RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Mini Project Report On

AUDENTIFY

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Adam Philip (U2103011)

Adish Rafique (U2103013)

Abhinav Shaji (U2103007)

Adhil Shibli (U2103012)

Under the guidance of

DR. RENU MARY DANIEL

Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
May 2024

CERTIFICATE

This is to certify that the mini project report entitled ***"Audentify"*** *is a bonafide record of the work done by* ***Adam Philip (U2103011)***, ***Adish Rafique (U2103013)***, ***Adhil Shibli (U2103012)***, ***Abhinav Shaji (U2103007)***, *submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr Renu Mary Daniel
Assistant Professor
Dept. of CSE
RSET

Mr Harikrishnan
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K. G.
Professor & Head of Department
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Head of the Department of Computer Science and Engineering for providing me with the opportunity to undertake my mini project, "Audentify".

I am highly indebted to my project coordinator, **Mr.Harikrishnan Sir**, Assistant Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Dr.Renu Mary Daniel** for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Adam Philip
Abhinav Shaji
Adish Rafique
Adhil Shibli

Abstract

Audentify is a web application designed for music genre classification using a Convolutional Neural Network (CNN) model. The system extracts key features from the audio data and feeds them into a trained CNN model to predict the music genre. Audentify was designed to simplify the process of music genre identification/discovery as music genres could be mixed up with one another in songs or not pronounced enough which complicates figuring out the genre for music lovers.

Audentify has many benefits in that it provides a user-friendly interface to classify music genres through file upload or real-time recording. It is flexible as the system can handle various audio input methods, catering to user preferences. The machine learning approach used provides accurate genre classification based on learned patterns from a labeled audio dataset.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN: Convolutional Neural Network

API: Application Programming Interface (provides a way for applications to interact with each other)

URL: Uniform Resource Locator (web address)

HTML: HyperText Markup Language (used for building web page structure)

CSS: Cascading Style Sheets (used for styling web pages)

JSON: JavaScript Object Notation (lightweight data format for data interchange)

Chapter 1

Introduction

1.1 Background

Music Genre identification can be a tricky process due to the large diversity of genres in music, furthermore the lack of any dedicated apps that offer this functionality makes it complicated for music lovers to identify/discover genres of music on the go. Audentify thus aims to help music lovers improve/diversify their music tastes by performing music genre detection.

1.2 Problem Definition

Current methods for music genre identification would require users to cross-reference external music databases to figure out the genre corresponding to a song which can be a time consuming multi-step process. Audentify addresses this issue by shortening the process to a single step with minimal actions to be taken by the user.

1.3 Scope and Motivation

The core functionality will provided as a web-application that recieves audio input from the user and displays the song's genre and suggests songs of a similar genre. Other features like artist recommendations (based on genre classified), customised playlist creation are also provided by the application.
The motivation for the application comes from the lack of existing applications that offer music genre detection capabilities and from the desire to make music genre identification/exploration a more straightforward process for music lovers

1.4 Objectives

1. Real-time Genre Detection: Accurately identify the genre of a song within seconds of receiving the audio input through the web interface.

2. High Accuracy: A high degree of accuracy in genre classification is expected , aiming for a specific target percentage (e.g., 85% or higher) for common music genres.

3. Similar Song/Artist Recommendations: Recommend songs similar to that of the detected genre and also provide artist recommendations based on the predicted genre

4. User-Friendly Interface: To create a web interface that is aesthetic and easy to navigate. This includes a simple way to upload or stream audio files and a clear presentation of the detected genre and recommended songs.

5. Scalability and Performance: Ensure the web application can handle various audio input formats and maintain responsiveness even with high traffic or complex music samples.

1.5 Challenges

The model is trained on a very large dataset which can increase the overhead on the application and make it less responsive whenever it's loaded. Genre classification might not always be accurate in certain instances where the provided song is of multiple genres.

1.6 Assumptions

It is assumed that users will have a device with an internet connection, that there is enough processing power on the device to run the genre classification model efficiently. It is also assumed that the training data is accurate and representative of a wide variety of music genres.

1.7 Societal / Industrial Relevance

1. Streamlined Music Discovery: By simplifying genre classification, Audentify allows users of all musical backgrounds to explore a wider range of music. This can lead to a better listening experience for music enthusiasts.

2. Explore/Improve Music Tastes: This tool can be valuable for people with limited knowledge of musical genres, helping them categorize and understand their music tastes.

1.8 Organization of the Report

In Chapter 1, the context for Audenitfy is specified by highlighting the current challenges in music genre identification and the need for a simple approach. Audentify's concept and core functionality is introduced here.
In Chapter 2, Audentify's requirements are mentioned in detail, such as the functional and non-functional requirements that need to be fulfilled for the user's needs to be met and for the app to run effectively
Chapter 3 explores the technical details of Audentify's architecture and design, such as addressing the various modules/components that make up the application and how they interact with one another. Design diagrams are also included to offer a more detailed view of the application's design

Chapter 2

Software Requirements Specification

2.1 Introduction

Figure 2.1: Audentify Logo

Streamlining music exploration for enthusiasts, Audentify is a web application designed to simplify the process of identifying song genres and discovering similar music. By utilizing machine learning, Audentify enables users to identify the genre of any song as well as music reccomendations to explore the genre deeper.

2.2 Overall Description

Audentify helps users explore music by identifying song genres and recommending

similar music. This application uses a database to store pre chosen songs for each genre and achieves its function by allowing users to upload audio files or provide real-time audio input. It then analyzes the audio to extract relevant features. Finally, it uses a pre-trained machine learning model to classify the genre and recommend similar songs based on the identified genre.

2.3 External Interface Requirements

1. User interface: The homepage of the application is designed with simplicity and user-friendliness in mind, featuring a clean layout with a prominent search bar at the center. Users are provided with clear options to either search for specific music or explore curated recommendations tailored to their preferences. Within the user profile section, users can conveniently manage their account settings, view their listening history, and access saved songs or playlists. This section enhances user engagement and customization, offering a personalized experience within the application. The recommendations interface serves as a platform to present personalized recommendations based on identified music or user preferences. Utilizing visually engaging layouts such as carousels or grids, users can easily discover recommended songs or playlists.

   Standard buttons and functions, including play buttons and options to save or share recommendations, ensure straightforward interaction and accessibility for users.

2. Hardware Interface: The application relies on two primary hardware components for its operation. Firstly, a web server is crucial for deploying the application, managing user requests, and delivering the web interface. users will access the application through their personal devices, including laptops, desktops, smartphones, and tablets, all equipped with standard web browsers. It is essential for the application to be compatible with a wide range of commonly used devices and operating systems to ensure accessibility and user convenience.

