A

PROJECT REPORT

ON

"MUSIC GENRE CLASSIFICATION AND RECOMMENDATION SYSTEM"

SUBMITTED TO

SHIVAJIUNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR. Aditya Kokate	19UCS060
MISS. Neha Khade	19UCS058
MISS. Rashi Khabani	19UCS057
MISS. Pratiksha Kumbhar	19UCS074
MISS. Riya Bisht	19UCS016

UNDER THE GUIDANCE OF

MRS.D. M. Kulkarni

DKTE

Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY'S TEXTILE AND ENGINEERING INSTITUTE,

ICHALKARANJI

2022-2023

D.K.T.E.SOCIETY'S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOUMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

"MUSIC GENRE CLASSIFICATION AND RECOMMENDATION SYSTEM"

is a bonafide record of project work carried out in this college by

MR. Aditya Kokate	**19UCS060**
MISS Neha Khade	**19UCS058**
MISS. Rashi Khabani	**19UCS057**
MISS. Pratiksha Kumbhar	**19UCS074**
MISS. Riya Bisht	**19UCS016**

is in the partial fulfillment of award of degree bachelor's in engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-2023.

D. M. Kulkarni
(PROJECT GUIDE)

PROF.(DR.) D.V. KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) L.S. ADAMUTHE
(DIRECTOR)

EXAMINER: ________________

DECLARATION

We hereby declare that the project work report entitled "**MUSIC GENRE CLASSIFICATION AND RECOMMENDATION SYSTEM**" which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.E.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

MR. Aditya Kokate	19UCS060
MISS. Neha Khade	19UCS058
MISS. Rashi Khabani	19UCS057
MISS. Pratiksha Kumbhar	19UCS074
MISS. Riya Bisht	19UCS016

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to **MRS.D. M. Kulkarni** for her valuable guidance, support, and encouragement in the completion of this project report.

Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MR. Aditya Kokate	19UCS060
MISS. Neha Khade	19UCS058
MISS. Rashi Khabani	19UCS057
MISS. Pratiksha Kumbhar	19UCS074
MISS. Riya Bisht	19UCS016

ABSTRACT

With the increasing availability and diversity of music in the digital age, there is a growing need for efficient systems that can automatically classify music into genres and recommend personalized music to users. This project aims to address this need by developing a music genre classification and recommendation system using machine learning techniques. The system utilizes a dataset of audio samples and extracts relevant features to train a classification model capable of accurately classifying music into different genres. Additionally, a recommendation system is built to suggest music genres based on user preferences and listening history. The project's objectives include implementing the classification and recommendation algorithms, evaluating their performance, and enhancing the user experience through personalized recommendations. The scope of the project focuses on popular music genres and aims to improve the music listening experience for users. However, it acknowledges the subjective nature of genre classification and the limitations that arise from such subjectivity. Through a well-defined timeline and project management plan, this project aims to contribute to the field of music genre classification and recommendation, offering a pathway to more personalized and engaging music experiences.

INDEX

1. INTRODUCTION

Music genre classification and recommendation systems have gained significant attention in recent years due to the vast amount of music available online. The ability to automatically classify music into different genres and recommend suitable music to users has become crucial for personalized music experiences and content discovery. This project aims to develop a music genre classification and recommendation system that utilizes machine learning techniques to accurately classify music and provide users with personalized music recommendations.

The advent of digital music platforms and the exponential growth of music availability on the internet have revolutionized the way people consume and interact with music. However, the sheer volume and diversity of music have posed new challenges for users in effectively exploring and discovering content that aligns with their preferences. Music genre classification and recommendation systems have emerged as crucial solutions to address these challenges and enhance the music listening experience for users.

The primary objective of this project is to develop a music genre classification and recommendation system that leverages machine learning techniques to automate the process of categorizing music into genres and providing personalized recommendations. By accurately classifying music and suggesting relevant genres, the system aims to facilitate efficient music discovery, enabling users to explore a vast array of musical content and discover new artists and songs that align with their tastes.

Music genre classification is a fundamental task in music information retrieval that involves assigning music tracks to predefined categories or genres based on their inherent musical characteristics. Traditionally, manual classification by human experts has been employed, but this approach is time-consuming, subjective, and unable to scale with the ever-growing music collections available today. By harnessing the power of machine learning algorithms, the project seeks to automate this process and provide accurate genre labels to music tracks.

Additionally, the project aims to develop a recommendation system that suggests music genres based on user preferences. Personalized music recommendations have become indispensable for providing users with a tailored music listening experience. By analyzing user interactions, listening history, and demographic information, the recommendation system will generate suggestions that align with the user's preferences, fostering music exploration and engagement.

The significance of this project lies in its potential to revolutionize the way users discover and engage with music. By automating the genre classification process and providing personalized recommendations, users can overcome information overload and navigate through extensive music libraries with ease. Moreover, the system has implications for various stakeholders, including music streaming platforms, artists, and listeners, as it can facilitate content curation, increase user satisfaction, and drive music discovery.

To achieve the project's objectives, a suitable dataset comprising diverse music tracks from different genres will be collected and preprocessed. Feature extraction techniques will be employed to represent the audio data in a meaningful way, capturing essential musical characteristics. Machine learning algorithms, such as support vector machines, random forests, or deep neural networks, will be implemented and evaluated for their performance in accurately classifying music genres.

The recommendation system component will leverage collaborative filtering algorithms to generate personalized music genre suggestions for users. By analysing patterns in user preferences and similarities among different users, the system will provide tailored recommendations, thus enhancing the user experience and promoting music exploration.

In conclusion, this project aims to develop a music genre classification and recommendation system that harnesses machine learning techniques to automate genre classification and provide personalized music recommendations. By accurately classifying music and suggesting relevant genres based on user preferences, the system aims to enhance the music listening experience, foster music discovery, and revolutionize the way users engage with and explore music in an era of abundant digital content.

1.1 Problem definition:

The problem addressed in this project is the lack of efficient and accurate methods for music genre classification and recommendation. With the increasing availability and diversity of music, manual classification and personalized music recommendations have become impractical. This project seeks to automate the process by developing a system that can accurately classify music into genres and recommend relevant music based on user preferences.

1.2 Aim and objective of the project:

Aim:

Develop a music genre classification and recommendation system that enhances the music listening experience for users.

Objectives:

The specific objectives of the project include:

- Implementing a machine learning algorithm for music genre classification.
- Building a recommendation system that suggests music based on user's genres preferences.
- Evaluating the accuracy and performance of the classification and recommendation models.
- Improving the user experience by providing user-friendly GUI.

1.3 Scope and limitation of the project:

1.3.1 Scope of the Project:

- Users: The software is usable by individual users who are interested in music genre classification and recommendation. It can be used by music enthusiasts, music streaming platforms, or anyone looking for automated genre classification and personalized music recommendations.

- Simultaneous Users: The software can handle multiple users concurrently as it is built using Flask and Streamlet frameworks. However, the actual number of simultaneous users it can support depends on various factors such as server capacity, computational resources, and network bandwidth. Proper scaling and optimization can increase the number of concurrent users.

- Music Genre Classification: The software provides an automated music genre classification system that can analyze and classify audio files into specific genres. It can help users organize their music libraries, provide metadata for music streaming platforms, or assist in genre-based analysis and research.

- Music Genre Recommendation: The software offers a music genre recommendation system based on user preferences and selected genres. It allows users to customize their desired music features and receive recommendations that match their preferences. This can enhance the music discovery experience and provide personalized playlists.

1.3.2 Limitations of the Project:

- Scalability: The project's scalability depends on the underlying infrastructure and resources allocated to the software. If the system resources are limited, it may not be able to handle a large number of users or process a high volume of music files simultaneously. Scaling measures such as load balancing and efficient resource management can mitigate this limitation.

- Real-Time Updates: The project does not consider real-time updates of the classification and recommendation models. If new music genres emerge or user preferences change, the system would require manual updates and retraining of the models. This limitation restricts the system's ability to adapt quickly to evolving music trends and user preferences.

- Performance: The project's performance, including classification speed and recommendation generation, depends on the computational resources available and the complexity of the models used. If the models or feature extraction processes are computationally intensive, it may affect the system's response time and overall performance.

- Genre Subjectivity and Diversity: Music genres can be subjective, and there is often ambiguity and overlap between genres. The project's classification model may not always accurately capture the nuances of certain songs or satisfy individual users' genre preferences. Additionally, if the training data is limited in representing diverse music genres, the classification and recommendation accuracy may be impacted.

- User Interface Customization: The project's user interface provided in the code snippets is limited and may not offer extensive customization options. Additional development may be required to enhance the user interface, improve user experience, and provide more interactive features.

- Data Availability and Quality: The project relies on the availability and quality of training data for music genre classification and recommendation. If the training data is limited or of low quality, it may impact the accuracy and reliability of the system's predictions and recommendations.

1.4 Timeline of the Project:

Sr.no	**Work**	**Start Date**	**End Date**
1	Information gathering and literature review	03/07/2022	15/08/2022
2	Data collection and preprocessing	18/08/2022	29/08/2022
3	Feature extraction and selection techniques for representing audio data	2/09/2022	14/09/2022
4	Implementation of machine learning algorithms for music genre classification	15/09/2022	4/11/2022
5	Development of the recommendation system	5/11/2022	26/01/2023
6	Evaluation and performance testing of the classification and recommendation models	27/01/2023	20/03/2023
7	Final Documentation	21/03/2023	08/06/2023

1.5 Project Management Plan:

The project will follow a structured management plan to ensure its successful execution. Key activities include:

- Task assignment and delegation.
- Regular progress tracking and updates.
- Communication and coordination among team members.
- Risk assessment and mitigation strategies.
- Quality assurance and testing procedures.
- Timely completion of milestones and deliverables.

1.5.1 Milestone List:

Milestone	Description	Start Date	End Date	Duration In Days	Priority
Literature Review and Research	Conduct a comprehensive review of existing literature and research on music genre classification and recommendation systems.	03/07/2022	18/07/2022	15	High
Dataset Collection and Preprocessing	Collect a diverse dataset of music tracks from various genres and preprocess the data for further analysis	19/07/2022	08/08/2022	20	High
Feature Extraction and Representation	Apply feature extraction techniques to represent audio data effectively, capturing relevant musical characteristics.	09/08/2022	24/08/2022	15	Medium
Machine Learning Model Implementation and Training	Implement and train machine learning models for music genre classification using the preprocessed dataset and extracted features.	25/08/2022	19/09/2022	25	High
Evaluation and Performance	Evaluate the performance of the trained	20/09/2022	5/10/2022	15	Medium

Testing	classification model using appropriate performance metrics and testing procedures.				
Recommendation System Development	Develop a recommendation system that suggests music genres based on user preferences and interactions.	06/10/2022	26/11/2022	20	High
Integration and System Testing	Integrate the music genre classification and recommendation system components and conduct thorough testing to ensure proper functionality.	27/11/2022	16/02/2023	20	High
Documentation and Report Writing	Document the project's methodology, findings, and outcomes, and write the final project report.	03/04/2023	25/05/2023	15	Medium
Project Presentation and Submission	Prepare and deliver a project presentation, and submit the final report and any other required deliverables.	27/05/2023	12/06/2023	15	Medium

1.6 Project Cost

COCOMO MODEL:
Effort Estimation:

Effort (in person-months) = 2.4 * (LOC) ^ 1.05

Applying this formula with 700 lines of code:

Effort = 2.4 * (700) ^ 1.05
= 5.81 person-months (approximately)

Duration Estimation:
Once the effort estimation is obtained, we can estimate the duration of the project. Assuming a standard productivity rate of 2 person-months per calendar month, the duration can be calculated as:

Duration (in months) = Effort / Productivity

Duration = 5.81 / 2
= 2.91 months (approximately)

2. Background study and literature overview

2.1 Literature overview:

Systems for categorizing music genres and making recommendations have been the focus of in-depth academic and professional research. Due to the expanding diversity of music genres, the growing number of music streaming services, and the demand to offer customers individualized music suggestions, the classification of music genres has gained importance in recent years. Systems for categorizing and recommending music genres use a variety of techniques. One method relies on the aural characteristics of the song itself and employs methods like signal processing and machine learning algorithms. A different strategy is focused on textual data about the music, such as metadata, bios of the artists, or reviews.

