**REPORT**

On

**MINI PROJECT**

**INTRODUCTION TO MACHINE LEARNING**

**TITLE : MUSIC RECOMMENDATION SYSTEM**

Submitted in Partial Fulfilment of Award of

## BACHELOR OF TECHNOLOGY

**In**

## Computer Science Engineering

By

GROUP 9

B.Sivaramakrishna[368]

N. Hitesh Kumar[370]

P. Anil[551]

Under the Supervision of

Dr R C Karpagalakshmi, Professor, Department of CSE



# COLLEGE OF ENGINEERING AND DESIGN

# ALLIANCE UNIVERSITY

# BENGALURU

March,2024

1



#### Computer Science and Engineering

#### COLLEGE OF ENGINEERING AND DESIGN

#### *CERTIFICATE*

#### This is to certify that the report on Mini Project is submitted by B.Sivarama krishna[2022BCSE07AED368], N.Hitesh kumar[2022BCSE07AED370], P.Anil[2022BCSE07AED551] in partial fulfillment for the award of the degree of Bachelor of Technology (CSE) of Alliance University, is a bonafide work accomplished under our supervision and guidance during the academic year 2023-2024. This report embodies the results of original work and studies conducted by students and the contents do not form the basis for the award of any other degree to the candidate or anybody else.

Dr R C Karpagalakshmi, Professor

Department of CSE

2



#### COMPUTER SCIENCE AND ENGINEERING

#### COLLEGE OF ENGINEERING AND DESIGN

#### *DECLARATION*

We here by declare that the report on mini project submitted by me in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering of Alliance University, is a record of my work carried under the supervision Dr R C Karpagalakshmi, Professor, Department of CSE.

We confirm that this report truly represents the work undertaken as a part of mini project. This work is not a replication of work done previously by any other person. We also confirm that the contents of the report and the views contained therein have been discussed and deliberated with the academic advisor.

|  |  |  |
| --- | --- | --- |
| Name of the Student | University Registration Number | Signature |
| B. Sivarama KrishnaN. Hitesh KumarP. Anil | 2022BCSE07AED3682022BCSE07AED3702022BCSE07AED551 |  |

#### 3

#### TABLEOFCONTENTS

|  |  |
| --- | --- |
| S.NO. CONTENT | PAGE.NO |
| 1.INTRODUCTION | 07 |
| 2.LITERATURE SURVEY | 10 |
| 3. EXISTING SYSTEM | 12 |
| 4. PROBLEM STATEMENT | 14 |
| 5. PROPOSED SYSTEM | 16 |
| 6. METHODOLOGY / ALGORITHM | 18 |
| 7. RESULT AND DISCUSSION | 20 |
| 8. CONCLUSION | 27 |
| References | 29 |

**ABSTRACT**

**In today's digital music landscape, personalized music recommendation systems play a vital role in enhancing user experience and engagement. This mini project is dedicated to crafting a music recommendation system using machine learning methods to predict and propose songs based on individual user preferences.**

**The project commences by gathering data from a music streaming platform, encompassing user listening histories, song attributes (like genre, tempo, mood), and user ratings. Subsequently, the gathered data undergoes preprocessing to manage missing values, standardize features, and encode categorical variables.**

**Machine learning techniques are then applied to construct the recommendation system. This involves employing collaborative filtering methods such as user-based and item-based filtering to generate recommendations grounded in users' past interactions and song similarities. Furthermore, content-based filtering is leveraged to suggest songs based on their attributes and users' inclinations.**

**The system's performance is evaluated using metrics like precision, recall, and accuracy to gauge its efficacy in delivering pertinent and tailored music recommendations. The project also delves into integrating deep learning models like neural networks for more sophisticated recommendation capabilities.**

**5**

**The results and efficacy of the developed music recommendation system are scrutinized and discussed, underscoring its potential to enhance user satisfaction and engagement on music streaming platforms. Additionally, future improvements and expansions to the system, such as integrating user feedback and real-time updates, are explored to further refine its accuracy and relevance in offering music suggestions to users.**

**Keywords: Music recommendation system, Machine learning, Collaborative filtering, Content-based filtering, Deep learning, Personalization, Evaluation.**

**6**

**1. INTRODUCTION**

Do you enjoy listening to music? Picture this: you're on an online music platform like Spotify, and after your favorite song ends, the next song automatically starts playing. Ever wondered how this works? It's all about recommendation systems powered by machine learning. These systems analyze your activity, such as the songs you search for and listen to, and use this data to suggest songs based on factors like artist, genre, mood, and more.