3. Software Interface:

   (a) Databases:

The application will utilize the external music database Gracenote (Echonest is an alternative) through an API to utilize music metadata to aid in suggesting recommendations to the user.

(b) Operating System:

Involvement of the OS in the application's utility is minimal. At most, the application will need file system access to retrieve the audio files for genre classification. Additionally, the application utilizes the OS's network functionalities to fetch music metadata.

(c) Libraries:

The core libraries used in the development of the application are:

i. Librosa: Contains the relevant tools needed for music and audio analysis. It is responsible for preprocessing the audio data and presenting it to the classification model in an appropriate form.

ii. TensorFlow: A deep learning framework that will be used to build, train, and deploy the CNN model for genre identification.

iii. Flask: The framework used to implement the web application's backend logic and to handle user interactions.

2.4 System Features

1. Audio input: This feature allows users to provide audio input for genre identification and recommendation. Users can either upload an audio file (MP3 file) or use real time audio.In terms of functionality, the system should be able to accept audio files in common formats like MP3 or WAV, with a maximum size of 15 MB. It should also display an error message if the file format is unsupported or if the file size exceeds the limit. When accessing the microphone, the system must use secure methods and ask for explicit user permission. Additionally, it should handle any potential errors related to microphone access, such as device unavailability, and notify the

user accordingly. Moreover, if the system doesn't receive sufficient information for an accurate genre prediction from real-time audio input, it should prompt the user to provide input again.

2. Genre Identification: This feature utilizes Machine Learning to analyze audio features and classify the genre of the uploaded/captured music. System converts the audio input into a spectrogram, a visual representation of frequencies over time. The patterns within the spectrogram (e.g., distribution of energy across frequencies) are analyzed by the system. System identifies the most likely genre based on the patterns observed in the spectrogram. In terms of functionality, the system should utilize a Machine Learning model trained on a diverse dataset of labeled spectrograms, encompassing various music genres. It should then analyze patterns within the spectrogram that are pertinent to genre classification, such as frequency distribution and rhythmic patterns. Additionally, the system must provide the identified genre along with a confidence score to indicate the accuracy of the prediction. Moreover, it should handle potential errors that may occur during audio processing, such as corrupted files, and inform the user accordingly.

3. Song/Artist Recommendation. This feature recommends similar artists and songs based on the identified genre. In terms of functionality, the system will maintain an extensive music database encompassing artist, song, and genre information. It will provide recommendations for artists and songs sharing strong genre similarities with the identified genre. Optionally, the system may also consider incorporating user preferences, like moods, for more personalized recommendations. Recommendations will be displayed in a user-friendly format, presenting artist names, song titles, and optionally album artwork for enhanced usability.

4. Genre Exploration with Custom Playlists: This feature allows users to explore genres by creating custom playlists based on recommendations and search functionality. In terms of functional requirements, the system will offer users functionalities to browse and select genres from identified recommendations or search results. Upon selection, it will display a list of artists and songs belonging to the chosen genre. Users will have the ability to add recommended songs or search for additional songs within the same genre. Furthermore, the system will facilitate the creation and

management of user playlists, allowing song addition, removal, and adjustment of the order. Additionally, it will utilize user-created playlists for future access and editing purposes.

2.5 Other Nonfunctional Requirements

1. Performance Requirements

   (a) Fast Response Times:

      i. Uploading audio files and receiving genre identification feedback should occur within a reasonable timeframe

      ii. Real-time genre identification through microphone access should have minimal latency to avoid a disjointed user experience

      iii. Generating recommendations based on identified genres should be quick to maintain user engagement

      be quick to maintain user engagement

   (b) Scalability:

      The system should be able to handle an increasing number of users and requests without significant performance degradation.

2. Safety Requirements

   (a) Error Handling and User Feedback:

      i. The system should appropriately handle potential errors during audio processing (e.g., corrupted files) and provide clear error messages to users.

      ii. The application should not generate misleading or incorrect genre identifications.

iii. Expand error handling to include potential issues duringthe recommendation. For example, if the identified genre has limited recommendations in the database, the system should inform the user and potentially offer suggestions for broadening the search criteria.

3. Security Requirements

   (a) Data Security:

      i. User-uploaded audio files and any associated data(e.g.,genre identification results) should be stored securely following appropriate practices.

      ii. Implement appropriate access controls to prevent unauthorized data access.

   (b) User Authentication:

      A secure user authentication mechanism should be implemented.

4. Software Quality Attributes

   (a) Usability:

      i. The web application interface should be user-friendly,catering to users with varying levels of technical expertise.

      ii. Provide clear instructions and informative feedback throughout the user interaction process.

   (b) Reliability: The genre identification and recommendation functionalities should be reliable and produce consistent results for the same audio input.

   (c) Maintainability:

      The code should adhere to well-established coding practices to promote maintainability. This includes utilizing clear and consistent coding styles, proper code documentation, and a modular architecture.

Chapter 3

System Architecture and Design

3.1 System Overview

This project aims to develop a web application that classifies music genres based on uploaded audio files and recommends similar songs based on the predicted category.

- User Interface (UI): Provides a user-friendly interface for uploading audio files, displaying results, and exploring recommendations.
- Music Genre Classification: Leverages a pre-trained Convolutional Neural Network (CNN) model to predict the most likely music genre for the uploaded audio. The model assigns a confidence score to indicate its prediction certainty.
- Song Recommendation Database: Stores pre-defined song recommendations associated with various music genres. Each recommendation might include details like title, artist, album (optional), and potentially, links to play the songs.
- Backend Processing: Handles server-side logic, including file upload validation, feature extraction, model interaction, database interaction to retrieve recommendations, and response preparation for the UI.