To categories music into different genres, audio-based categorization systems use elements including tempo, rhythm, melody, and harmony. Classification models are frequently trained using machine learning algorithms including support vector machines, neural networks, and decision trees. For instance, utilizing a combination of audio data and textual information about the music, genre classification models have been trained using the Million Song Dataset (MSD). Textual information about the music, such as artist names, album titles, song titles, and user-generated metadata, is the foundation of text-based classification systems. These systems use information retrieval and natural language processing to extract pertinent data and categorise music into different genres. Last.fm and Pandora are two well-known instances of text-based classification algorithms.

Utilizing algorithms, music recommendation systems offer users music based on their listening interests and behaviors. Both collaborative filtering and content-based filtering are frequently used methods for recommending music. Individual users are given music recommendations using collaborative filtering, which makes use of information on listening preferences and user reviews. Content-based filtering generates suggestions based on facts about the music itself, including audio aspects and metadata. Overall, by offering individualized and pertinent music recommendations, music genre classification and recommendation systems have the potential to enhance consumers' musical listening experiences. These technologies will continue to be honed and improved by ongoing study in the subject, giving consumers access to a wider variety of interesting music

2.2 Critical appraisal of other people’s work:

In recent years, researchers have become interested in the development of systems for classifying and recommending music genres. The current body of literature does, however, have several drawbacks and holes that need to be filled. As an illustration, some researches have concentrated on using just one kind of feature or method for classification, which may not be enough for precise genre identification. Furthermore, a lot of recommendation systems only use user ratings, which are subject to bias and are not always accurate. Therefore, it is crucial that future studies investigate more robust and diverse methods of identifying and recommending musical genres.

There is still much to learn about how cultural and social elements interact with other variables like age, gender, and location, despite the fact that several research have looked at the influence of these aspects on music tastes. Further investigation is also required on the ethical implications of music recommendation algorithms, particularly with regard to concerns over data security and privacy. To create more accurate and inclusive music genre classification and recommendation algorithms, interdisciplinary and collaborative research in this area is needed, according to a critical analysis of the existing literature.

2.3 Investigation of current project and related work:

There is still much to learn about how cultural and social elements interact with other variables like age, gender, and geography, despite the fact that several research have looked at the influence of these aspects on music tastes. Further investigation is also required on the ethical implications of music recommendation algorithms, particularly with regard to concerns over data security and privacy. In order to create more accurate and inclusive music genre classification and recommendation algorithms, interdisciplinary and collaborative research in this area is needed, according to a critical analysis of the existing literature.

Utilizing spectral features like chroma features and Mel-Frequency Cepstral Coefficients (MFCCs) is one typical strategy. While chroma features record the harmonic content of the music, MFCCs record the spectral envelope of the audio stream. Another approach is to use deep neural networks such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to learn more complex features from the audio signal. On many music categorization tasks, these networks have been found to perform better than conventional techniques, especially when working with huge datasets.

In addition to categorizing music by genre, there has been an increase in interest in creating playlist-generating music recommendation systems that are based on user preferences. To suggest music that is like the user's listening history or preferences, these systems often use collaborative filtering algorithms. There are other methods as well, like content-based filtering, which suggests music based on elements like tempo, key, and timbre.

To improve performance, there has been a recent trend towards hybrid recommendation systems, which mix collaborative filtering and content-based filtering. To produce more reliable suggestions, these systems often combine user and item characteristics with collaborative filtering. Recommendation systems and music genre classification are crucial study fields in music information retrieval, to sum up. While numerous conventional approaches have been suggested, current developments in machine learning and deep neural networks have significantly enhanced these challenges. It is hopeful that the current trend towards hybrid recommendation systems, which integrate both collaborative filtering and content-based filtering, may eventually result in even greater performance.

3. Requirement analysis

3.1 Requirements Gathering:

1. User Requirements:
    - Users should be able to upload audio files in various formats (e.g., MP3, WAV) to the system for genre classification and recommendation
    - Users should receive the predicted genre label for each uploaded file
    - Users should be able to specify their preferred music genres or select from a predefined list of genres. Based on user preferences and selected genres, the system should generate personalized music genre recommendations
    - The system should allow users to customize the features used for genre recommendation. For example, users can adjust sliders for features like acousticness, dance ability, energy, instrumentalness, valence, and tempo to fine-tune their recommendations
    - The system should have a user-friendly interface that enables easy file upload, genre selection, and customization of features

2. Functional Requirements:
    1. Music Genre Classification:
        - Extract MFCC features from audio files.
        - Classify audio files into predefined music genres using a pre-trained CNN model.
        - Display the predicted genre to the user.

    2. Music Genre Recommendation:
        - Load a dataset of processed track information.
        - Allow users to customize genre, year range, and audio features.
        - Retrieve and display a list of recommended songs based on user preferences.
        - Provide audio playback and visualizations of audio features for recommended songs.

    3. User Interface:
        - Implement a Flask web application for music genre classification.
        - Use Streamlet for the music genre recommendation system.
        - Allow users to upload audio files for genre classification.
        - Provide a user-friendly interface for input customization and song recommendations.
    4. Model Management:
        - Load a pre-trained CNN model for music genre classification.
        - Save and load the model for future use.
        - Handle model inference on user-uploaded audio files.

3. Non-functional Requirements:

- The system should ensure the privacy and security of user data by implementing appropriate security measures, such as encryption and user authentication.
- The system should have an intuitive and user-friendly interface that makes it easy for users to navigate and interact with the system.
- The system should be scalable and able to handle a large number of users and data, with the ability to scale up as the user base grows.

4. System Requirements:

a) Hardware requirements:

i. RAM-8GB
ii. Hard disk-500GB
iii. Desktop machine

b) Software requirements:

i. Operating System -windows 11
ii. Python 3.6.5
iii. Libraries:

a. streamlit
b. pandas
c. plotly
d. scikit-learn
e. pip
f. tensorflow
g. librosa
h. flask
i. Tensorflow keras
j. Werkzeug

5. Testing Requirements:

- The system should be tested thoroughly to ensure that it meets all the functional and non-functional requirements.
- The system should be tested on multiple browsers and devices to ensure compatibility.
- The system should be tested with a large dataset of music tracks and user profiles to evaluate its performance.

3.2 Requirement Specification:

1. User Interface: The software should have a user-friendly interface for users to interact with and input their preferences.

2. Music Input: Users should be able to upload audio files in various formats for genre classification and recommendation.

3. Music Genre Classification: The software should classify the uploaded music files into specific genres using machine learning techniques and audio feature extraction.

4. Music Genre Recommendation: Based on user preferences and selected genres, the software should provide personalized music recommendations.

5. Customization: Users should be able to customize their music preferences by selecting genres and adjusting feature sliders.

6. Scalability: The software should be capable of handling multiple users simultaneously, subject to server capacity and resources.

7. Data Processing: The software should process the audio files, extract relevant features, and perform genre classification and recommendation tasks.

8. Integration: The software should integrate with external libraries and frameworks such as Flask, Streamlit, Tensorflow, and scikit-learn.

9. Model Training: The software should train machine learning models for genre classification using a pre-labeled dataset.

10. Performance: The software should be optimized for speed and responsiveness to ensure efficient classification and recommendation processes.

3.3 Use case:

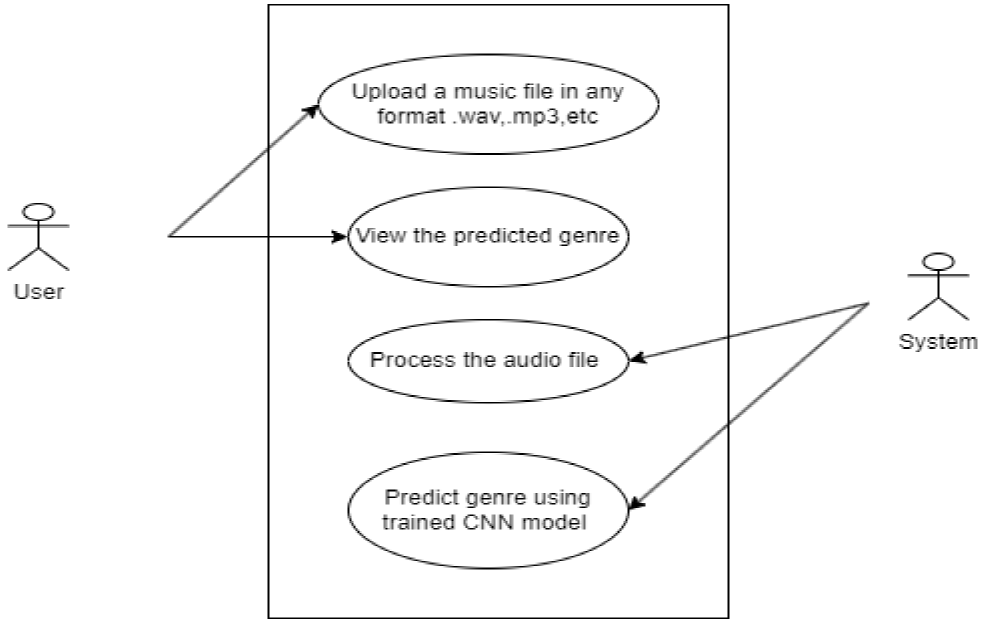

Upload a music file in any format .wav,.mp3,etc
View the predicted genre
Process the audio file
Predict genre using trained CNN model
User
System

