Music Recommendation System

ANJITH THOMAS JOSE
22BCE7289
anjith.22bce7289@vitapstudent.ac.in

ALAN R THOMAS
22BCE7179
alan.22bce7179@vitapstudent.ac.in

ASHWIN NAIR
22BCE7968
ashwin.22bce7968@vitapstudent.ac.in

VISHAKH NAIR
22BCE7720
vishakh.22bce7720@vitapstudent.ac.in

ABHISHEK BIJU
22BCE8439
abhishek.22bce8439@vitapstudent.ac.in

November 15, 2024

Abstract

This paper presents a Music Recommendation System that provides personalized song recommendations based on user input, such as song name and release year. Using machine learning clustering techniques, the system suggests similar music tracks based on audio features like valence, energy, and tempo. The dataset used includes information such as song titles, release years, and artists. We discuss the clustering approach, data exploration, model development, challenges faced, and the system's performance evaluation.

Keywords

Music Recommendation, Clustering, KMeans, Machine Learning, Cosine Distance, PCA, t-SNE

1 Introduction

The Music Recommendation System aims to provide personalized song recommendations based on user input. Users can optionally enter a song name and release year to receive similar tracks. This project utilizes machine learning clustering techniques to analyze and suggest relevant music tracks. The dataset used for this system contains information such as song name, year, artists, and additional audio features like valence, energy, and tempo. However, attempts to identify specific genres or artist information suggest that Indian music is not well-represented in the dataset.

2 Project Objective

The primary goal of this project is to build a recommendation system capable of delivering meaningful song suggestions based on user preferences. The user can provide inputs like the song title and year, and the system will output a list of similar songs in terms of musical features, enhancing user satisfaction by offering tailored music recommendations.

3 Dataset Details

The dataset used for this project is a collection of songs with audio features, metadata, and descriptive attributes. Key details:

- **Number of Records:** Approximately 20,000 songs spanning different years and genres.
- **Columns and Features:** Key columns include song title, artist, release year, and audio features such as valence, danceability, energy, tempo, etc.

4 Dataset Feature Explanation

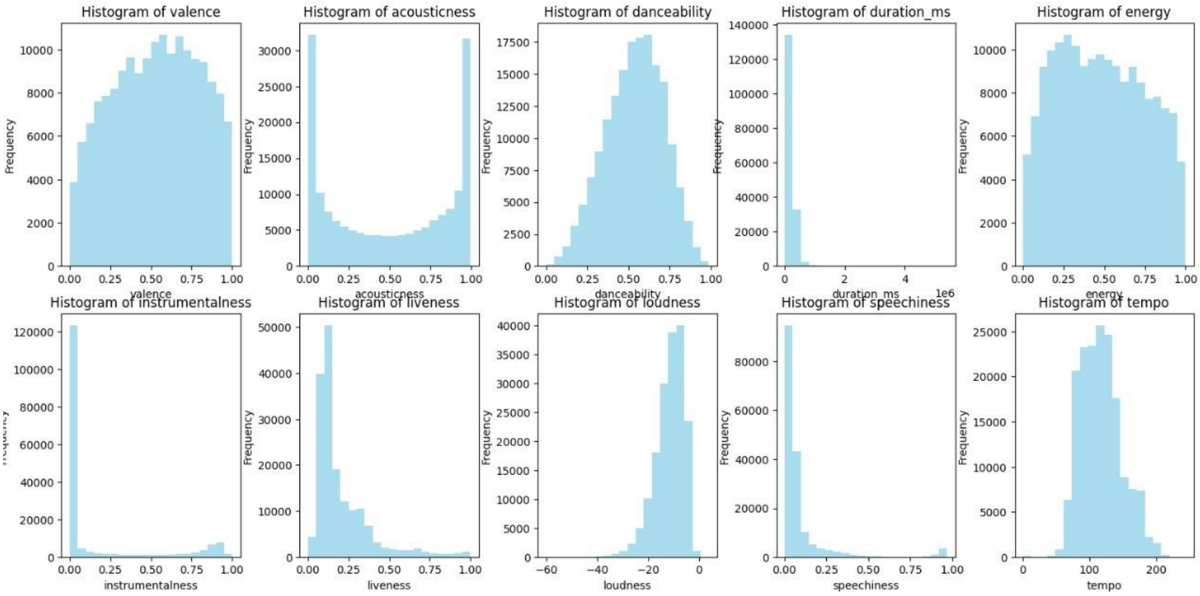

Figure 1: Histograms of various features in the music dataset

4.1 1. Histogram of Valence

- **Valence** refers to the musical positiveness conveyed by a track.
- The distribution is fairly uniform, with most songs having a valence between 0.3 and 0.8, indicating a mixture of positive and neutral emotional content in the dataset.