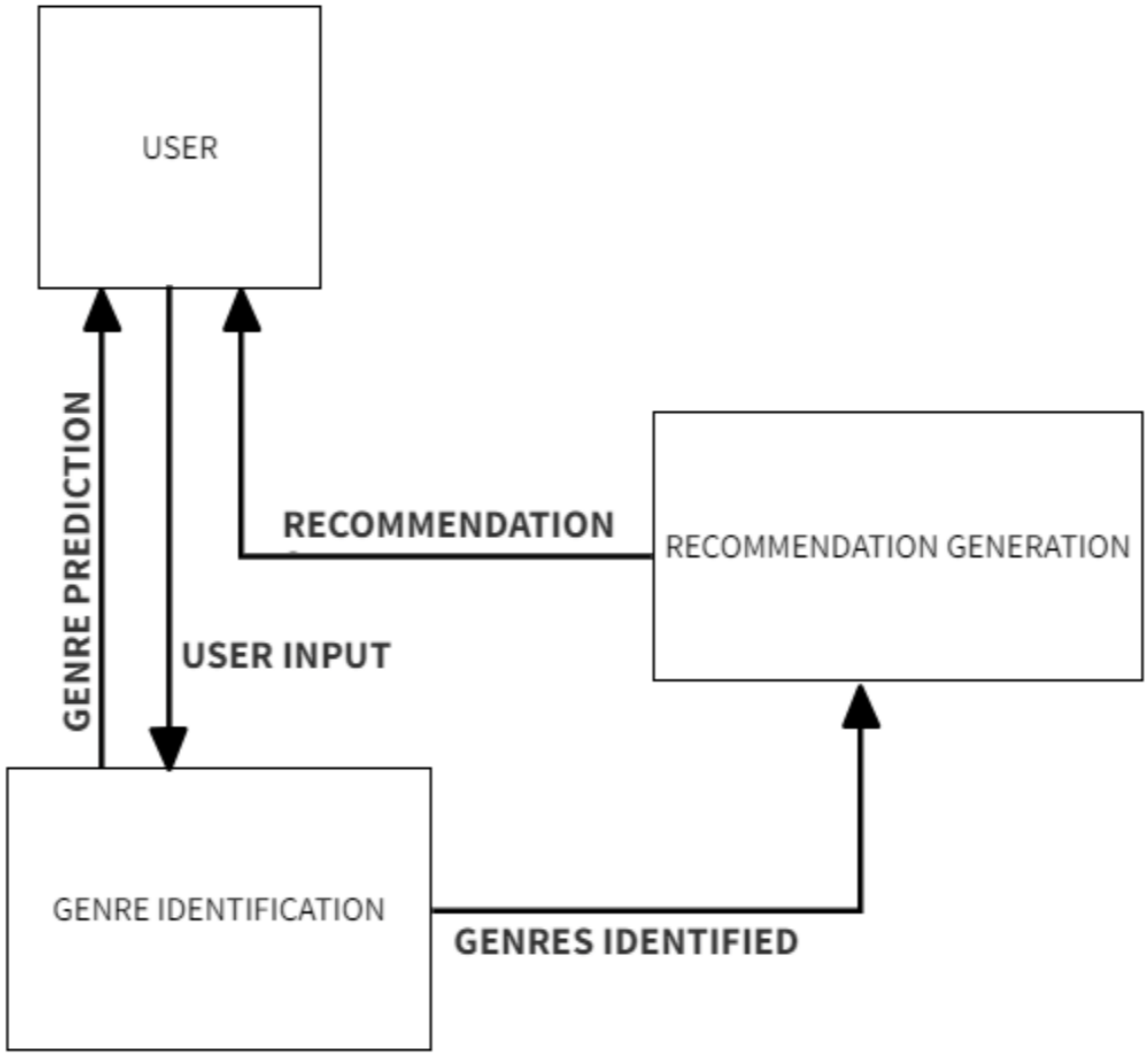

Figure 3.1: Architecture Diagram

The architecture diagram showcases the system's components and how they interact. The user submits a song, the genre prediction model classifies it, and the recommendation module uses this classification to offer tailored suggestions.§

3.2 Architectural Design

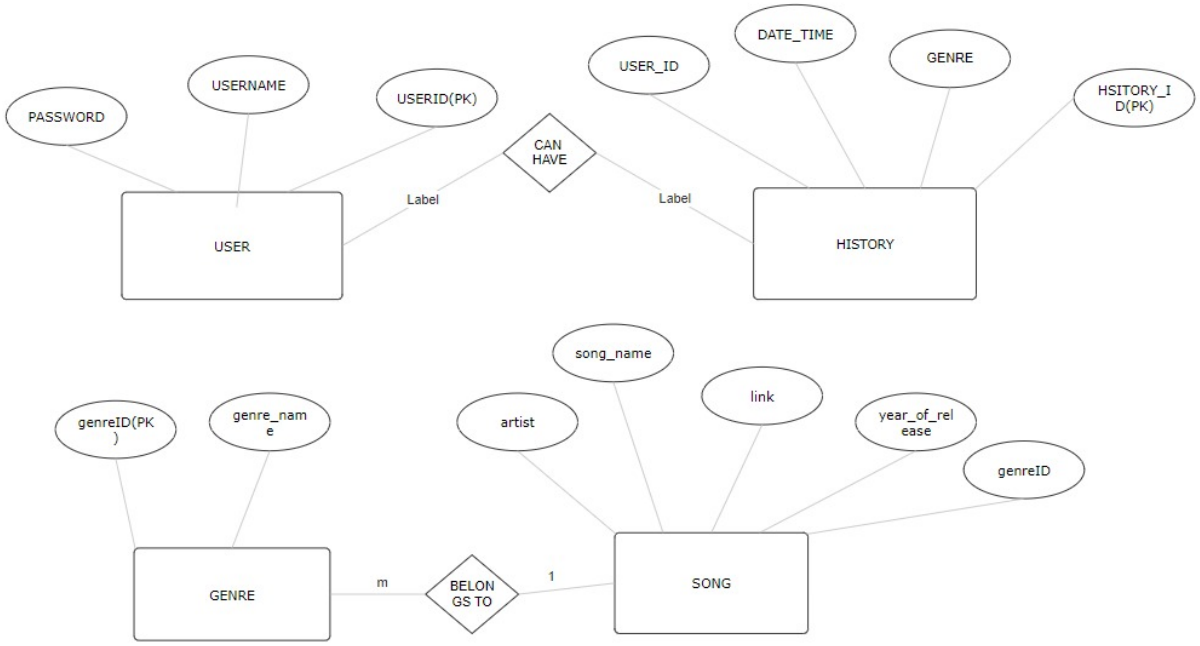

Figure 3.2: ER Diagram

The ER diagram consists of a User database designed for authentication, storing usernames and login credentials. It tracks user behavior with a History table, recording each search query alongside the time it was made. To categorize music recommendations, a Genre table lists available music genres. Lastly, a Songs table contains pre-defined recommendations, each linked to a specific genre.