Fig.3.3.1: Use case diagram for classification

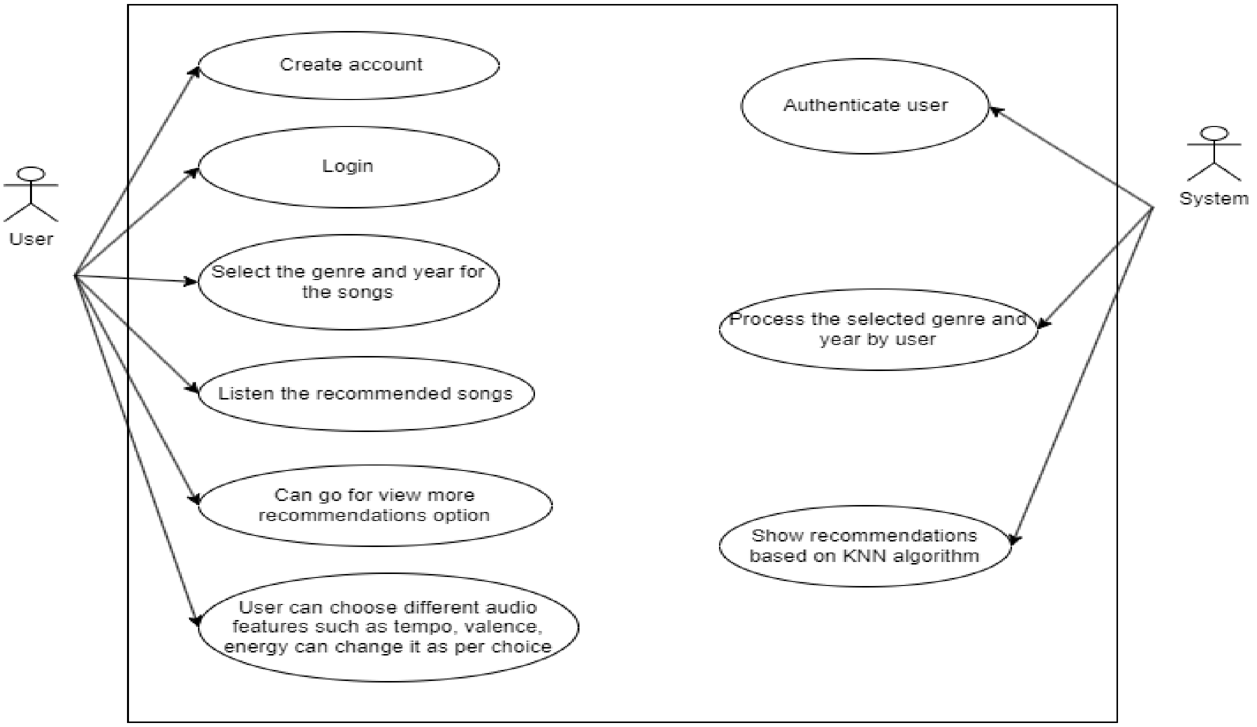

Create account
Login
Select the genre and year for the songs
Listen the recommended songs
Can go for view more recommendations option
User can choose different audio features such as tempo, valence, energy can change it as per choice
Authenticate user
Process the selected genre and year by user
Show recommendations based on KNN algorithm
User
System

Fig.3.3.2: Use case diagram for Recommendation

4. SYSTEM DESIGN

4.1 Architecture Design

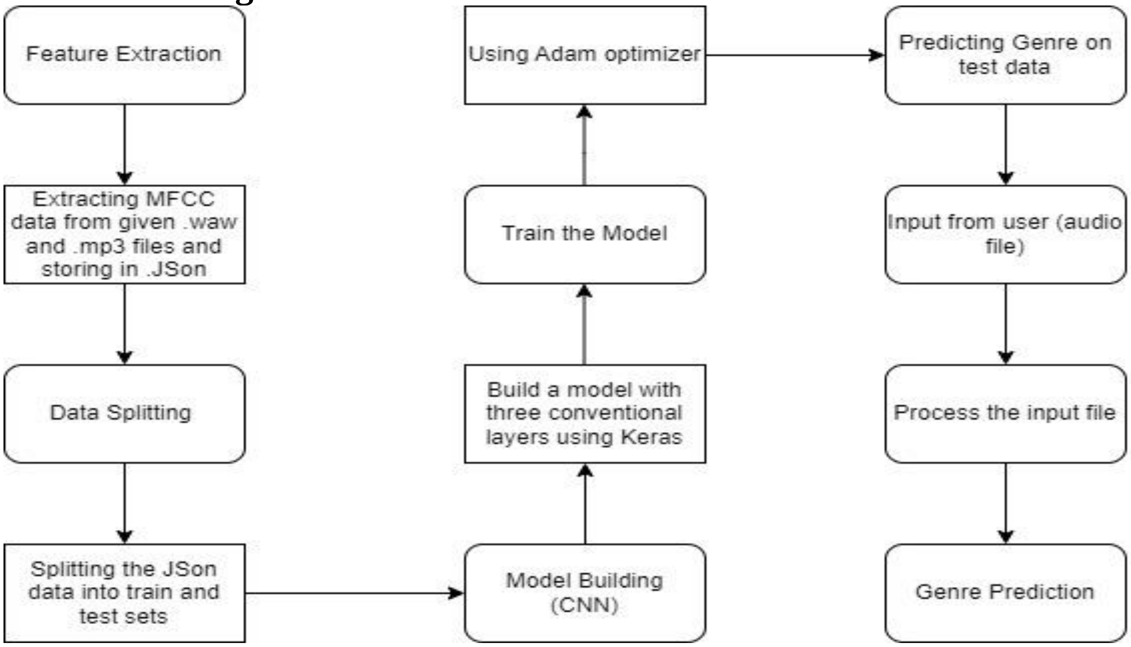

Feature Extraction
Extracting MFCC data from given .waw and .mp3 files and storing in .JSon
Data Splitting
Splitting the JSon data into train and test sets
Model Building (CNN)
Build a model with three conventional layers using Keras
Train the Model
Using Adam optimizer
Predicting Genre on test data
Input from user (audio file)
Process the input file
Genre Prediction

Fig.4.1.1: Classification

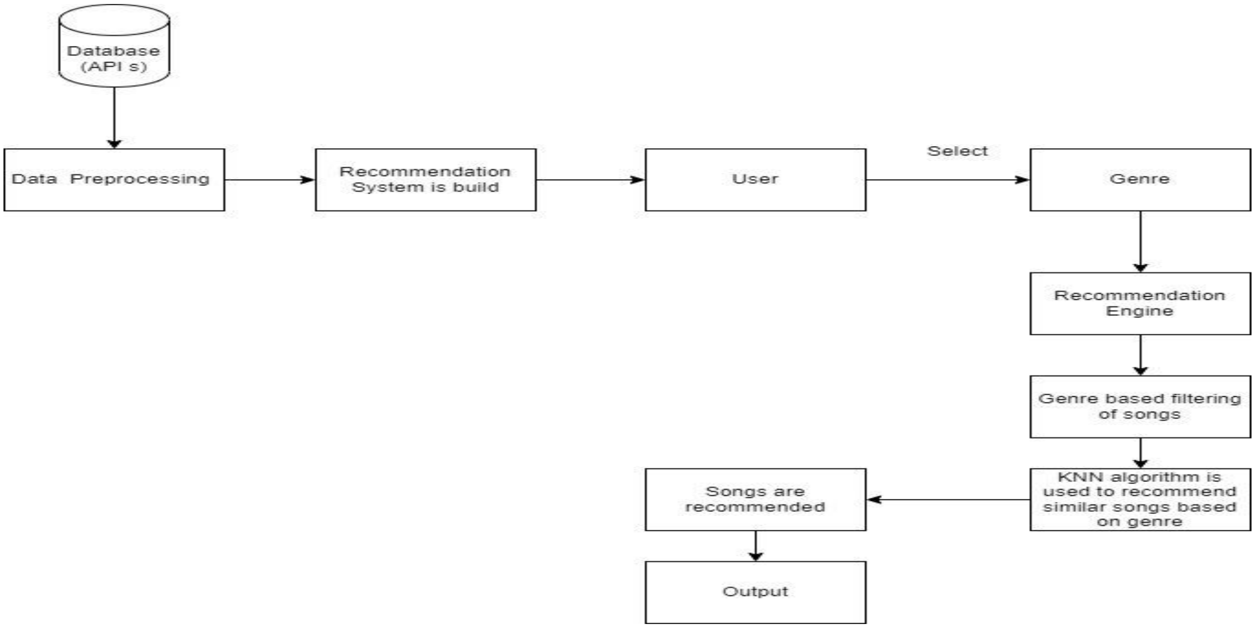
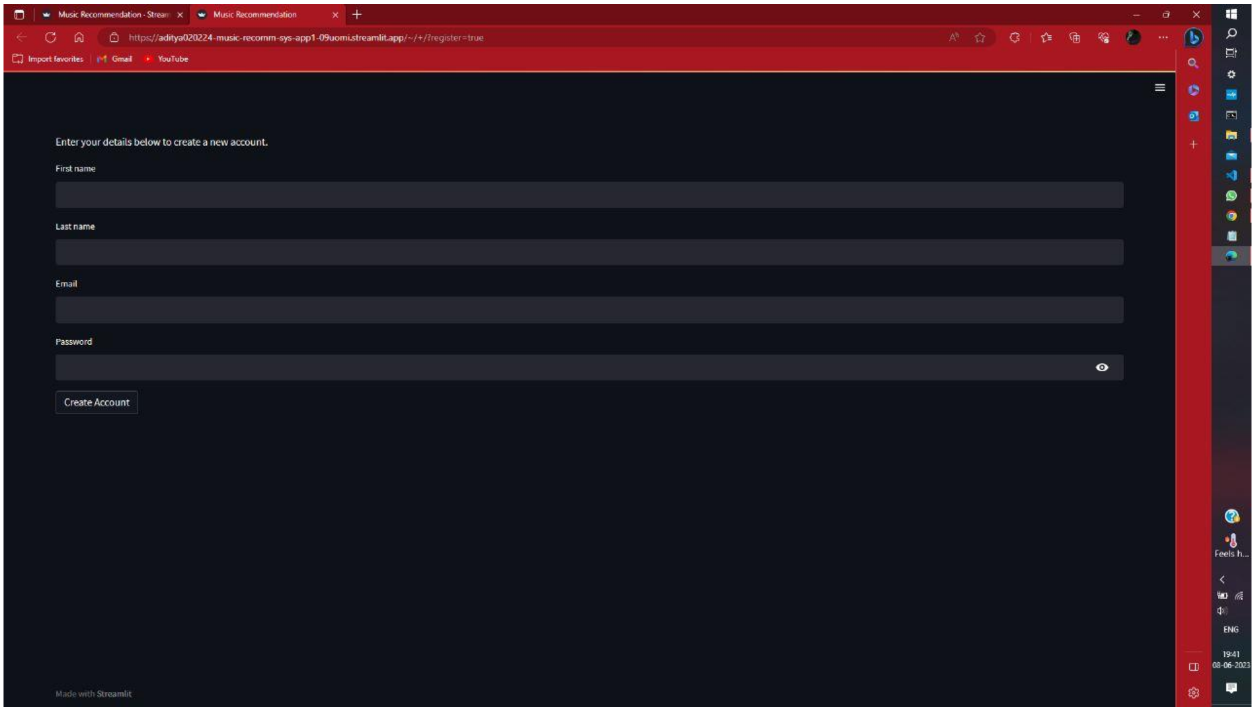

Database (API s)
Data Preprocessing
Recommendation System is build
User
Select
Genre
Recommendation Engine
Genre based filtering of songs
KNN algorithm is used to recommend similar songs based on genre
Songs are recommended
Output

Fig.4.1.2: Recommendation

4.2 User Interface Design:


Music Recommendation · Stream
Music Recommendation
https://aditya020224-music-recomm-sys-app1-09uomi.streamlit.app/~/+/?register=true
Import favorites
Gmail
YouTube
Feels h...
ENG
19:41
08-06-2023

Enter your details below to create a new account.