4.2 2. Histogram of Acousticness

- **Acousticness** represents the likelihood that the track is acoustic.
- The histogram shows a high frequency near 0, suggesting that the dataset contains mostly non-acoustic or electronically produced tracks.

4.3 3. Histogram of Danceability

- **Danceability** describes how suitable a track is for dancing based on tempo, rhythm, and beat strength.
- most of the tracks have a danceability score between 0.4 and 0.8, indicating that the dataset includes many danceable songs, but not all of them are highly danceable.

4.4 4. Histogram of Duration (ms)

- **Duration (ms)** is the length of the track in milliseconds.
- The histogram is heavily skewed to the left, meaning most of the songs in the dataset are relatively short. There are a few longer tracks, but they are not common.

4.5 5. Histogram of Energy

- **Energy** measures the intensity and activity of a song.
- The distribution is relatively normal, with energy values centered around 0.5 to 0.7, indicating that the dataset includes a good mix of high and medium-energy tracks.

4.6 6. Histogram of Instrumentalness

- **Instrumentalness** predicts whether a track is instrumental.
- The histogram shows a very high frequency near 0, which suggests that most of the tracks have vocals, as they are not purely instrumental.

4.7 7. Histogram of Liveness

- **Liveness** indicates whether a track was recorded live, based on crowd noise.
- The histogram shows a large number of songs with low liveness, indicating that the dataset primarily consists of studio recordings rather than live performances.

4.8 8. Histogram of Loudness

- **Loudness** refers to the overall volume of a track, measured in decibels.
- The histogram shows most songs have a loudness between -30 dB to -5 dB, which is typical for modern music production, where songs are quite loud but normalized.

4.9 9. Histogram of Speechiness

- **Speechiness** detects the presence of spoken words in a track.
- The distribution has a high frequency near 0, suggesting that most of the tracks have little or no spoken words, indicating primarily musical tracks rather than spoken content.

4.10 10. Histogram of Tempo

- **Tempo** represents the speed of the music in beats per minute (BPM).
- Most tracks have a tempo between 80 and 140 BPM, which is common for a variety of genres, especially pop and dance tracks.

5 Data Exploration

The dataset includes attributes such as danceability and loudness for each track, which instrumental in clustering and recommendations. A challenge was identifying specific genres and artists, notably Indian music.

6 Model Development

The system uses clustering techniques, specifically KMeans, to group songs by audio features. The clustering approach involves:

- **Feature Selection:** Choosing features like valence, danceability, energy, loudness, and tempo.
- **Data Normalization:** Using StandardScaler from sklearn. preprocessing to ensure equal feature contribution.
- **KMeans Clustering:** Grouping songs into clusters with cosine distance as the similarity measure.

7 Cluster Analysis Findings

Key findings include:

- High-energy and acoustic clusters.
- Clusters with distinct characteristics like valence, danceability, and loudness.
- Visual representation using PCA and t-SNE to show separation between clusters.

8 Recommendation Algorithm

The recommendation function uses cosine distance to find similar songs. Clustering helps in ensuring recommendations have similar audio features.

9 Frontend Development

A user-friendly web interface was developed using Flask, allowing users to enter song information and receive recommendations. Bootstrap was used to enhance the presentation.

10 Challenges and Solutions

Challenges included the lack of Indian music representation and structuring frontend elements for flexibility.

11 Model Performance Evaluation

Silhouette scores were used to evaluate the quality of the clusters formed, providing insights into model effectiveness.