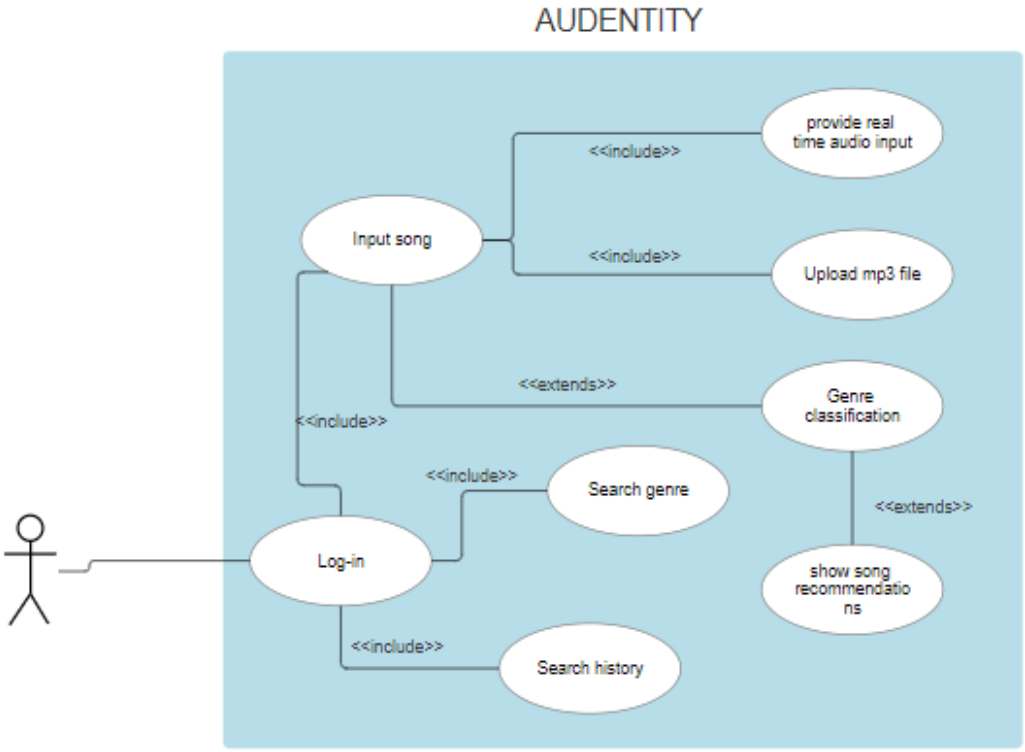

Figure 3.3: Use Case diagram

This Use Case diagram demonstrates a user's journey within the application. After logging in, a user can input audio for analysis, view their search history, search directly for songs by genre, and receive song recommendations based on the system's genre classification.

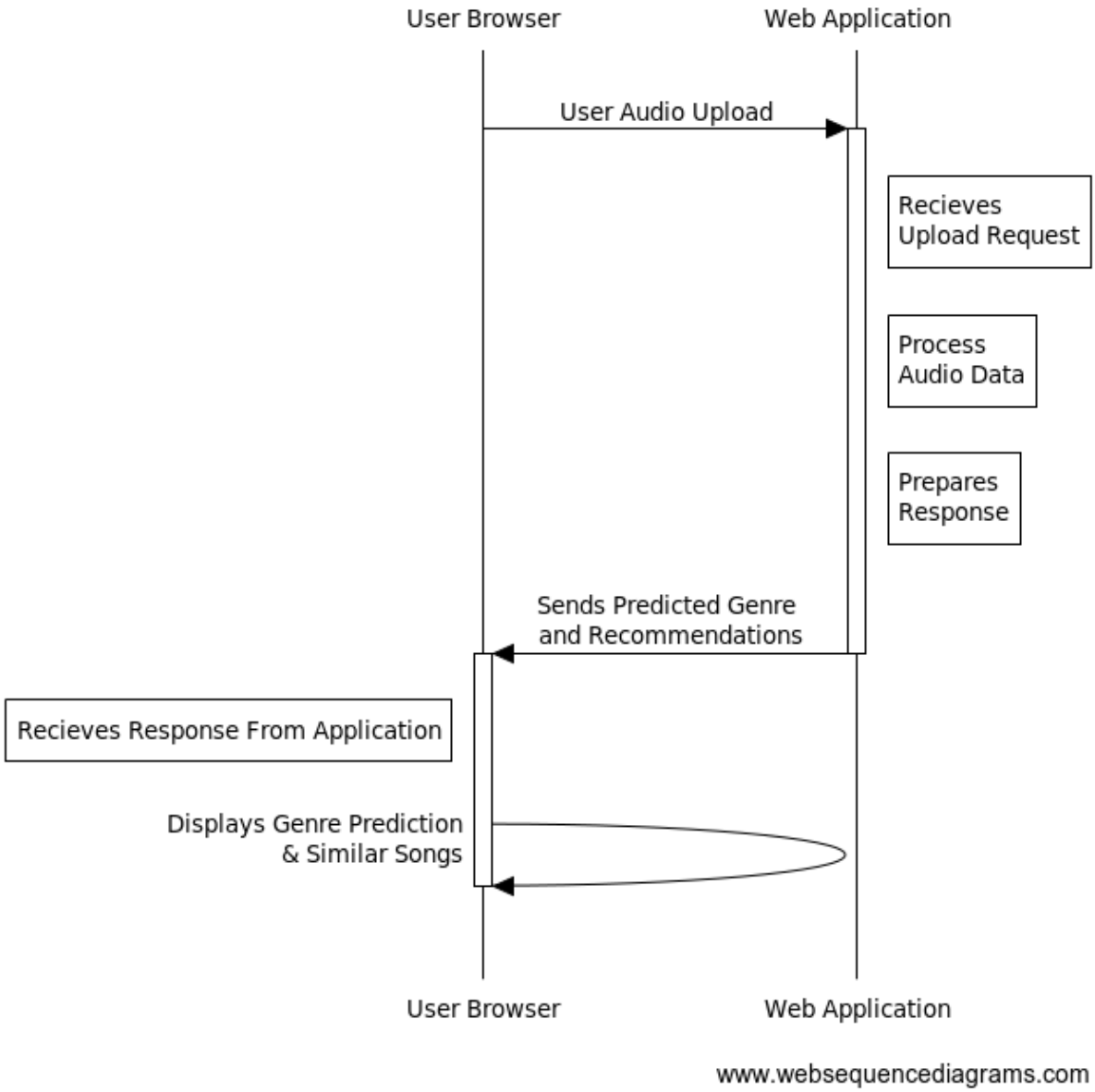

Figure 3.4: Sequence Diagram

This diagram focuses on the interactions that occur when a user submits a song. The user's web browser sends the song to the web application. The web application then analyses the song's characteristics and determines its genre. Finally, the predicted genre is sent back to the user's browser for viewing.

3.3 Dataset identified

GTZAN dataset

- Content: The GTZAN dataset consists of audio recordings categorized into 10 different musical genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock.
- Format: Each audio recording is provided in a WAV format with a sampling rate of 22.05 kHz and a mono channel.
- Size: The dataset contains a total of 1,000 audio recordings, with 100 recordings for each genre.
- Availability: The GTZAN dataset is freely available for download from various online repositories - https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

FMA Dataset

- Content: The FMA dataset consists of a diverse collection of music recordings spanning various genres, artists, and recording conditions. Each recording is accompanied by rich metadata, including genre labels.
- Format: Audio recordings are typically provided in compressed formats like MP3 or FLAC. The metadata is available in structured formats like JSON or CSV.
- Size: The FMA dataset offers a massive collection, with millions of recordings across hundreds of genres.
- Availability: The FMA dataset is freely accessible for download at Kaggle - https://www.kaggle.com/datasets/imsparsh/fma-free-music-archive-small-medium

3.4 Proposed Methodology/Algorithms

Preprocessing with Librosa

The application utilizes Librosa, a Python library for audio analysis, to preprocess the input audio data. The algorithms used are:

- Signal Conversion: The raw audio data is converted into a NumPy array using Librosa functions like librosa.load. The algorithms involved are:

- Segmentation: The audio is divided into short clips of fixed duration (e.g., 5 seconds) using techniques like windowing and striding.