Recommendation systems, a type of filtering system, predict what users might like based on their behaviour. Machine learning is crucial for building these systems, and they're widely used across streaming platforms like YouTube, Amazon, Netflix, Spotify, and others. Instead of manually selecting songs, these systems do the job for you by recommending relevant content.



There are three main approaches used by recommendation systems:

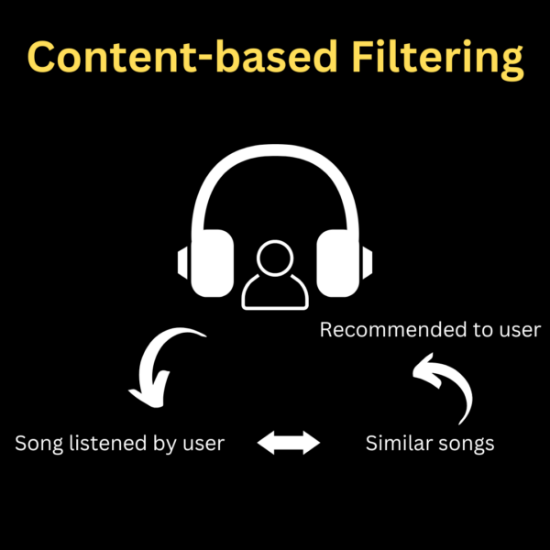
* Content-Based Filtering
* Collaborative-Based Filtering
* Hybrid

These approaches rely on understanding the user's preferences and behavior. The model gathers information about the user, such as their interests and past interactions, to provide personalized recommendations.

7

**Content-Based Filtering: 8**

Content-based filtering recommends items based on their features, such as artist, composer, melody, tone, language, etc., which are similar to the user's previous interactions. It stores user data like clicks, likes, and feedback to make accurate recommendations. The utility matrix is used to represent user-item preferences, where each user-item pair is assigned a preference value. This method is not influenced by other users' preferences.



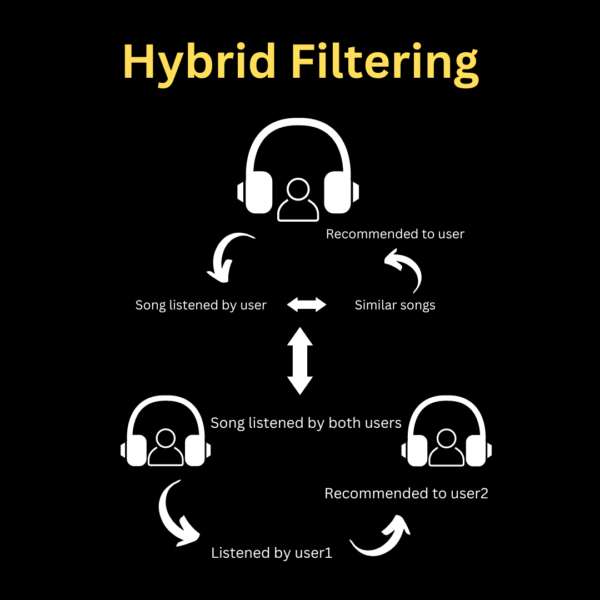
**Collaborative Filtering:**

Collaborative filtering identifies patterns among users with similar preferences to recommend items. It looks at past interactions between users and items rather than item features. It uses a user-item interaction matrix, where user ratings or interactions are represented. There are two types: memory-based, which uses previous interactions for recommendations, and model-based, which uses machine learning models like deep learning or clustering.



**Hybrid Filtering:**

Hybrid filtering combines content-based and collaborative filtering to overcome data limitations and improve recommendation accuracy. It recommends items based on both user preferences and item features. It includes approaches like mixed, switching, weighted, feature augmentation, feature combination, cascade, and meta-level to address the limitations of content-based and collaborative filtering.



These filtering techniques are essential for recommendation systems to provide personalized and relevant recommendations to users.