First name

Last name

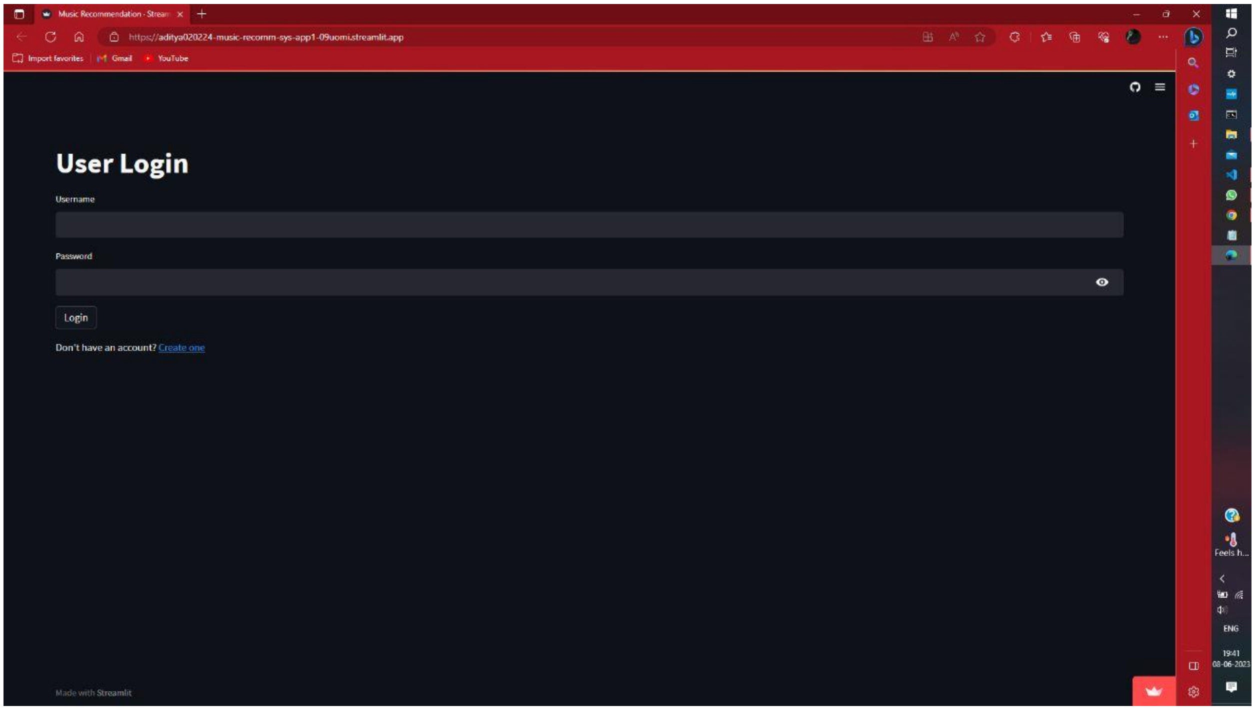
Email

Password

Create Account

Made with Streamlit

Fig.4.2.1: Create Account


Music Recommendation · Stream
https://aditya020224-music-recomm-sys-app1-09uomi.streamlit.app
Import favorites
Gmail
YouTube
Feels h...
ENG
19:41
08-06-2023

User Login

Username

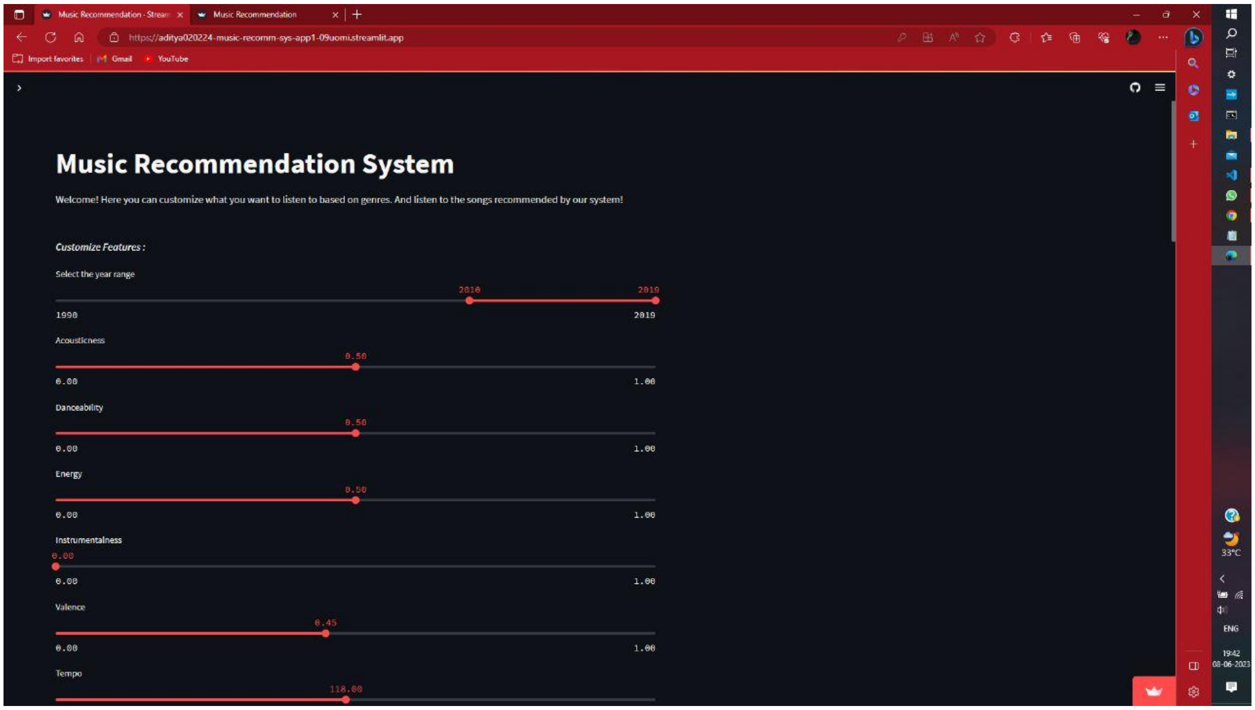
Password

Login

Don't have an account? Create one

Made with Streamlit

Fig.4.2.2: User Login


Music Recommendation - Stream
Music Recommendation
https://aditya020224-music-recomm-sys-app1-09uomi.streamlit.app
Import favorites
Gmail
YouTube

Music Recommendation System

Welcome! Here you can customize what you want to listen to based on genres. And listen to the songs recommended by our system!

Customize Features :

Select the year range: 2010 – 2019 (1990 – 2019)

Acousticness: 0.50 (0.00 – 1.00)

Danceability: 0.50 (0.00 – 1.00)

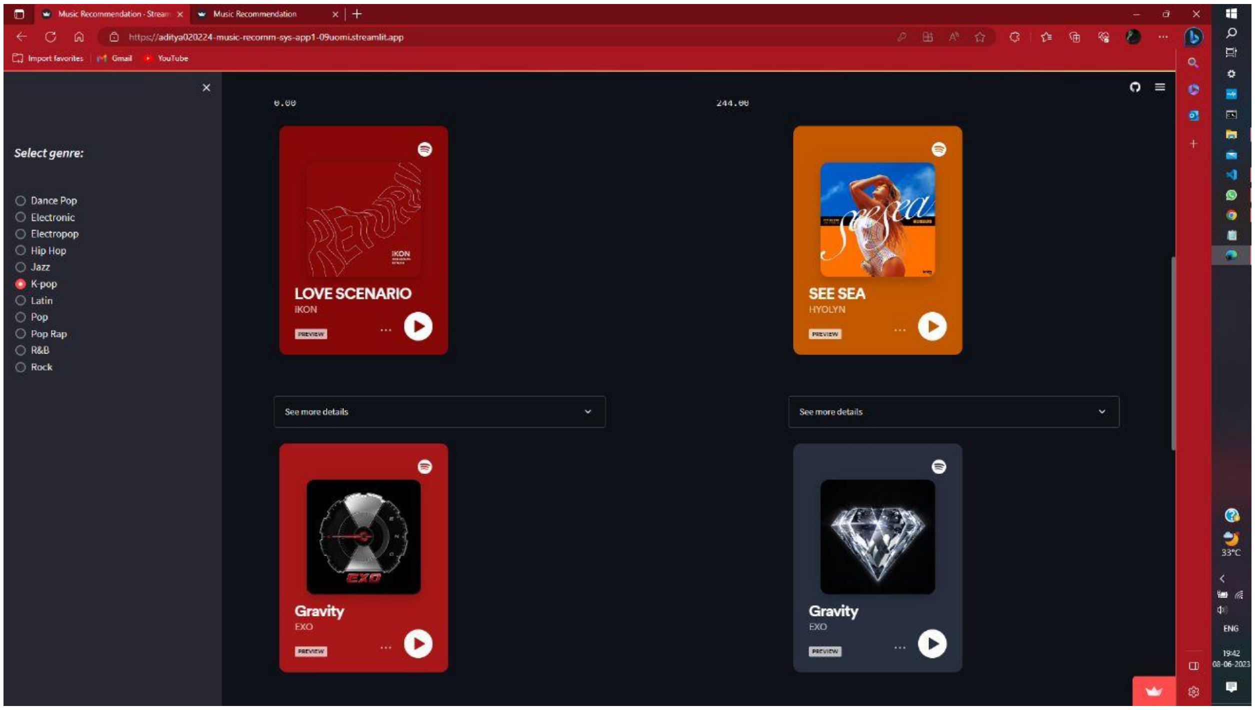
Energy: 0.50 (0.00 – 1.00)

Instrumentalness: 0.00 (0.00 – 1.00)

Valence: 0.45 (0.00 – 1.00)

Tempo: 118.00

Fig.4.2.3: Music Recommendation System


Music Recommendation - Stream
Music Recommendation
https://aditya020224-music-recomm-sys-app1-09uomi.streamlit.app
Import favorites
Gmail
YouTube
0.00
244.00
PREVIEW
See more details

Select genre:

○ Dance Pop
○ Electronic
○ Electropop
○ Hip Hop
○ Jazz
● K-pop
○ Latin
○ Pop
○ Pop Rap
○ R&B
○ Rock