- Feature Extraction: Informative features are extracted from each audio clip suitable for genre classification. Common features employed include:

Mel-frequency cepstral coefficients (MFCCs):

A Short-Term Fourier Transform (STFT) is first applied to transform the audio signal into a spectrogram. Mel filters are then used to warp the frequency scale according to human auditory perception. Finally, a Discrete Cosine Transform (DCT) is applied to extract the MFCCs, capturing the spectral envelope of the audio. The librosa method - (librosa.feature.mfcc) is used for efficient MFCC calculation.

Classification with Convolutional Neural Network (CNN) and Support Vector Machine (SVM)

- Input Layer: The model accepts a 3D tensor representing the extracted features (e.g., time steps, frequency bins, feature channels).

- Convolutional Layers: These layers apply convolutional filters to the input feature map (e.g., MFCCs). Each filter learns to detect specific patterns or features within the data. The activation function used (ReLu) introduces non-linearity.

- Convolution: The filter slides across the input, performing element-wise multiplication and summation with the input data. This operation extracts local features within the data.

- Pooling Layers: These layers downsample the feature maps by applying a pooling operation (Max Pooling is used here). This reduces dimensionality and computational cost while preserving essential information.

- Flatten Layer: This layer transforms the pooled feature maps into a 1D vector, preparing the data for the fully connected layers.

- Output Layers: Here, the SVM is integrated into the model architecture to perform the task of genre classification, the inputs of the model are the feature embeddings provided as output from the CNN (VGGish) based on which the SVM makes predictions

Training Process

The model will be trained on a labeled dataset containing audio clips with corresponding genre labels in the following way:

Data Preparation:

- Preprocess the audio data in the dataset using Librosa (segmentation, feature extraction).

- Split the data into training, validation, and (optional) test sets.

Model Training:

- A pre-trained VGGish model is used to extract audio features from the input music files. These features capture essential characteristics of the music relevant for genre classification.

- The extracted VGGish features are used as input for the model. The corresponding genre labels serve as the target output.

- A Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is employed for classification. The RBF kernel allows the SVM to learn complex non-linear relationships between the audio features and genre labels.

- Use the validation set to monitor the model's performance and prevent overfitting.

Evaluation:

- Evaluate the model's performance on the held-out test set (if available) to assess its generalization capabilities on unseen data.
- Metrics like accuracy, precision, recall, and F1-score can be used for evaluation.

3.5 User Interface Design

For the user interface we use HTML , CSS and JAVASCRIPT. HTML is the basic building blocks of the website, CSS (Cascading style sheets) is basically used to stylize the webpages and make it visually appealing and Javascript is also utilized to make the website more dynamic and improves user interaction with the web application

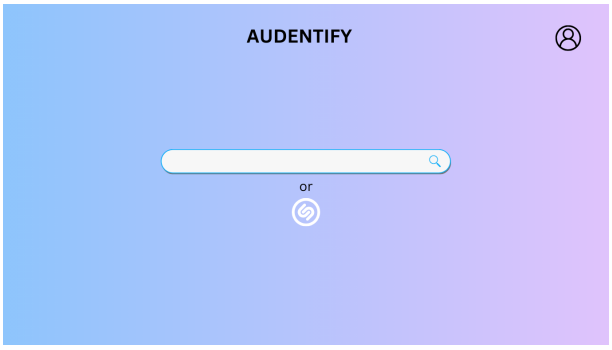

Figure 3.5: Main Page of Audentify

3.6 Database Design

SQL is the database used in the application for storing metadata as it excels at storing and managing well-defined, structured data, which in this case would be:

- Genre Information: The Genres table can efficiently store genre names and IDs, allowing for easy retrieval and management.
- Pre-defined Recommendations : Since a fixed set of recommendations is offered per genre, an SQL table can efficiently store song titles, artist names, and links.
- User Information : For user accounts, an SQL database can handle user credentials, profile information, and potentially basic preferences.

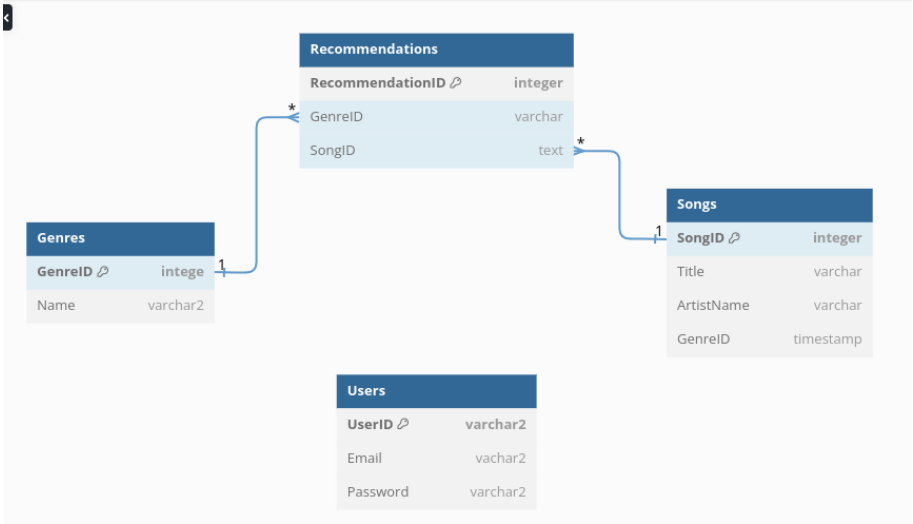

Figure 3.6: Audentify Database Schema

3.7 Description of Implementation Strategies

(1) Audio Capture

- **Python Library:** *Librosa, Json, NumPy* (https://librosa.org/doc/https://librosa.org/doc/)
- **Associated Methods:**

1. librosa.load(file_path, sr=SAMPLE_RATE): This function from librosa loads an audio file at the specified path with a desired sampling rate.
2. librosa.feature.mfcc(y=signal[start:finish], sr=sample_rate, n_mfcc=num_mfcc, n_fft=n_fft, hop_length=hop_length): This function extracts Mel-frequency cepstral coefficients (MFCCs) from a specific audio segment of the loaded audio data.
3. json.load(fp): This method is used to load the data stored in JSON format from the data.json file.
4. np.array(data["mfcc"]) and np.array(data["labels"]): These lines convert the lists containing MFCC features and genre labels from the loaded JSON data into NumPy arrays for easier manipulation.