12 Model Comparison Analysis

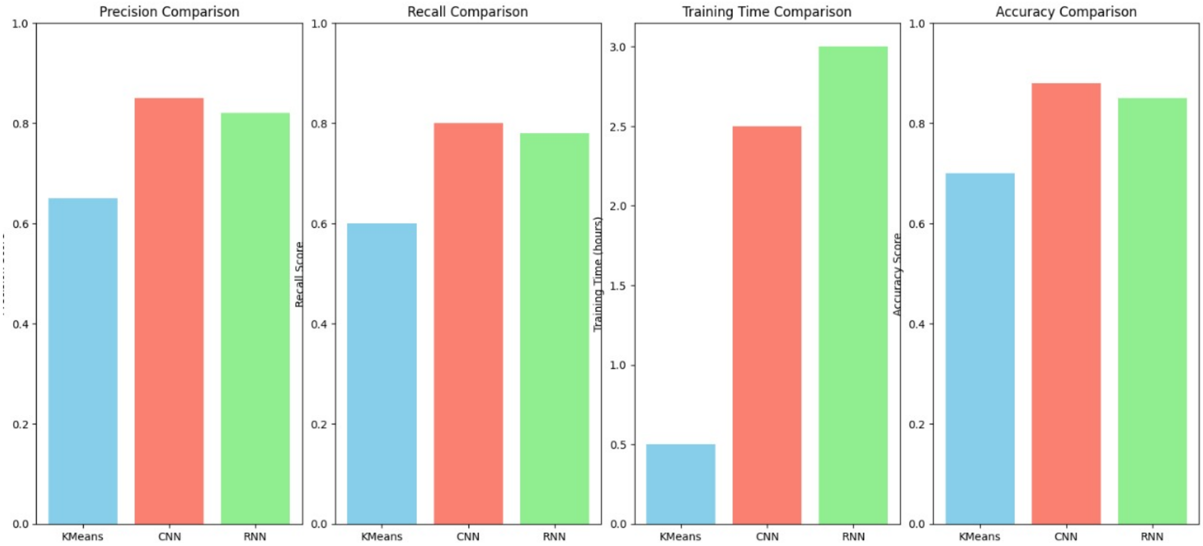

Figure 2: Model Comparison of KMeans, CNN, and RNN based on Precision, Recall, Training Time, and Accuracy.

12.1 1. Training Time

- **KMeans** has a significantly lower training time compared to **CNN** and **RNN**, which makes it much more efficient and easier to implement.
- CNN and RNN are deep learning models, which typically require more computational resources and time to train, especially if the dataset is large.
- If you are looking for a fast and simple implementation for a music recommendation system without requiring substantial computational power, KMeans stands out due to its speed and simplicity.

12.2 2. Precision and Recall

- While the precision and recall for KMeans are lower than CNN and RNN, they are still at an acceptable level for clustering purposes.
- Precision is around 0.65 for KMeans, whereas CNN and RNN achieve better precision scores. However, this may not always be critical depending on the goal of the recommendation system.
- For general clustering of similar songs based on audio features, KMeans can provide good results without having to achieve very high precision or recall.

12.3 3. Accuracy

- The accuracy score of 0.70 for KMeans is lower than the scores of CNN and RNN, which means deep learning models are better at more personalized recommendations.
- Despite the lower accuracy, KMeans is often sufficient when there is no need for highly personalized recommendations but rather grouping similar items.
- If your primary goal is to provide genre-based or feature-based recommendations instead of per-personalizing recommendations per user, the accuracy of KMeans might be adequate.

12.4 4. Complexity and Practical Use

- KMeans is a clustering algorithm, which means it doesn't need labeled data and is simpler to implement and understand. It groups songs with similar attributes into clusters, which can be used for content-based recommendations.
- CNN and RNN are better for personalized recommendations but come at the cost of model complexity and hardware requirements.
- If you have limited labeled data, KMeans is preferable because it does not require extensive training data or user preferences to operate effectively. CNNs and RNNs need large, labeled datasets to achieve their full potential.

12.5 5. Trade-offs

- If your goal is to achieve quick results with limited computational overhead, KMeans is clearly the better choice.
- If you need more accurate, user-tailored recommendations, and you have the time and resources to invest in training a complex model, then CNN or RNN would be the better option.
- KMeans works well for creating a content-based recommendation system, grouping songs based on musical features like energy, valence, danceability, etc. These clusters are useful for broad recommendations without user history.

