Spotify Recommendation System

AUTHORS

Sai Donepudi, Mateo Lauzardo, Debora Tantu

AFFILIATIONS

AI4ALL

How can the process of creating AI/ML solutions amplify or mitigate bias in our group's project?

01. Introduction

AI-powered music recommendation systems have been evolving rapidly in recent years. Spotify, for example, introduced its AI-powered recommendation feature in 2021, significantly improving its ability to suggest music based on users' preferences. The rise in demand for personalized music recommendations can be observed across various platforms like Spotify, SoundCloud, and others, which have adopted similar features. This demand stems from the ability to discover new music effortlessly and enjoy a personalized listening experience.

For this project, the team chose the dataset "Top 10000 songs on Spotify (1950-now)" to create a recommendation system. The system aims to provide personalized song suggestions based on various audio features like "Danceability," "Tempo," and "Volume." The main focus is to enhance user experience by generating similar song recommendations tailored to individual preferences.

02. Objective

The main objective of this project is to develop a recommendation system using unsupervised learning, specifically through the K-Means clustering algorithm. By leveraging key audio features such as danceability, tempo, and volume, we aim to create an AI-powered system that recommends songs similar to the ones a user inputs. The system is designed to facilitate music discovery, enhancing the user experience by providing personalized suggestions. Additionally, we aim to understand how AI/ML algorithms can mitigate or amplify biases in recommendations, especially considering that biases can be inherent in the dataset itself. The goal is to ensure that the recommendation system provides diverse and fair suggestions that represent a broad range of music styles and genres.

Citations

- Unsupervised Learning: K-Means Clustering
- Unsupervised Learning - MLQ.ai
- Unsupervised Clustering: A Guide
- Unraveling Spotify's Music Universe: A Clustering Analysis Approach
- Unsupervised learning
- Song Discovery with K-Means Clustering: A Music Recommendation System
- K-Means Clustering Algorithm - Anallytics Vidhya Unsupervised Learning Explained Using K-Means Clustering

03. Methodology

The core methodology used in this project is unsupervised learning through the K-Means clustering algorithm. The K-Means algorithm groups songs into K distinct clusters based on their similarity in features like danceability, tempo, and volume. The following steps outline the approach taken:

1. Clustering with K-Means:
2. The K-Means algorithm partitions the dataset into K clusters by minimizing the variance within each cluster. Each song is assigned to the cluster whose centroid is closest to the song's feature values. This ensures that songs within the same cluster are similar to each other in terms of their key features.
3. Cosine Similarity:
4. After the clustering step, cosine similarity is used to measure the similarity between songs within the same cluster. Cosine similarity calculates the cosine of the angle between two vectors, with values close to 1 indicating high similarity. This similarity score is then used to recommend songs that are similar to the user's input song.

Algorithm Considerations:

- The K-Means algorithm requires choosing the number of clusters (K) beforehand, which can be challenging. Multiple iterations were conducted to find the optimal K.
- High-dimensionality is a potential challenge, as K-Means can struggle with large numbers of features, where the distances between points become less meaningful.

04. Model

The recommendation system is based on the K-Means clustering algorithm. This model is trained on features such as danceability, tempo, and volume to group songs into clusters. After the clustering step, the model calculates cosine similarity between songs within the same cluster to find the most similar tracks.

Pros:

- Efficiency: K-Means is computationally efficient and well-suited for large datasets like the one used in this project.
- Scalability: The algorithm can handle a large volume of songs, making it a good fit for real-world applications like Spotify.

Cons:

- Choosing K: The algorithm requires specifying the number of clusters in advance, which can be a limitation if the optimal K is unknown.
- High-Dimensionality: In cases where there are many features, the distances between points may lose significance, which affects the clustering performance.

05. Results/Findings

The model successfully clustered the songs into groups based on their danceability, tempo, and volume. By calculating cosine similarity between songs within the same cluster, the system was able to recommend similar songs effectively. The results are:

1. Personalized Recommendations: Users can input a song, and the system will recommend other songs that share similar characteristics, enhancing music discovery.
2. Visualization: Visual representations of clusters were provided, showcasing how songs are grouped based on the selected features.
3. Model Performance: The model was able to deliver accurate and meaningful recommendations, based on similarity measures, demonstrating the success of the K-Means clustering approach.