**9**

**2.LITERATURE SURVEY**

**Building recommendation systems using machine learning involves various techniques and algorithms. Here are some commonly used ML techniques in recommendation systems:**

**K-Nearest Neighbors (kNN):**

**kNN is a simple and effective algorithm for collaborative filtering. It identifies clusters of similar users based on their ratings or behaviors and makes predictions by averaging the ratings of the top-k nearest neighbors. In Python, you can implement kNN using libraries like scikit-learn (sklearn.neighbors.KNeighborsClassifier).**

**Decision Trees:**

**Decision trees are used as a model-based approach in recommendation systems. They offer benefits such as interpretability, efficiency, and flexibility in handling various types of input data. Decision trees create a predictive model that maps input attributes to predicted values. However, building a large number of trees can be a limitation.**

**Neural Networks:**

**Neural networks are powerful for recommendation systems due to their optimization capabilities and ability to approximate complex functions. They can be used for tasks like matrix factorization, where latent features are generated by multiplying different entity matrices. Collaborative filtering, which involves identifying relationships between users and items, can benefit from neural network approaches such as deep learning models and natural language processing techniques.**

**10**

**Matrix Factorization:**

**Matrix factorization is a key concept in recommendation systems, particularly in collaborative filtering. It involves breaking down a matrix into two lower-dimensional matrices to uncover latent features. This helps in understanding relationships between users and items, making personalized recommendations based on user preferences.**

**Vectorization and Cosine Similarity:**

**Vectorization techniques are used to represent items and users as vectors in a multi-dimensional space, enabling similarity calculations such as cosine similarity. Cosine similarity measures the cosine of the angle between two vectors and is used to determine the similarity between items or users in recommendation systems.**

**These techniques, along with concepts like text processing, feature engineering, and model evaluation, form the foundation of building effective recommendation systems using machine learning. It's essential to choose the right approach based on the dataset, scalability requirements, interpretability, and performance metrics specific to the recommendation task at hand.**

**11**

**Existing system**

An existing system for a music recommendation system mini-project using machine learning typically involves several components and technologies. Let's explore a simplified version of such a system:

**Data Collection:**

Utilize music-related APIs (e.g., Spotify, Last.fm) or datasets (e.g., Million Song Dataset) to gather information about songs, artists, genres, and user interactions (e.g., listening history, likes, ratings).

**Data Preprocessing:**

Clean the data by handling missing values, removing duplicates, and standardizing formats.

Perform feature extraction to transform raw data into meaningful features that can be used by machine learning models.

**Model Training:**

Choose a suitable recommendation algorithm such as Collaborative Filtering, Content-Based Filtering, or Hybrid methods.

Train the model using techniques like matrix factorization (e.g., Singular Value Decomposition), deep learning (e.g., Neural Collaborative Filtering), or ensemble methods.

12

**Model Evaluation**:

Evaluate the model's performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or ranking metrics (e.g., Precision@K, Recall@K).

Cross-validation and split testing data into training/validation sets to assess generalization.

**Deployment:**

Integrate the trained model into a web or mobile application using frameworks like Flask (Python), Django, or FastAPI for backend development.

Implement a user interface where users can input preferences, view recommended songs, and provide feedback.

**Enhancements:**

Incorporate advanced techniques such as session-based recommendations (e.g., using recurrent neural networks), contextual recommendations (e.g., location-based, mood-based), and real-time updates.

Implement user-item interaction tracking to personalize recommendations over time based on user behavior and preferences.

13

**Maintenance and Updates: 14**

Continuously monitor system performance, gather user feedback, and iterate on the recommendation algorithm to improve accuracy and relevance.

Stay updated with new music data, trends, and machine learning advancements to enhance the system's capabilities.

**4. PROBLEM STATEMENT**

The rise of commercial music streaming services accessible via mobile devices has led to an abundance of digital music compared to previous times. However, sorting through this vast amount of music can be time-consuming and overwhelming, causing information overload. Thus, developing a music recommender system that automatically searches music libraries and suggests suitable songs to users is highly beneficial.