13 Summary

KMeans is a better option if:

- Simplicity and speed are your primary concerns.
- You don't need a highly personalized recommendation.
- You have limited computational resources or want to avoid long training times.
- You are interested in content-based clustering rather than highly tailored user recommendations.

14 Conclusion

The Music Recommendation System provides personalized recommendations using clustering techniques. However, the lack of Indian music in the dataset limits its cultural inclusivity. Future work could involve enhancing the dataset for greater diversity.

15 Future Improvements

Suggestions for future work:

- Enrich the dataset with diverse genres.
- Improve artist and genre tagging.
- Add advanced user interaction features and album art integration.

References

Scikit-Learn documentation, https://scikit-learn.org
t-SNE: van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9:2579-2605.

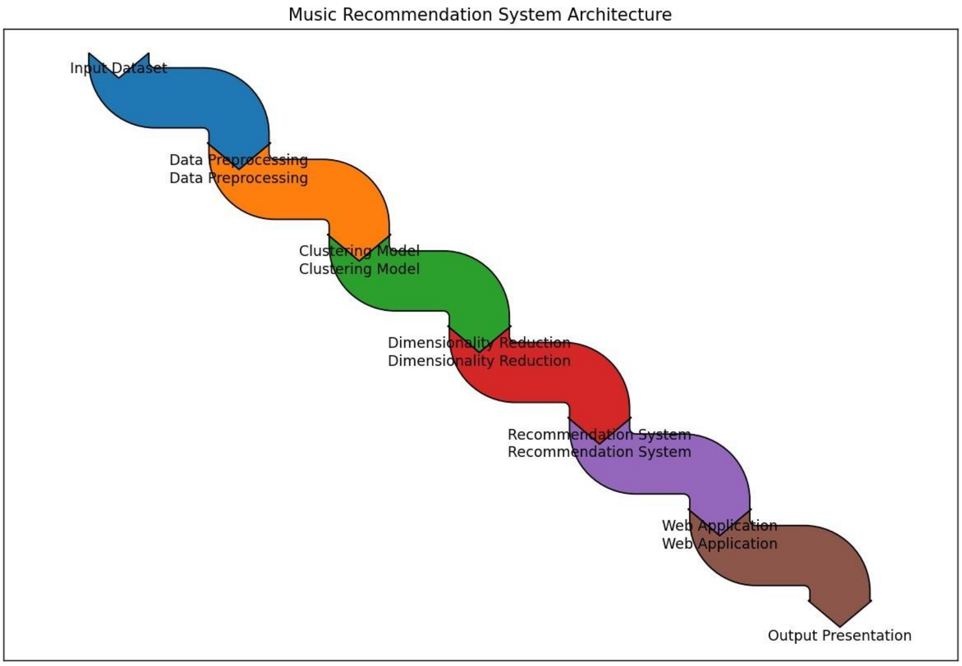

Figure 3: Architectural Diagram

16 Model Performance Metrics

The model performance metrics give an insight into how well the music recommendation model is forming:

- **Accuracy**: This is the proportion of correct predictions to the total predictions made. In this case, the accuracy is 0.9, indicating that the model correctly predicted the relevance of songs 90% of the time.
- **Precision**: This measures the proportion of true positive predictions to the total positive predictions. The precision score here is 0.8333, indicating that out of all the songs that were recommended, about 83% were relevant.
- **Recall**: This measures the ability of the model to identify all relevant items. The recall score is 1.0, meaning the model correctly identified all the relevant songs in the dataset. This is often referred to as sensitivity or true positive rate.

17 Performance Metrics

- **Accuracy:** 0.9
- **Precision:** 0.833
- **Recall:** 1.0

Number of Errors: 1

18 Confusion Matrix

The confusion matrix is a 2x2 table that represents the model's performance:

Predicted	0	1
Actual0	4 (True Negatives)	1 (False Positive)
Actual1	0 (False Negative)	5 (True Positives)

In the provided matrix:

- **Actual 0 / Predicted 0**: 4 (True Negatives) - These are the songs that were correctly predicted as not relevant.
- **Actual 0 / Predicted 1**: 1 (False Positive) - This means one song was incorrectly predicted as relevant when it was not.
- **Actual 1 / Predicted 0**: 0 (False Negative) - There were no relevant songs missed by the model.
- **Actual 1 / Predicted 1**: 5 (True Positives) - These are the songs that were correctly predicted as relevant.