Demo of Model:

Spotify Recommendation System Demo

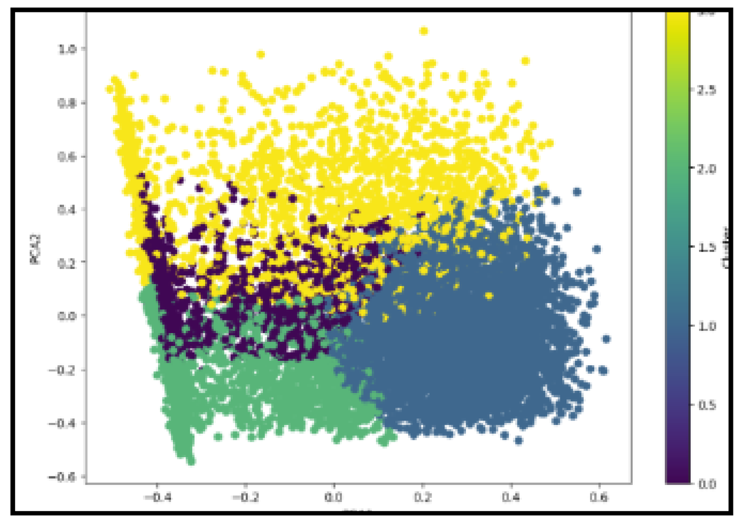
GitHub Repository:

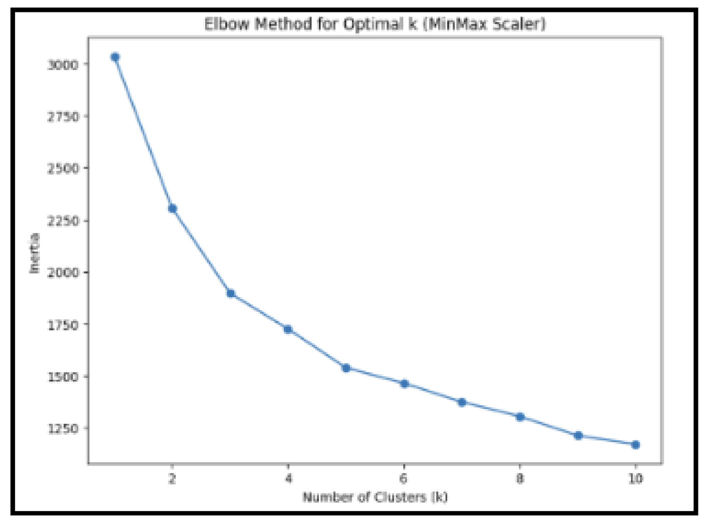
Spotify Recommendation System Repository

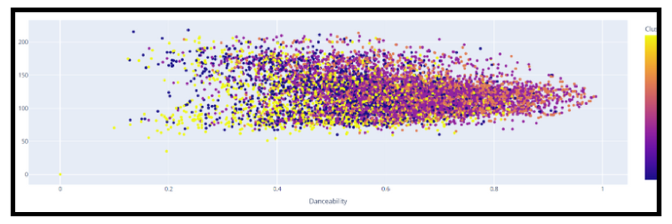
06. Analysis

The K-Means algorithm was successfully implemented to group songs based on their features. We then used cosine similarity to generate personalized recommendations for users. Some key insights from the analysis are:

- Effectiveness of Clustering: The K-Means algorithm was computationally efficient and produced distinct clusters of songs that shared similar characteristics. This structure allowed the recommendation system to suggest songs within the same cluster, ensuring relevance to the user's taste.
- Challenges:
  - Optimal Number of Clusters: Determining the right number of clusters (K) was a crucial decision, as a poorly chosen K could result in ineffective clustering. Iterative testing was done to arrive at the most appropriate value.
  - Bias in Data: The dataset used had a bias towards popular and mainstream songs, which could affect the diversity of recommendations. However, careful feature selection and further exploration of non-mainstream data could help mitigate this bias in future versions.
- User Experience: By testing the system with various input songs, the recommendations provided were relevant and aligned well with the expected similarity based on audio features. This demonstrated the potential of using K-Means clustering for music recommendations.







07. Conclusion

In conclusion, the AI-powered recommendation system built with K-Means clustering was successful in generating personalized music suggestions based on key audio features like danceability, tempo, and volume. The system was able to group songs into meaningful clusters and provide recommendations based on cosine similarity.

However, challenges such as determining the optimal number of clusters and addressing potential biases in the dataset remain. The reliance on mainstream music in the dataset could limit the exposure to emerging or niche artists, a bias that needs to be addressed in future iterations of the model.

Next Steps:

- Further optimize the clustering process and explore other clustering techniques that may improve results.
- Incorporate more diverse datasets to mitigate the bias towards mainstream music and increase the system's fairness.
- Continue developing and refining the system for real-world applications, such as integration with music streaming platforms like Spotify.