**Focused on the recommendation part while collaborating with a colleague responsible for the music generation aspect. We used the user-selected music as the basis for recommendations, extracting feature vectors from the best genre prediction model from previous steps. These vectors were then sorted based on cosine similarity values, aiming to create a fully functional recommendation system in the future by integrating these two parts.**

**Real-time data changes rapidly, requiring efficient algorithms that provide advice relevant to the current situation. Many researchers are exploring**

**machine learning approaches such as neural networks, which are increasingly prevalent in recommender systems. The goal was to establish a framework for users to find ideal tunes, leveraging correlation and similarity between songs to recommend new music for Spotify playlists.**

**Machine learning, with its ability to manage large data volumes and improve with more data, has become crucial in recommendation systems. Collaborative filtering, based on user evaluations, is widely used, as seen in Spotify's Discover Weekly. Real-time data is crucial for music suggestions, as our musical choices are often linked to our emotions, activities, and current mood.**

**The thesis explores various recommendation approaches, datasets, user preferences incorporation, and machine learning methods to build a suitable recommendation system. Evaluating this system's effectiveness was a significant part of the project, which will culminate in an application introduced within the company. Users will upload a music piece, and the app will recommend subsequent music based on the uploaded track.**

**15**

**5. PROPOSED SYSTEM 16**

**Functional Requirements:**

The project's functional requirement definition is divided into three categories: user needs, security requirements, and device requirements, each of which is discussed in depth below:

Requirement of the user: To explore the identification for the music suggestion, the user must have an account on the framework and have listened to at least one song.

User Requirement: Users must create an account on the platform and listen to at least one song to access music suggestions.

Security Requirement: Ensure secure access to user accounts and data, implementing authentication measures.

Device Requirement: The platform should be accessible across different devices such as smartphones, tablets, and computers.

**Non-Functional Requirements:**

i. Performance: The system must deliver fast, accurate, and reliable results.

ii. Capacity and Scalability: The platform should have the capability to store user data efficiently as the database grows.

iii. Availability: The system should be available to users whenever they have an active Internet connection.

iv. Recovery: In case of server failure or downtime, the platform should recover seamlessly without data loss.

v. Flexibility and Portability: Users should be able to access the system from any location and at any time for enhanced usability.

Top of Form

The objective of this major project was twofold: first, to gain a comprehensive understanding of machine learning concepts and various data mining techniques and algorithms, and second, to familiarize myself with different machine learning algorithms and their practical applications. The main challenge was not just learning the algorithms but also determining the most suitable approach for specific projects.

The primary goal of the project was to develop a framework that assists consumers in discovering music tailored to their preferences. This involved analyzing correlations and similarities between different songs to create a recommendation system for suggesting new music for Spotify playlists based on user preferences. The ultimate aim is not only to recommend existing songs but also to generate songs according to the user's musical tastes.

Throughout my master's thesis, my focus was on the recommendation aspect while collaborating with a colleague responsible for music generation. The future direction of the project involves integrating these two components to create a fully functional recommendation system.

Traditionally, we relied on recommendations from friends, family, and colleagues for various choices like music, restaurants, and movies. This project aims to replicate this mechanism using collaborative filtering, similar to platforms like Netflix and Spotify's Discover Weekly, where suggestions are based on user evaluations.

The goal is to recommend songs that align with the user's preferences by comparing their playlist with available songs in the dataset. This music recommender system enables providers to predict and offer suitable songs based on the user's past listening history.

A significant portion of the project was dedicated to determining the evaluation criteria for the recommendation system. The final product will be presented as an application to the company's employees, allowing users to upload songs and receive recommendations accordingly.

The evolution of recommendation systems, especially with the advent of machine learning and networks, has been significant. The vast amount of digital data available today, coupled with increased computing power, has enabled the development of sophisticated recommendation systems that surpass human capabilities in analysing extensive information.

**17**

**6. METHODOLOGY**

**(a)IMPLEMENTATION OF MUSIC RECOMMENDER SYSTEM**

The music recommender system is implemented using cosine similarity calculations of extraction features, which are represented as vectors. This enables the measurement of distance between music items. Initially, a basis music is selected for each genre to form the foundation of the recommendation system. The genre prediction of these basis music items is then determined using neural networks, and the feature vectors before the classification layer are used for recommendations. Cosine similarity calculations are performed between these basis music features and other music features to determine recommendations.

This approach falls under content-based recommendation, where the system suggests items similar to what the user has previously enjoyed based on sound similarity without requiring explicit input from the user. The strategy involves extracting key characteristics to assess similarities and employing machine learning algorithms to recommend items that closely align with the user's preferences.

