it does not depend on them. We can also see that the Recall is low suggesting that it is not identifying the correct data points.

Random Forest

Another one of the models chosen for the recommendation system was Random forest.

Datasets	Accuracy	Precision	Recall	F1-score
Before Cleaning	86.0%	85.0%	55.0%	67.0%
After Cleaning	86.0%	85.0%	55.0%	67.0%

Table 2: Random Forest results for the recommendation system

From what we can see from the table above, Random Forest did not have a difference in the results before and after dealing with outliers. It did have a high accuracy of 86% with a good score for precision and recall making it the most accurate recommendation system when it comes to suggesting music to users.

K-Nearest Neighbors

For KNN, being one of one of the models chosen for the recommendation system, the classification results are listed below.

Datasets	Accuracy	Precision	Recall	F1-score
Before Cleaning	75.0%	38.0%	50.0%	43.0%
After Cleaning	75.0%	38.0%	50.0%	43.0%

Table 3: KNN results for the recommendation system

From what we can see from the table above, KNN has an accuracy of 75% which is good but the precision being low pulls down the F1-score. This makes it more fit that Decision Tree but less fit than Random Forest when it comes to this dataset and the recommendation system.

LSTM

For LSTM, we chose the activation function softmax with the addition of dense and dropout layers. The accuracy, recall, precision, and F1-score for the three different datasets are listed in the table below.

Datasets	Accuracy	Precision	Recall	F1-score
Before Cleaning	74.9%	55.0%	29.5%	5.6%
After Cleaning	75.2%	52.8%	16.4%	25.0%

Table 4: LSTM results for the recommendation system

From what we can see from the table above, LSTM has an accuracy score of 75%. There is not that much of a change before and after dealing with outliers either. However, it has the lowest Recall score out of all the models chosen making it the less fit model when it comes to making the music recommendation system using this dataset using the data cleaning and preprocessing measure that we chose.

Conclusion

In conclusion, Our project encompasses a broad use of machine learning algorithms to come up with the best song recommendation system based on 10 Top Similar Songs. The models used were KNN, Random Forest, Decision Tree, and LSTM, each with its implementation performance metrics.

High accuracy indicates that the model performs well across both classes of popularity. A high precision indicates that most of the songs recommended as popular by the model were popular, which is crucial for maintaining user trust in the recommendations. A high recall would mean that the model successfully identifies most of the popular songs, thus reducing the risk of missing out on potentially popular recommendations. A high F1 score indicates that the model has a robust overall performance with both high precision and recall.

Balancing precision and recall is crucial, ensuring users receive popular song recommendations that are likely to be relevant (precision) while minimizing the chances of missing other potentially popular songs (recall).

The Random Forest model consistently shows superior performance in accuracy, precision, and F1-Score, highlighting its robustness and suitability for handling binary classification in complex datasets like songs. Its ability to manage feature interactions and dependencies likely contributes to its strong performance. The recommendation system successfully integrates classification and similarity measures to provide targeted suggestions, enhancing user engagement and satisfaction.