LOVE SCENARIO – iKON

SEE SEA – HYOLYN

Gravity – EXO

Gravity – EXO

Fig.4.2.4: Recommendation Based on Genre

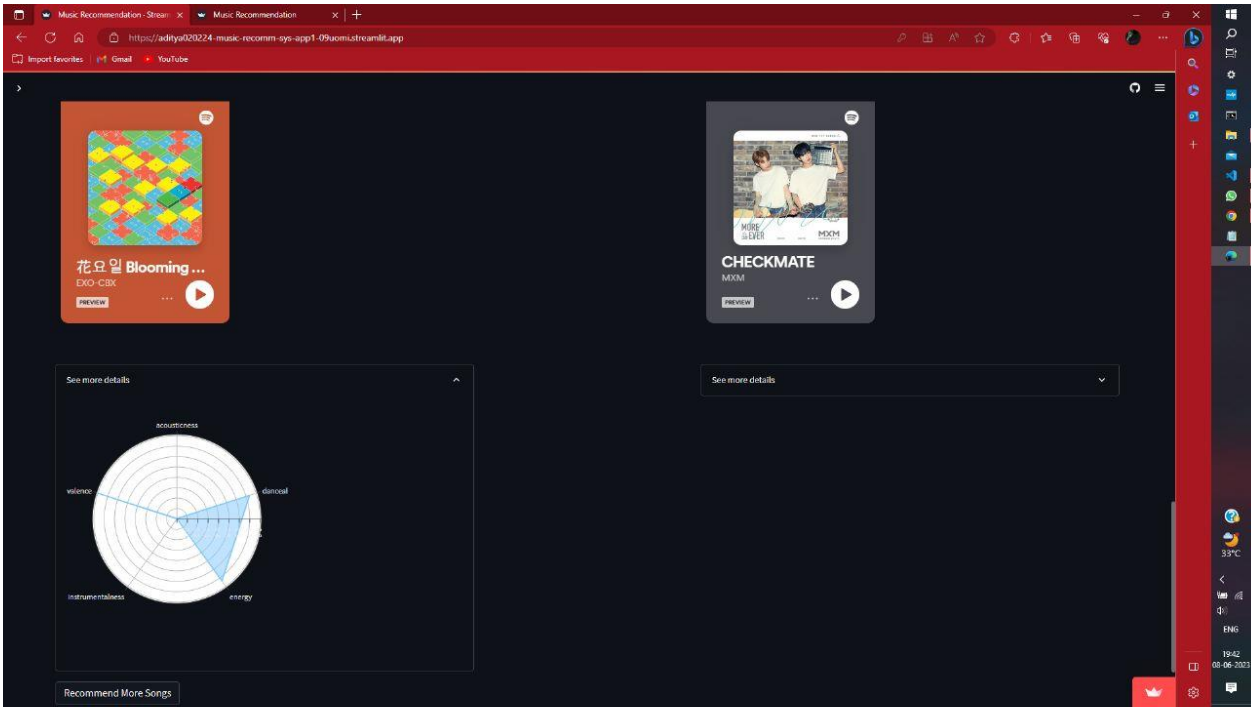
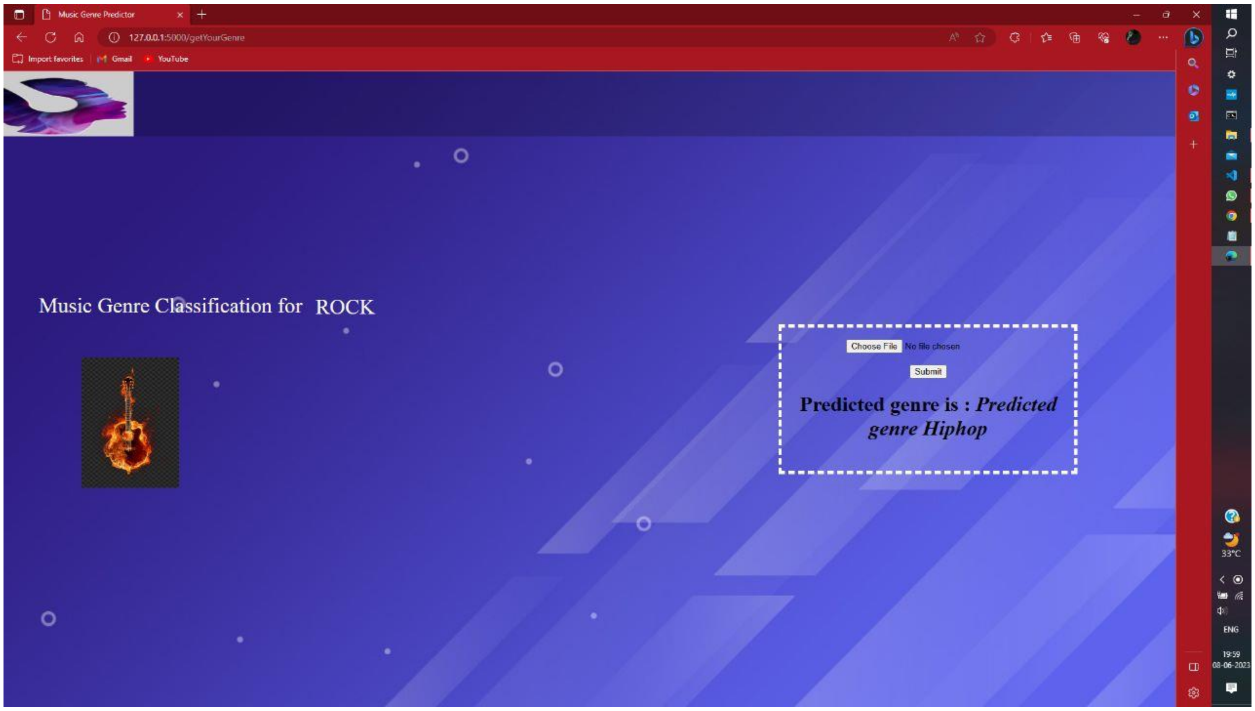


Fig.4.2.5: Details of Recommended Song



Music Genre Classification for ROCK

Predicted genre is : ***Predicted genre Hiphop***

Fig.4.2.6: Music Genre Classification System

4.3 Algorithmic Description of each module

1. Music Genre Classification Module:

   Input: Audio file in various formats.

   Algorithm:

   - Preprocess the audio file by resampling it to a fixed sample rate.
   - Divide the audio file into segments.
   - Extract MFCC (Mel-frequency cepstral coefficients) features from each segment.
   - Transform the MFCC features into a suitable format for classification.
   - Load a pre-trained CNN (Convolutional Neural Network) model for genre classification.
   - Pass the transformed features through the CNN model to obtain genre predictions.
   - Identify the genre with the highest prediction probability as the classified genre.

   Output: Predicted music genre.

2. Music Genre Recommendation Module:

   Inputs:

   - User preferences (selected genres and feature sliders).
   - Start and end years for filtering music tracks.
   - Sound characteristics (acoustic, energy, instrumentation, valence, etc.).

   Algorithm:

   - Retrieve a dataset of music tracks with associated genres and audio features.
   - Filter the dataset based on user preferences, start and end years, and selected genres.
   - Use the filtered dataset to train a Nearest Neighbors model.
   - Generate feature vectors based on user-selected audio features.
   - Find the nearest neighbors in the feature space using the Nearest Neighbors model.
   - Retrieve the recommended music tracks (URIs) from the nearest neighbors.
   - Display the recommended tracks and associated audio features.

   Output: Recommended music tracks and associated audio features.