(2) CNN Design

- **Library:** Tensorflow.keras
- **Network Architecture:**
  - keras.Sequential(): This function is used to define a sequential convolutional neural network model.
  - keras.layers.Flatten(): This layer flattens the input data from a 3D tensor to a 1D vector suitable for feeding into dense layers.
  - keras.layers.Dense(): This function creates fully-connected (dense) layers in the CNN architecture. Three hidden layers are used with 512, 256, and 64 neurons respectively, all using the ReLU activation function and L2 regularization to prevent overfitting. Dropout layers with a rate of 0.3 are added after each hidden layer to further reduce overfitting.
  - keras.layers.Dense(10, activation="softmax"): The output layer has 10 neurons (one for each genre class) and uses the softmax activation function to generate probabilities for each class.

(3) Evaluation Methods

- model.compile()**:** This method configures the training process by specifying the optimizer (Adam with a learning rate of 0.001), loss function (sparse categorical crossentropy), and evaluation metrics (accuracy).
- model.fit()**:** This method trains the model on the provided training data (inputs_train and targets_train) with validation data (inputs_test and targets_test) for monitoring performance during training. We use a batch size of 32 and train for 50 epochs.

3.8 Module Division

This application can be divided into several modules for better organization and maintainability.

1. Data Preprocessing Module

(a) Input Audio Processing:

i. Reads the input audio file.

ii. Extracts audio features like spectrograms.

(b) Dataset Management:

i. Loads the music dataset with genre labels.

ii. Preprocesses the audio files in the dataset (similar to input processing).

2. Machine Learning Model Module

(a) Trains a machine learning model on the preprocessed dataset. (e.g., Support Vector Machine (SVM), Convolutional Neural Network (CNN) for image classification of spectrograms).

(b) Saves the trained model for future use.

3. Genre Identification Module

(a) Loads the trained model.

(b) Feeds the extracted features from the input audio to the model.

(c) Obtains the predicted genre for the input audio.

4. Recommendation Module

(a) Queries the dataset for songs belonging to the predicted genre.

(b) Calculates similarity scores between the input spectrogram and spectrograms of

songs in the dataset.

(c) Ranks songs based on similarity scores.

(d) Recommends songs with the highest similarity scores.

5. User Interface Module

   (a) Provides an interface for users to interact with the application.

   (b) Allows users to upload audio files.

   (c) Displays the identified genre and recommended songs.

3.9 Work Schedule - Gantt Chart

Figure 3.7: Gantt Chart

Work is divided into 4 phases of which are:

- Planning and Design
- Backend Development and Integration
- Testing and Integration
- Deployment and Maintenance

Chapter 4

Results and Discussions

4.1 Overview

The web application effectively categorizes music into 6 out of 10 trained genres. File uploads achieved approximately 80% accuracy, while real-time audio classification was slightly less accurate at around 70%. The application excels at distinguishing distinct genres but struggles with closely related ones, likely due to the limited size of the training dataset.

4.2 Testing

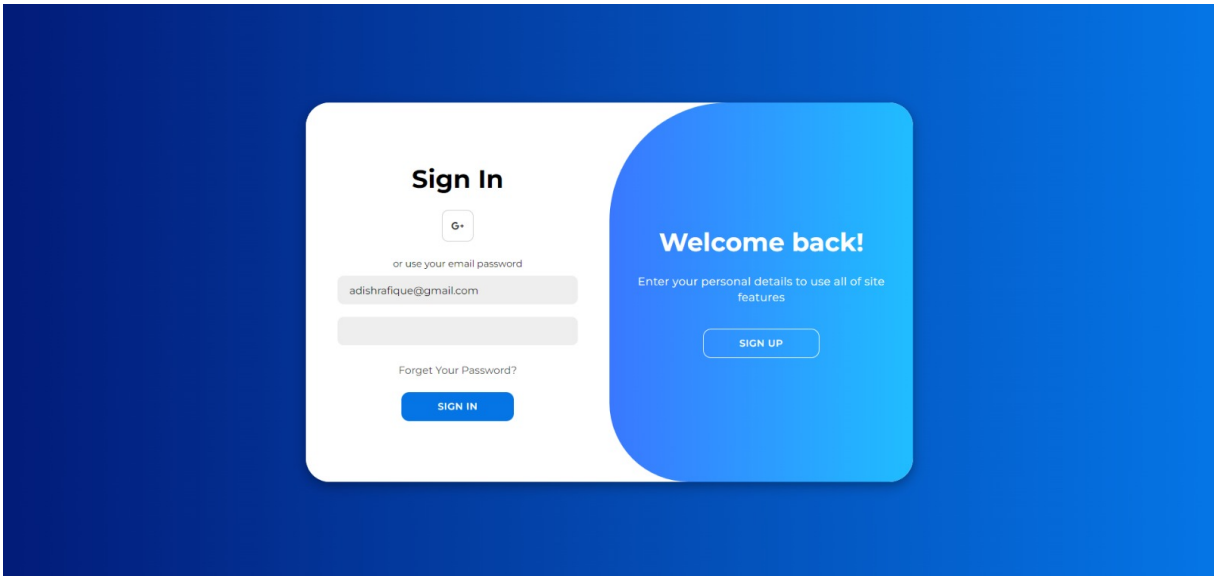

Figure 4.1: Login Page

Figure 4.1 displays the login page which authenticates each of the users attempting to use the web application, users also have the option to create an account if they don't already have one