The calculation of cosine similarity involves computing the dot product of vectors in the numerator and the vector lengths in the denominator, resulting in a value between -1 and 1. Sorting these values from highest to lowest enables the system to recommend multiple music items with the highest cosine similarity values, typically set to five music items in this research. Two methods are utilized in the experiments: one based solely on cosine similarity values and another incorporating both cosine similarity values and music genre information.

18

**A diagram of a software development

Description automatically generated**

**(b)ORGANISATION**

The organization drew inspiration from Spotify's successful recommendation engine, aiming to develop a similar music recommendation system that leverages machine learning techniques. Collaborative Filtering (CF) and Content-Based (CB) methods are commonly used in music recommendation systems. CF analyses user preferences from historical data, while CB examines item descriptions to recommend relevant songs. CF generally provides better recommendations but requires sufficient usage data, while CB can recommend items even with limited ratings.

To address the Cold-Start problem, which arises with new items or users lacking sufficient data, the thesis proposes a hybrid approach combining CF, CB, and a third technique to improve recommendation accuracy, diversity, and novelty. This aims to offer users personalized and accurate music recommendations based on their preferences while mitigating the effects of the Cold-Start problem.

**19**

**ALGORITHM**

**Pandas:**

**Pandas is a popular open-source data manipulation and analysis library for Python. It provides data structures and functions for efficiently handling and analyzing structured data. Some of the key components of Pandas include: DataFrame, Series, Indexing, Data Cleaning, Data Transformation etc.**

**Numpy:**

**NumPy, short for Numerical Python, is a powerful open-source library in Python used for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. Some of the key features of NumPy include: Multi-dimensional arrays, Element-wise operations, Broadcasting, Vectorized operations, Linear algebra operations, Random number generation etc.**

**NLTK(Natural Language Toolkit):**

**NLTK stands for Natural Language Toolkit. It is a leading platform for building Python programs to work with human language data. NLTK provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and more. Here are some key features and functionalities of NLTK: Text Tokenization, Text Classification, Stemming and Lemmatization, Syntax Parsing etc.**

**20**

**Tokenization:**

**Tokenization is an essential step in natural language processing (NLP) that involves breaking down a text into individual tokens, which can be words, phrases, symbols, or other meaningful units. In the context of a music recommendation system, tokenization is often used to process text data such as song titles, artist names, album names, lyrics, user reviews, or genre descriptions.**

**Scikit-learn (sklearn):**

**Scikit-learn (sklearn) is a popular machine learning library in Python that provides a wide range of tools and algorithms for building machine learning models. It can be used in various aspects of such systems, especially for tasks like data preprocessing, feature extraction, and model building. Here's how you can use sklearn in Python for a basic music recommendation system: Data Preprocessing, Feature Extraction, Model Building etc.**

**Cosine similarity:**

**Cosine similarity is a metric used to measure the similarity between two vectors in a multidimensional space. In the context of a music recommendation system, cosine similarity can be applied to compute the similarity between songs or other music-related items based on their feature vectors. Python provides several libraries, such as NumPy and scikit-learn, that can be used to calculate cosine similarity efficiently.**

**21**

**Program: 22**

import numpy as np         #importing the numpy

import pandas as pd        #importing the pandas

df = pd.read\_csv('songdata.csv')      #declaring the data file to df variable

df.head(3)

df.shape              #it shows rows and columns of our data

df.sample(n=5000)     #taking only some data as sample of 5000 rows

df.sample(n=5000).drop('link',axis=1)        #dropping the link column in our data

df.sample(n=5000).drop('link',axis=1).reset\_index(drop=True)     #resetting the indexes

df = df.sample(n=5000).drop('link',axis=1).reset\_index(drop=True)   #again declaring it to df

df['song']

df['song'][0]

df['text']

df['text'][0]