4.4 System Modeling:

4.4.1 Dataflow Diagram:

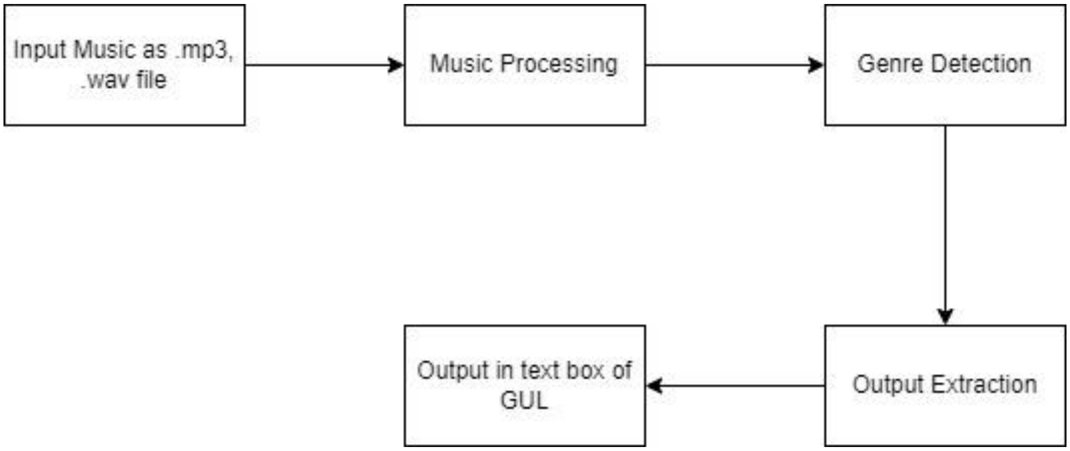


Fig.4.4.1.1: Dataflow Block Diagram

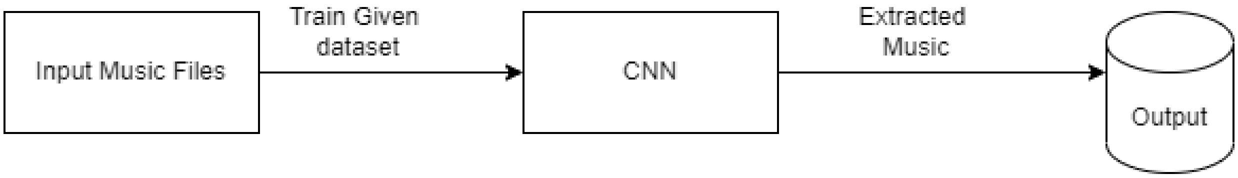


Fig.4.4.1.2: Dataflow Diagram

4.4.2 Sequence Diagram:

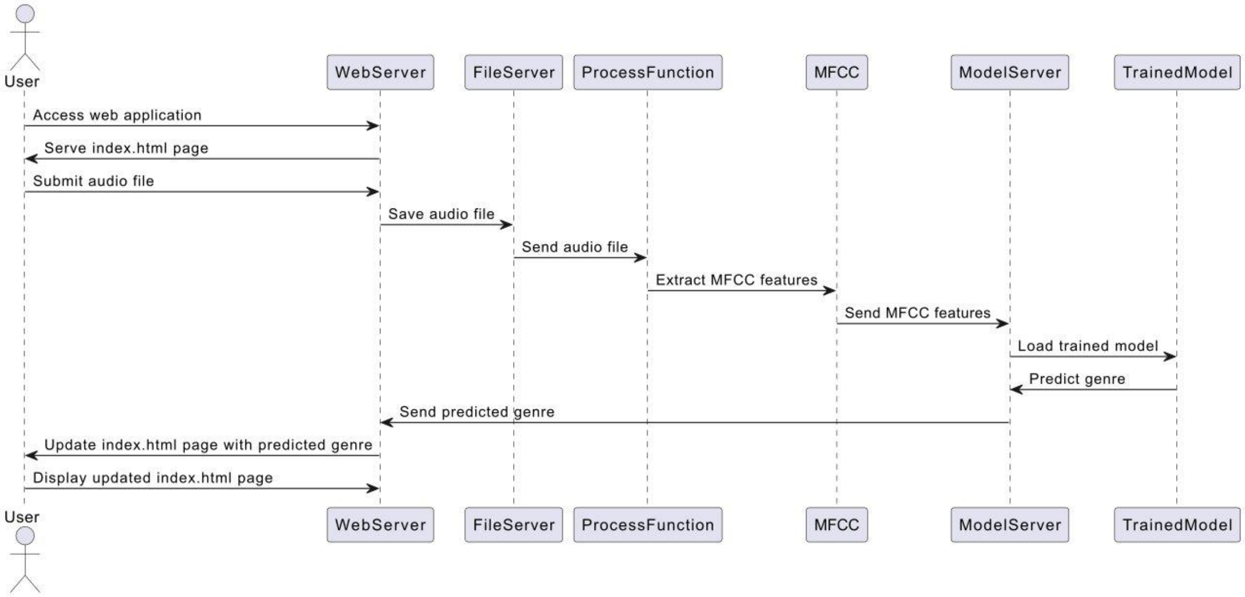


Fig.4.4.2.1: Sequence Diagram for classification

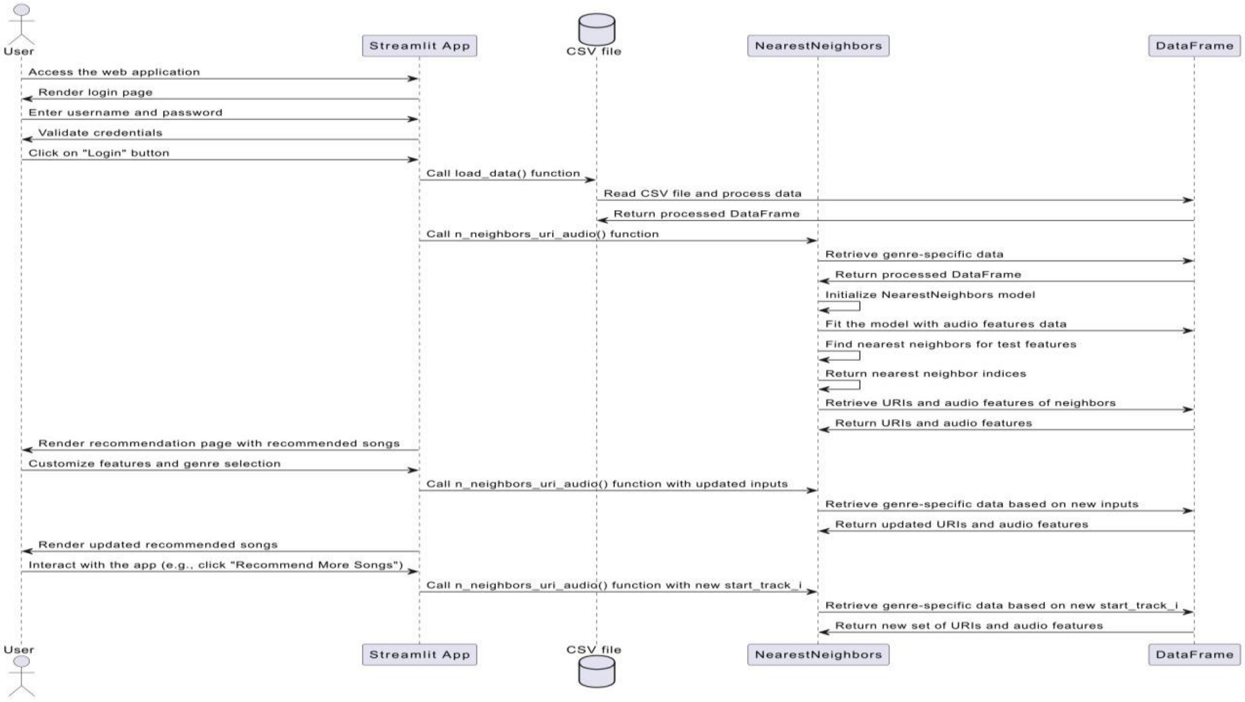

User
Streamlit App
CSV file
NearestNeighbors
DataFrame
Access the web application
Render login page
Enter username and password
Validate credentials
Click on "Login" button
Call load_data() function
Read CSV file and process data
Return processed DataFrame
Call n_neighbors_uri_audio() function
Retrieve genre-specific data
Return processed DataFrame
Initialize NearestNeighbors model
Fit the model with audio features data
Find nearest neighbors for test features
Return nearest neighbor indices
Retrieve URIs and audio features of neighbors
Return URIs and audio features
Render recommendation page with recommended songs
Customize features and genre selection
Call n_neighbors_uri_audio() function with updated inputs
Retrieve genre-specific data based on new inputs
Return updated URIs and audio features
Render updated recommended songs
Interact with the app (e.g., click "Recommend More Songs")
Call n_neighbors_uri_audio() function with new start_track_i
Retrieve genre-specific data based on new start_track_i
Return new set of URIs and audio features
User
Streamlit App
CSV file
NearestNeighbors
DataFrame