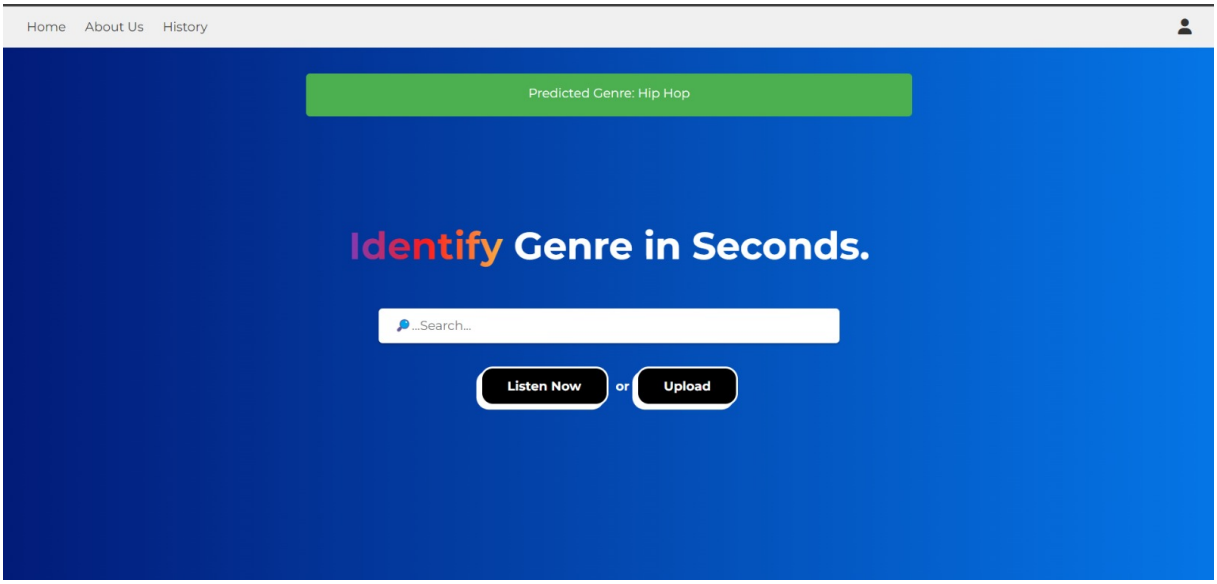

Figure 4.2: Main Page w/ Prediction Output

Figure 4.2 demonstrates the main page of Audentify where users can upload a song either via real time input or as a file upload, the predicted genre is then displayed along with music recommendations. The search bar allows users to seach up songs of a particular genre

Figure 4.3: History Page

Figure 4.3 shows a history page where users can keep track of all previous queries made on the website

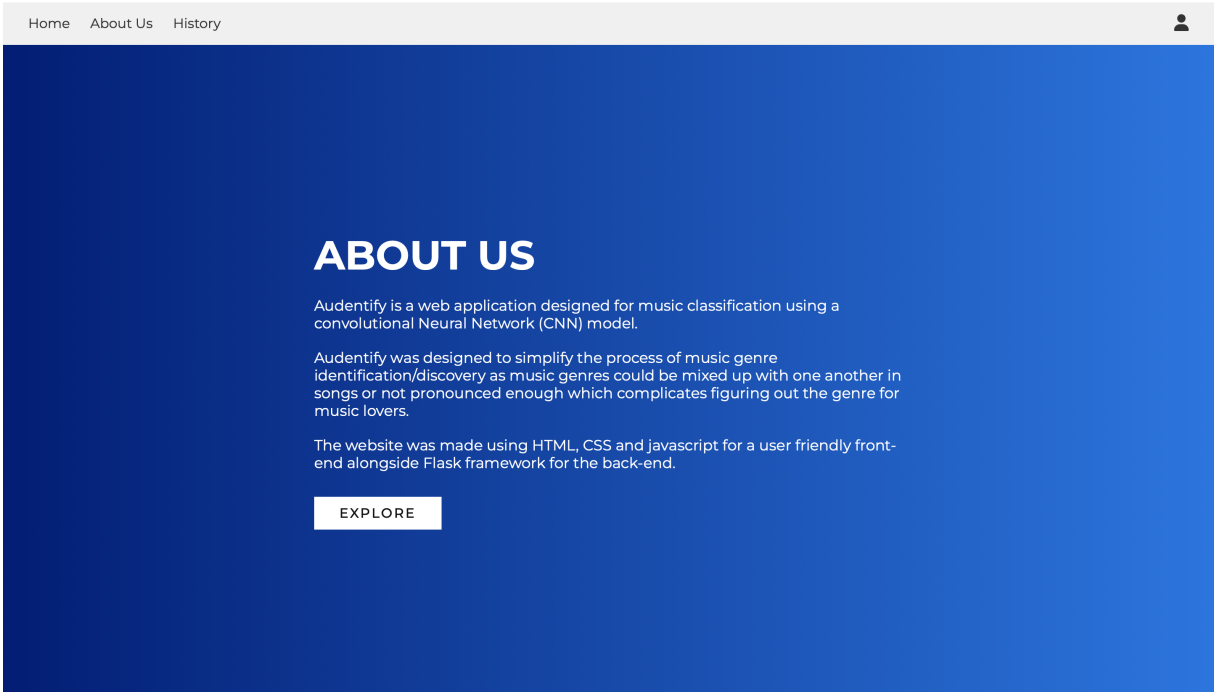

Figure 4.4: About Us Page

Figure 4.4 explains briefly the background behind Audentify and provides an overview of its implementation

4.3 Quantitative Results

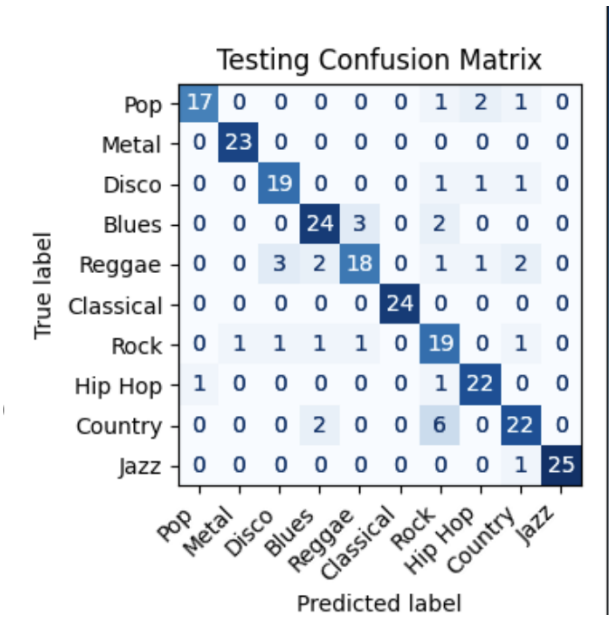

Figure 4.5: Testing Confusion Matrix

The Confusion Matrix is found to be most effective at predicting metal and classical songs with the least error. Rock, Reggae and Hip-Hop predictions have been frequently accurate but with more inaccurate predictions than those of metal and classical