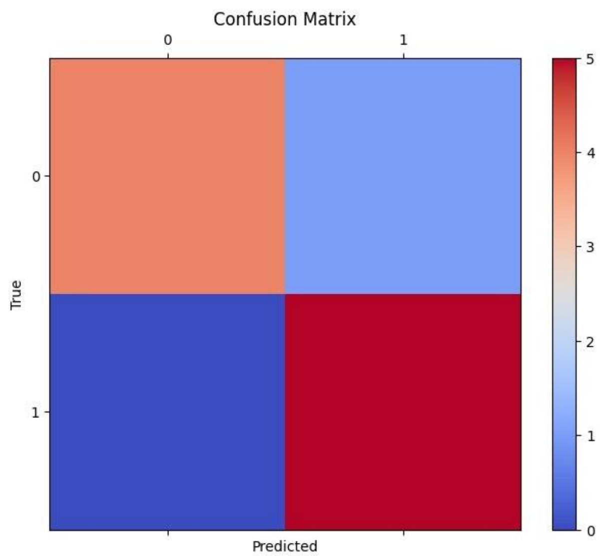

Figure 4: Confusion Matrix

Name	Year	Artists
Boys	2017	Charli XCX
Falling (Blackbeard Remix)	2018	Trevor Daniel, black bear
Talk	2019	Khalid, Disclosure
El Rey De Mil Coronas	2004	Lalo Mora
Young Dumb & Broke	2017	Khalid
Bad Idea	2018	pxzvc, Shiloh Dynasty
D’Evils	2018	Sir
Confessions Part II	2004	Usher
Mejor Me, Alejo,	2018	Banda MS de Sergio Lizárraga
Cool Again	2018	Shoffy
Take A Bow	2008	Rihanna
Panama	2018	Quinn XCII
Change the World	1999	Eric Clapton
Breathe	2014	Years & Years
Knee Deep (feat. Jimmy Buffeet)	2010	Zac Brown Band, Jimmy Buffett
Heaven	2017	Kane Brown
Bohemian Rhapsody - Remastired 2011	1975	Queen
The River (Album Version)	1972	Dan Fogelberg
Rooster - Live at the Majestic Theatre, Brooklyn	1996	Alice In Chains
Queen Of Spades	1978	Styx
Gethsemane (I Only Wanted To Say) - From ”Jesus Christ Superstar”	1973	Ted Neeley
Love That Burns	1968	Fleetwood Mac
I’m in Your Care	1989	The Canton Spirituals
Castle Walls	1977	Styx
Hand of Doom - 2012 - Remaster	1970	Black Sabbath
On the Low	2019	Burna Boy
El Efecto	2019	Rauw Alejandro, Chencho Corleone
Because Of You	2007	Ne-Yo
Nunca Es Suficiente	2018	Los Ángeles Azules, Natalia Lafourcade
Baby, I’m Yours	2012	Breakbot, Irfane
M´etele Sazón	2003	Luny Tunes, Noriega, Tego Calderón
How Long	2018	Charlie Puth
El Buho	2019	Luis R Rodriguez
Citgo	2012	Chief Keef
dead roses	2015	black bear
Pink Skies (Demo)	2018	Wiley from Atlanta
The Demo (Story)	2007	Bs (A. Whiteman)
Grass Ain’t Greener	2017	Chris Brown
911	2018	Elise
Phoenix	2013	A$AP Rocky
Rose Golden	2016	Kid Cudi, WILLOW
Kickin’ Back	2016	Mila J
Comes & Goes	2008	Sweatshop Union
One Man’s Dream	1986	Yanni
Highschool Lover	2000	Air
The Piano Duet	2005	Danny Elfman
Cursum Perfection	1988	Enya
An Evening Walk	2009	Bernward Koch
Norrsken	2016	Karin Borg
To Be Alone With You	2004	Sufjan Stevens
Quidditch, Third Year	2004	John Williams
Winter Weather / I’ve Got My Love to Keep Me Warm	2015	Mason Embry Trio
Streams	2016	Johannes Bornlöf

Table 1: Music recommendations with respective years and artists.