Fig.4.4.2.2: Sequence Diagram for Recommendation

4.4.3 Activity Diagram:

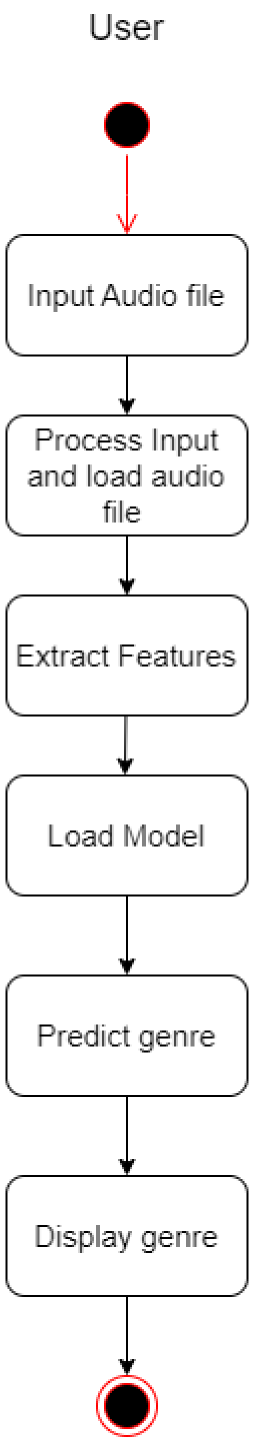


Fig.4.4.3.1: Activity Diagram for classification

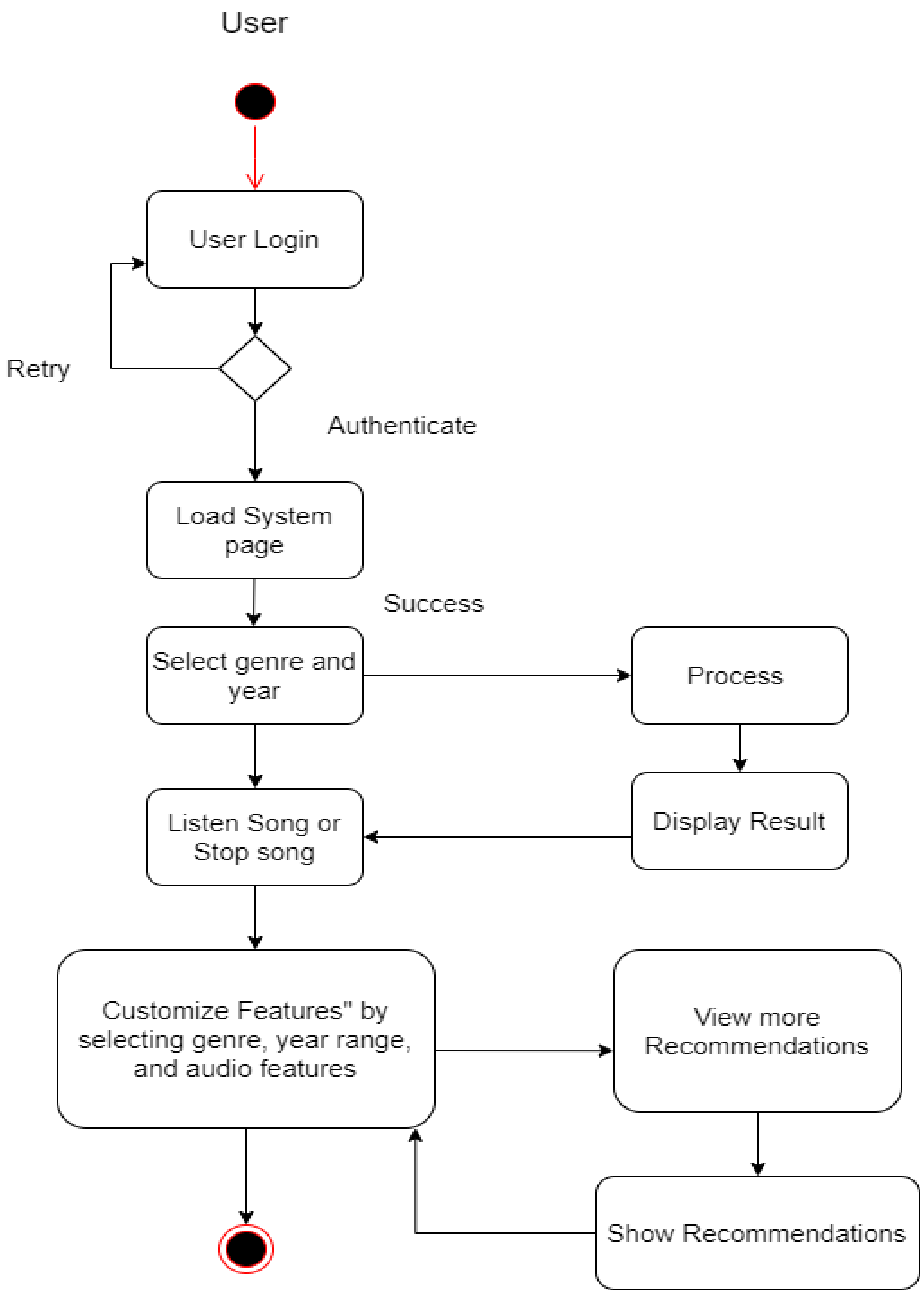


Fig.4.4.3.2: Activity Diagram for recommendation

4.4.4. Component Diagram:

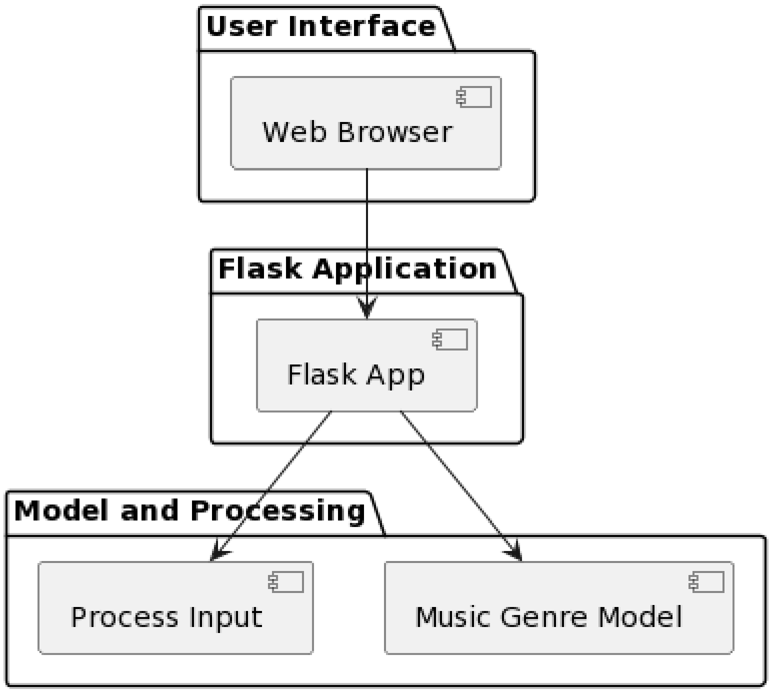


Fig.4.4.4.1: Component Diagram for classification

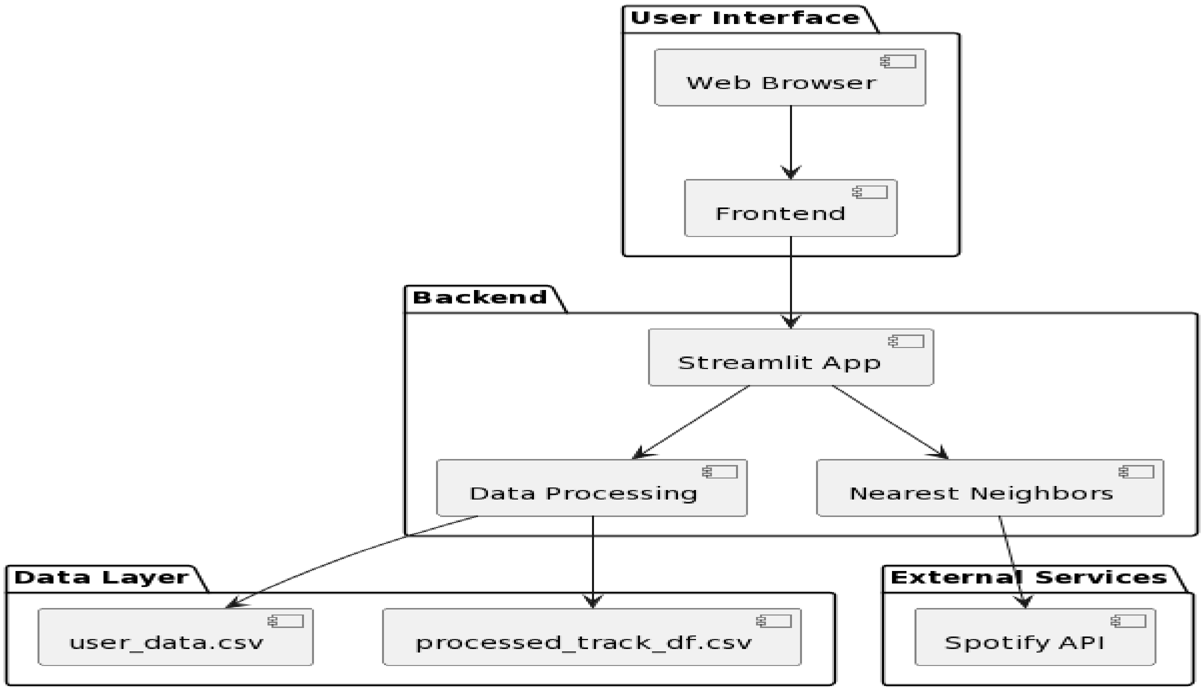


Fig.4.4.4.2: Component Diagram for recommendation

4.4.5 Deployment Diagram:

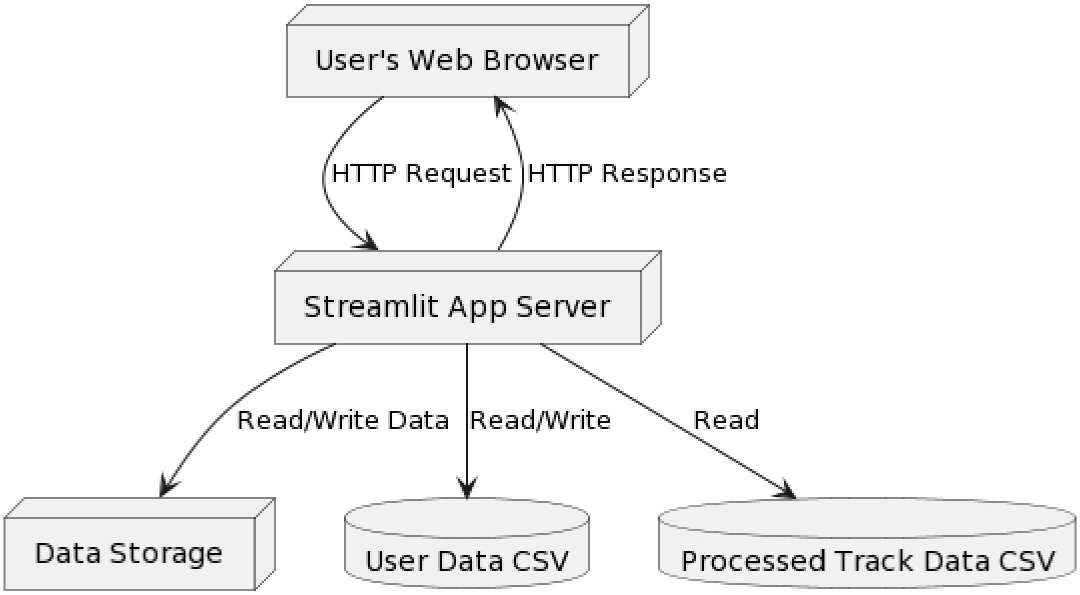


Fig.4.4.5.1: Deployment Diagram

5. Implementation

5.1 Environmental Setting for Running the Project

1. Programming Language: Python 3.x
2. Libraries and Frameworks:
3. Flask (for the classification part)
4. TensorFlow and Keras (for the classification part)
5. Librosa (for audio processing in the classification part)
6. Streamlit (for the recommendation part)
7. Pandas, NumPy, Scikit-learn, Plotly (for data manipulation and visualization in the recommendation part)
8. Werkzeug (for file handling in Flask)
9. Machine Learning Model:
10. "Music_Genre_10_CNN.h5" (a pre-trained CNN model for music genre classification)
11. Data:
12. Processed_track_df.csv (a dataset containing track information and audio features for recommendation)
13. user_data.csv (a CSV file to store user login information)

5.2 Detailed Description of Methods:

1. `process_input(audio_file, track_duration)`: This method processes the input audio file for music genre classification. It takes an audio file and the track duration as input and performs the following steps:

- Loads the audio file using Librosa library.
- Divides the audio into segments.
- Extracts Mel-frequency Cepstral coefficients (MFCC) features from each segment.
- Returns the MFCC features.

2. `predict()`: This method is the route for the main page of the Flask web application. It renders the index.html template.

3. `getGenre()`: This method is the route for genre prediction in the Flask web application. It performs the following steps:

- Retrieves the uploaded audio file from the request.
- Calls the `process_input()` method to extract MFCC features from the audio file.
- Loads the pre-trained music genre classification model.
- Predicts the genre using the model based on the extracted features.
- Returns the predicted genre to be displayed on the index.html template.

4. `load_data()`: This method is used in the recommendation part of the Streamlit application. It loads the processed_track_df.csv file containing track information and audio features.

5. `n_neighbors_uri_audio(genre, start_year, end_year, test_feat)`: This method is used in the recommendation part of the Streamlit application. It takes genre, start year, end year, and test features as input and performs the following steps:

- Filters the tracks based on the given genre, start year, and end year.
- Performs nearest neighbor search using the test features and audio features of the filtered tracks.
- Returns the URIs and audio features of the nearest neighbors.

6. `login_page()`: This method is used in the Streamlit application to display the user login page. It takes username and password as input and checks if the provided credentials are valid.
7. `register_page()`: This method is used in the Streamlit application to display the user registration page. It takes user details such as first name, last name, email, and password as input and creates a new user account.
8. `recommendation_page()`: This method is the main recommendation page in the Streamlit application. It allows users to customize their music preferences and displays recommended songs based on the selected genre and audio features.
9. `main()`: This method is the entry point of the Streamlit application. It checks if the user is logged in and redirects to the appropriate page (login, registration, or recommendation).

5.3 Implementation Details:

The implementation details of the project include:

1. Backend Implementation:

a. The backend is implemented using the Flask framework for the music genre classification part.
b. The Flask routes are defined to handle different requests and render appropriate templates.
c. The `process_input()` method is used to extract MFCC features from the input audio file.
d. The pre-trained music genre classification model is loaded using Keras.
e. The predicted genre is returned and displayed on the web interface.

2. Frontend Implementation:

a. The frontend is implemented using HTML templates and Flask's templating engine.
b. The main page is rendered using the `index.html` template, which displays an interface for uploading audio files.
a. The predicted genre is dynamically updated on the web page after the classification is performed.

3. Recommendation System Implementation:

a. The recommendation system is implemented using the Streamlit framework.
b. The `load_data()` method loads the track data from the processed_track_df.csv file.
c. The `n_neighbors_uri_audio()` method performs nearest neighbor search based on the selected genre and audio features.

d. The `recommendation_page()` method displays the recommendation interface, allowing users to customize their preferences and showing recommended songs.

4. User Authentication:

   a. User authentication is implemented using a login page and registration page in the Streamlit application.
   b. User credentials are stored in a CSV file named "user_data.csv".
   c. User input is validated to ensure proper registration and login.

5. External Libraries:

a. The project relies on external libraries such as Librosa for audio processing, TensorFlow and Keras for the music genre classification model, and Streamlit for the recommendation system.

6. Integration and Testing

6.1 Description of the Integration Modules:

1. Flask Integration Module:
   - This module integrates the Flask framework into the project.
   - It handles HTTP requests and routes them to appropriate functions.
   - It allows the project to have a web interface for uploading audio files and displaying the predicted music genre.

2. Keras Integration Module:
   - This module integrates the Keras library into the project.
   - It loads the pre-trained music genre classification model.
   - It allows the project to make predictions on audio files and determine their genre.

3. Librosa Integration Module:
   - This module integrates the Librosa library into the project.
   - It provides functions for audio processing, such as extracting MFCC features.
   - It allows the project to process audio files and extract relevant features for genre classification.