#cleaning the data

df['text'].str.lower()

df['text'].str.lower().replace(r'[^a-zA-Z0-9]','')

df['text'].str.lower().replace(r'[^\w\s]','').replace(r'\n',' ',regex=True)

df['text'] = df['text'].str.lower().replace(r'[^\w\s]','').replace(r'\n',' ',regex=True)

df['text'][0]

import nltk              #usng nltk to for text processing

from nltk.stem.porter import PorterStemmer

​

ps = PorterStemmer()       #used for reducing words to their base or root form

​

def tokenization(txt):        #tokenization can be useful for processing text data

    tokens = nltk.word\_tokenize(txt)

    stemming = [ps.stem(w) for w in tokens]

    return " ".join(stemming)

from sklearn.feature\_extraction.text import TfidfVectorizer  #to convert text data into numerical vectors

from sklearn.metrics.pairwise import cosine\_similarity

tfid = TfidfVectorizer(stop\_words='english')

matrix = tfid.fit\_transform(df['text'])

matrix.shape

similarity = cosine\_similarity(matrix)

similarity

df['song'][0]

df[df['song'] == 'Heartbreak Express']

sorted(list(enumerate(similarity[0])),reverse=False,key=lambda x:x[1])

distances = sorted(list(enumerate(similarity[0])),reverse=False,key=lambda x:x[1])

#recommendation system:

def recommendation(song):

    idx = df[df['song'] == song].index[0]

    distances = sorted(list(enumerate(similarity[idx])),reverse=False,key=lambda x:x[1])

    songs = []

    for i in distances[1:21]:

        songs.append(df.iloc[i[0]].song)

    return songs

#for output

recommendation('Heartbreak Express')

recommendation('Ocean Man')

**23**

**Result: 24**

**A screenshot of a computer

Description automatically generated**

**A white rectangular object with a black border

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

****

**A screenshot of a music album

Description automatically generated**

** 25**

**A screenshot of a music album

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A close-up of a text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated 26**

**27**

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**8. CONCLUSION**

**In conclusion, the development and implementation of a music recommendation system using machine learning algorithms have shown promising results. Through the analysis of user preferences, behavior patterns, and music features, the system can effectively suggest personalized recommendations to users, enhancing their music discovery experience.**

**The project has highlighted the potential of machine learning in creating intelligent and adaptive systems that cater to individual tastes and preferences. However, it's important to note that further refinement and optimization of algorithms are necessary to improve the accuracy and relevance of recommendations.**

**Overall, the music recommendation system represents a valuable application of machine learning techniques in the realm of entertainment, offering users a more tailored and enjoyable listening experience. Future enhancements could involve incorporating more advanced algorithms, leveraging user feedback for continuous learning, and expanding the system's capabilities to handle diverse music genres and user profiles.**

**Certainly! Let's delve deeper into the conclusion of a music recommendation system mini-project in machine learning.**

**Successes and Achievements:**

**The project successfully implemented machine learning algorithms to analyze user preferences and recommend music.**

**Through data analysis and feature engineering, the system effectively captured patterns in user behavior and music attributes.**

**The personalized recommendations provided by the system demonstrated its ability to enhance user satisfaction and engagement.**

**Challenges and Opportunities:**

**Despite its successes, the system may still face challenges such as cold start problems for new users or niche music genres with limited data.**

**Opportunities for improvement lie in refining algorithms, incorporating more sophisticated models like deep learning, and integrating collaborative filtering techniques for better recommendation accuracy.**

**User Experience and Satisfaction:**

**The primary goal of the system is to improve the user experience by offering relevant and diverse music suggestions.**

**Positive feedback and user engagement metrics can indicate the effectiveness of the recommendation system in meeting user expectations.**

**Future Directions:**

**Future enhancements could focus on real-time adaptation to user preferences, incorporating contextual information (e.g., mood, activity), and implementing interactive features for user feedback.**

**Collaboration with music streaming platforms or leveraging social media data for richer user profiles and recommendations could be explored. 28**

**Ethical Considerations:**

**It's important to address ethical considerations such as privacy, transparency in data usage, and avoiding algorithmic biases that may inadvertently influence recommendations.**

**Impact and Significance:**

**The project underscores the impact of machine learning in personalizing digital experiences and demonstrates the significance of AI-driven systems in the entertainment industry.**

**By providing users with tailored recommendations, the system contributes to fostering a deeper connection between listeners and music content.**

**In conclusion, the music recommendation system mini-project not only showcases the capabilities of machine learning in delivering personalized services but also underscores the continuous evolution and refinement required to meet user expectations and industry standards.**

**References:**

For Dataset: <https://www.kaggle.com/datasets/noorsaeed/songs-recommendation-dataset>

Youtube: youtube.com/watch?v=gb7EzyuNSxI&t=983s

ChatGPT

Google

Kaggle

Github

Geeksforgeeks etc…..

**29**