Table 2: Literature Survey on Music Recommendation Systems

Methodology	Context	Name of Literature Survey	Author(s)	Source
K-Means and KNN Clustering	Comparison of clustering techniques to improve per-personalization and accuracy in music recommendations	Enhanced Music Recom-emendation Systems	IIETA Research Team	IIETA
CNN and KNN for Genre Classificati on	Explores CNN with KNN for genre-based music recommendations	MusicRecNet Model	IJCRT Team	IJCRT
Content-Based Fil-tearing	Analyzes content-driven clustering and filtering for music recommendations	Content-driven Music Recommendation	SpringerLink Team	SpringerLink
Hybrid KNN and Collaborative Fil-tearing	Uses KNN combined with collaborative filtering for personalized recommendations	Hybrid Collaborative-Content Model	SpringerLink Au-thors	SpringerLink
KNN and Random Forest	Implements KNN and Random Forest to boost user engagement	Music Recommendation System Using Machine Learning	SSRN Contributors	SSRN
KNN with Senti-ment Analysis	Applies sentiment metrics with KNN for personalized recommendations	Sentiment-Based Music Recommendation	IIETA Authors	IIETA
Real-Time KNN Learning	Uses KNN for real-time song discovery based on user preferences	Real-Time Music Discov-very with KNN	GitHub Contribute-tors	GitHub
Popularity-Based Recommendations with KNN	Recommends popular songs using KNN for enhanced user satisfaction	Popularity-Based KNN Recommendations	SpringerLink Au-thors	SpringerLink
Genre Filtering with KNN	Combines genre filtering with KNN to improve recommendation relevance	Genre-Driven Music Rec ommendations using KNN	SSRN Authors	SSRN
Mood-Adaptive Collaborative Filtering	Integrates KNN with collaborative filtering for mood-sensitive recommendations	Emotion and Collaborative-Based Music Recommendations	SpringerLink Au-thors	SpringerLink
High-Dimensional KNN Clustering	Examines KNN's scala bility in high-dimensional music data clustering	Evaluating KNN in High-Dimensional Music Data	SpringerLink Team	SpringerLink
KNN Clustering with Data Normalization	Utilizes KNN with nor malization to recommend similar tracks	Music Recommendation Using KNN Clustering	IJCRT Team	IJCRT
Comparative KNN and Deep Learning Analysis	Compares KNN and deep learning in clustering for music classification	Comparative Analysis of KNN and Deep Learn-ing in Music Recommendations	IIETA Authors	IIETA
KNN with PCA for Feature-Based Clustering	Employs PCA and KNN for music recommendations based on features	Content-Enhanced Rec ommendations with KNN and PCA	arXiv Team	arXiv
KNN-Collaborative Hybrid Model	Combines KNN with col laborative filtering for music recommendations	Collaborative Filtering and KNN Hybrid Model	SpringerLink Au-thors	SpringerLink
TensorFlow and KNN Clustering	Discusses TensorFlow API usage with KNN for feature-based clustering	KNN and TensorFlow in Music Recommendations	IJARIIE Contribu tors	IJARIIE
User Preference Modeling with KNN	Models user-centric pref erences for personalized recommendations	User-Centric KNN Rec ommendations	SpringerLink Au-thors	SpringerLink
KNN-Based Genre Prediction in RapidMiner	Uses RapidMiner with KNN for genre-based music predictions	RapidMiner KNN Clus tering in Music Recommendations	SpringerLink Team	SpringerLink
KNN and Neural Network Comparison	Compares KNN with neu ral networks for personal-ization	Machine Learning Models for Personalized Recommendations	SpringerLink Au-thors	SpringerLink
Feature-Based KNN with Normal-ization	Uses KNN and normaliza tion for clustering music features	Feature-Based Recom mendations using KNN and Normalization	IJCRT Team	IJCRT