4. Streamlit Integration Module:
   - This module integrates the Streamlit framework into the project.
   - It handles the user interface for the recommendation system.
   - It allows users to customize their preferences and displays recommended songs based on selected genres and audio features.

5. Pandas Integration Module:
   - This module integrates the Pandas library into the project.
   - It handles the loading and manipulation of data from CSV files.
   - It allows the project to load the track data and perform data exploration in the recommendation system.

6.2 Testing:

Sr.no	Test Case Description	input	expected output	Actual output	status
1	Music genre classification - Valid audio file	Valid audio file	Predicted genre: Rock	Predicted genre: Rock	Passed
2	Music genre classification - Invalid audio file	Invalid audio file	Error message: Invalid file format	Error message: Invalid file format	Passed
3	Music recommendation - Valid inputs	Genre: Pop, Year range: 2010-2019, Audio features: [0.5, 0.7, 0.8, 0.2, 0.6, 120]	Recommended songs based on inputs	Recommended songs based on inputs	Passed
4	Music recommendation - Invalid genre	Genre: Unknown	Error message: Genre not found	Error message: Genre not found	Passed
5	Music recommendation - Invalid year range	Year range: 2022-2025	Error message: No songs found for the given year range	Error message: No songs found for the given year range	Passed

7. Performance Analysis

Performance analysis is important to evaluate the efficiency and responsiveness of the song recommendation engine. Some aspects to consider for performance analysis are:

- Scalability: Test the system's performance with varying data sizes and user loads. Assess how the system handles a large number of users and data points without compromising performance.

- Resource Utilization: Monitor the system's resource utilization, such as CPU and memory usage, during different operations. Optimize resource consumption to ensure efficient performance.

- User Feedback: Collect user feedback on the responsiveness and overall performance of the system. Identify all areas for improvement and adjust if necessary.

- Response Time: Measure the time taken by the system to generate recommendations based on user input. Ensure that the response time is within acceptable limits to provide a seamless user experience.

8. Future Scope

The song recommendation engine can be further enhanced and expanded in several ways:
Additional Features: Include more audio features for customization, such as loudness, speechiness, and liveness. This can provide users with a wider range of options to personalize their song recommendations.

- User Profiles: Implement user profiles to allow users to save their preferences and receive personalized recommendations based on their listening history and preferences.

- Collaborative Filtering: Incorporate collaborative filtering techniques to recommend songs based on the listening habits and preferences of similar users. This can improve accuracy and range of view.

- Genre Exploration: Provide users with the ability to explore different genres and subgenres, allowing them to discover new music based on their interests.

- Enhanced Visualization: Improve the visualizations of the recommended songs and audio feature radar charts to provide a more engaging and informative user experience.

9. Applications of the Project

The song recommendation engine can have various applications in the music industry and beyond:

- Music Streaming Platforms: The recommendation engine can be integrated into music streaming platforms to provide personalized song recommendations to users based on their preferences and listening history.

- Radio Stations: Radio stations can use the recommendation engine to curate playlists and create personalized radio stations for listeners.

- Event Organizers: Event organizers can utilize the recommendation engine to suggest suitable songs for different types of events, such as parties, weddings, and corporate functions.

- Music Discovery Platforms: The engine can be used in music discovery platforms to help users explore new artists, genres, and songs based on their preferences. It can assist in expanding users' music knowledge and introducing them to diverse music options.

- Personalized Music Apps: The recommendation engine can be integrated into personalized music apps, allowing users to customize their music experience and receive tailored recommendations based on their mood, activity, or specific preferences.

- Advertising and Marketing: The engine's ability to analyze user preferences and recommend songs can be utilized by advertisers and marketers to target specific music genres or artists in their campaigns. It can help in delivering relevant and engaging music-related advertisements to the target audience.

- Music Research and Analysis: Researchers and analysts in the music industry can leverage the recommendation engine to study trends, patterns, and user preferences. It can aid in market analysis, genre popularity assessment, and understanding user behavior in the music consumption landscape.

- Overall, the song recommendation engine has broad applications across various sectors, including music streaming, event management, personalized apps, marketing, and research. Its ability to provide personalized song recommendations enhances the user experience and facilitates music discovery in a tailored and engaging manner.

10. Installation Guide and User Manual

Installation Guide:

1. System Requirements:
    - Specify the minimum hardware requirements, such as processor, RAM, and storage.
    - Specify the operating system and any dependencies required.
2. Install Dependencies:
    - List the software dependencies required for the system to run.
    - Provide instructions for installing each dependency, including any specific versions or configurations.
3. Download the System:
    - Provide a link or instructions for downloading the system files.
    - Specify any additional data or models required for the system to function.
4. Configuration:
    - Explain the configuration files or settings that need to be modified, if any.
    - Provide instructions for updating the configuration based on the user's requirements.
5. Set Up Datasets:
    - Explain how to import or set up the music datasets for genre classification and recommendation.
    - Include instructions for organizing the dataset and any necessary preprocessing steps.
6. Train the Model:
    - Describe the process of training the genre classification model using the prepared dataset.
    - Specify any command-line or script-based instructions for initiating the training process.
7. Evaluate and Test:
    - Explain how to evaluate the trained model's performance on test data.
    - Provide instructions for running sample tests and interpreting the results.

User Manual:

1. Introduction:
    - Provide an overview of the music genre classification and recommendation system.
    - Explain the benefits and features of the system.
2. System Setup:
    - Explain the installation process, referring to the installation guide if needed.
    - Provide troubleshooting tips for common installation issues.
3. User Interface:
    - Describes the user interface components and functions.
    - Provide instructions for navigating the system and accessing different features.
4. Music Genre Classification:

- Explain how the system classifies music into different genres.
- Describe the input requirements for genre classification, such as audio files or metadata.

5. Music Recommendation:
    - Describe how the system recommends music based on user preferences or listening history.
    - Explain the recommendation algorithms or techniques used by the system.
6. Personalization:
    - Explain how users can personalize their recommendations.
    - Provide instructions for updating preferences, creating playlists, or favoriting specific genres.
7. Troubleshooting:
    - Address common issues or errors that users may encounter.
    - Provide solutions or suggest contacting support for assistance.
8. Frequently Asked Questions (FAQ):
    - Include a list of frequently asked questions and their answers related to the system.
9. Glossary:
    - Define any technical terms or acronyms used in the system.
10. References:
    - Include references to external resources, such as research papers or libraries used in the system

11. Plagiarism Report

Document Viewer

Similarity Index

7%

test

By: test

As of: Jun 9, 2023 5:27:41 PM
6,528 words - 40 matches - 32 sources

sources:

41 words / 1% - Internet from 05-Apr-2020 12:00AM

www.slideshare.net

19 words / < 1% match - Internet from 26-Jan-2023 12:00AM

www.slideshare.net

13 words / < 1% match - Internet from 21-Dec-2022 12:00AM

www.slideshare.net

32 words / < 1% match - from 04-Apr-2023 12:00AM

WWW.coursehero.com

11 words / < 1% match - Internet from 28-Feb-2021 12:00AM

www.coursehero.com

21 words / < 1% match - Internet from 07-Feb-2023 12:00AM

www.researchgate.net

14 words / < 1% match - Internet from 02-Feb-2023 12:00AM

www.researchgate.net

19 words / < 1% match - Internet from 30-Dec-2022 12:00AM

www.scribd.com

9 words / < 1% match - Internet from 22-Apr-2020 12:00AM

www.scribd.com

25 words / < 1% match - Crossref

Deepak S, B.G. Prasad. "Music Classification based on Genre using LSTM", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020

19 words / < 1% match - Internet from 29-Nov-2022 12:00AM

www.mdpi.com

18 words / < 1% match - Internet from 07-Feb-2019 12:00AM

textilevaluechain.com

18 words / < 1% match - from 26-May-2023 12:00AM

www.irjmets.com

16 words / < 1% match -

Vera Trejo Alejandro Castor. "Construccion de plataformas marinas de concreto apoyadas por gravedad", TESIUNAM, 1986

15 words / < 1% match - Internet from 14-Dec-2022 12:00AM

etd.lib.metu.edu.tr

12 words / < 1% match - Crossref

"Information and Communication Technology for Competitive Strategies (ICTCS 2020)", Springer Science and Business Media LLC, 2022

12 words / < 1% match - Crossref

K. Murugesh Babu, M. Selvadass, R. Somashekar. "Characterization of the conventional and organic cotton fibres", Journal of the Textile Institute, 2013

11 words / < 1% match - Internet from 25-Mar-2020 12:00AM

link.springer.com

11 words / < 1% match - from 26-May-2023 12:00AM

www.science.gov

10 words / < 1% match - Internet

Sanden, Christopher, University of Lethbridge. Faculty of Arts and Science. "An empirical evaluation of computational and perceptual multi-label genre classification on music / Christopher Sanden", Department of Mathematics and Computer Science, 2010

9 words / < 1% match - Internet from 25-Jan-2023 12:00AM

limbimoderne.lls.unibuc.ro

9 words / < 1% match - from 07-Jun-2023 12:00AM

www.ijert.org

9 words / < 1% match - Internet from 17-Oct-2010 12:00AM

www.nitk.ac.in

8 words / < 1% match - ProQuest

Alubala, Amos. "Toward a Secure Account Recovery: Machine Learning Based User Modeling for Protection of Account Recovery in a Managed Environment", Columbia University, 2023

8 words / < 1% match - Internet from 13-Aug-2022 12:00AM

careersdocbox.com

8 words / < 1% match - Internet from 23-Nov-2022 12:00AM

dspace.dtu.ac.in

8 words / < 1% match - from 28-May-2023 12:00AM

ijircce.com

8 words / < 1% match - from 23-May-2023 12:00AM

onshow.iadt.ie

8 words / < 1% match - Internet

Senjaya, Rudy. "Web-based library for student projects/theses and faculty research papers", CSUSB ScholarWorks, 2007

8 words / < 1% match - Internet from 18-Jan-2023 12:00AM

www.yumpu.com

6 words / < 1% match - Crossref

Hamid R. Arabnia. "Preface", Elsevier BV, 2015

4 words / < 1% match - from 28-Mar-2023 12:00AM

moam.info

12. Ethics

1. Surf the internet for personal and non-class related purposes during classes.
2. Make a copy of software for personal or commercial use.
3. Make a copy of software for friend.
4. Download printed software from internet.
5. Distribute printed software from internet.
6. Buy software with single user license and then install it on multiple computers.
7. Share printed copy of software.
8. Install a printed copy of software.

13. References and Bibliography

1. Dias, J., Pillai, V., Deshmukh, H., & Shah, A. (2022). Music Genre Classification & Recommendation System Using CNN. Available at SSRN 4111849.
2. Elbir, A., Çam, H. B., Iyican, M. E., Öztürk, B., & Aydin, N. (2018, October). Music genre classification and recommendation by using machine learning techniques. In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-5). IEEE.
3. Bahuleyan, H. (2018). Music genre classification using machine learning techniques. arXiv preprint arXiv:1804.01149.