4.4 Graphical Analysis

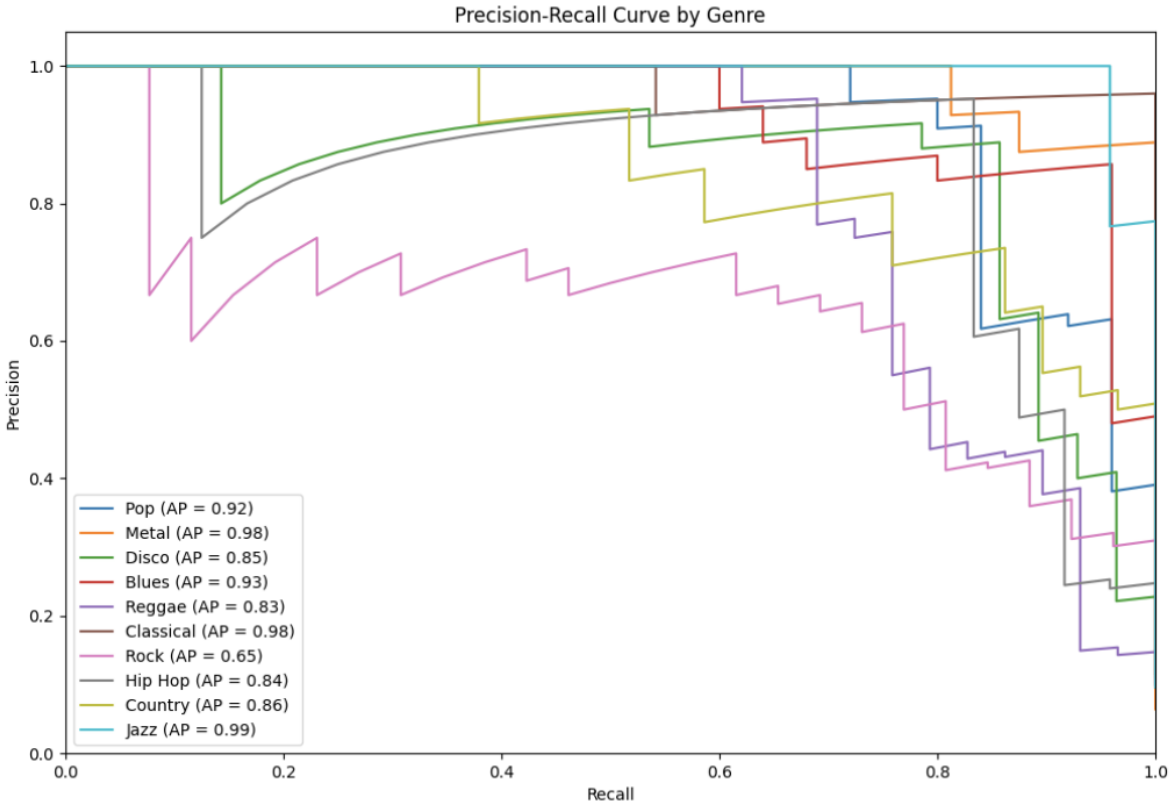

Figure 4.6: Precision Recall Curve

According to the graph:

- **High-performing genres (Jazz, Metal, Classical, Blues):** These genres have high AP scores (0.99, 0.98, 0.93, and 0.98 respectively), indicating that the model is excellent at predicting these genres with minimal false positives or negatives.
- **Moderate performers (Pop, Disco, Reggae, Country, Hip Hop):** These genres have AP scores ranging from 0.80 to 0.90. The model performs well on these genres, balancing precision and recall effectively.
- **Low performer (Rock):** Rock has the lowest AP (0.65), suggesting the model

has the most difficulty predicting this genre, possibly due to its diverse subgenres or overlapping features with other genres.

4.5 Discussion

Therefore, real-time audio and file-upload genre classification proved effective for a subset of genres, particularly Metal and Classical, demonstrating high accuracy. The model performed reasonably well for Hip-Hop, Rock, Reggae, and Pop, but with some room for improvement. However, it struggled to accurately identify Blues, Jazz, Country, and Disco. This difficulty likely stems from the limited size of the training dataset, which may not have captured the specific features that distinguish these genres.

Chapter 5

Conclusion

5.1 Conclusion

In conclusion, Audentify was successfully implemented to perform the task of music genre classification with the aid of Machine Learning Models and a clean and interactive user interface to aid users in diversifying their taste in music in a simple manner.

5.2 Future Scope

Audentify could be improved upon with a larger training dataset to allow it to classify more genres of music than the existing ones. Personalised recommendations feature can also be enhanced by utilising API's from external music databases to offer a wider range of recommendations.

Bibliography

[1] H. Garg and M. Dave, "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/IoT-SIU.2019.8777334.

[2] M. Ebrahimi, Y. Chai, H. H. Zhang and H. Chen, "Heterogeneous Domain Adaptation With Adversarial Neural Representation Learning: Experiments on E-Commerce and Cybersecurity," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 2, pp. 1862-1875, 1 Feb. 2023, doi: 10.1109/TPAMI.2022.3163338.

[3] Music Genre Classification with Deep Learning: A Review (2019), X. Wu et al. Publication: In Proceedings of the IEEE, vol. 108, no. 2, pp. 201-222, Feb. 2019.

[4] Music Recommendation Systems (2018): By F. Ricci et al. Title: Recommender Systems Handbook (Second Edition) Editors: F. Ricci, L. Rokach, B. Shapira, and P. Adomavicius Publisher: Springer, Cham Pages: 737-775

[5] Content-Based Music Recommendation Systems: State of the Art and Future Directions (2017):Bogdanov et al. Publication: ACM Computing Surveys (CSUR), vol. 51, no. 2, pp. 1-62, Feb. 2018

[6] A Survey on Music Emotion Recognition: Systems, Measurements, and Applications (2011):Y. Hu et al. Publication: ACM Computing Surveys (CSUR), vol. 43, no. 2, pp. 1-29, Feb. 2011.

[7] Music Information Retrieval (2007):J. Stephen Downie Book title: Springer Information Science and Engineering Publisher: Springer

[8] Convolutional Neural Networks for Music Classification (2016):K. H. Kwon et al. Publication: In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6438-6442, March 2016.

[9] A Survey on Music Recommendation Systems for Streaming Services (2022): B. Yu et al. Publication: ACM Computing Surveys (CSUR), vol. 55, no. 2, pp. 1-38, Feb. 2022.

[10] Deep Learning for Music Recommendation (2018): X. Liang et al. Publication: In Proceedings of the 2018 ACM on Conference on Information and Knowledge Management, pp. 1919-1928, Oct. 2018.

Appendix A: Presentation

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039
(Affiliated to APJ Abdul Kalam Technological University)

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes

After the completion of the course the student will be able to:

CO1:

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

CO2:
Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

CO3:
Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

CO4:
Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

CO5:
Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Appendix C: CO-PO-PSO Mapping

SL. NO	DESCRIPTION	Blooms' Taxonomy Level
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Apply (Level 3)
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Apply (Level 3)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Apply (Level 3)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Apply (Level 3)
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Apply (Level 3)

Table 5.1: Course Outcomes

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

Figure 5.1: CO-PO AND CO-PSO MAPPING

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.1-PO1	HIGH	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6-22T.1-PO2	HIGH	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
101003/CS6-22T.1-PO3	HIGH	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
101003/CS6-22T.1-PO4	HIGH	Identify technically and economically feasible problems by analysis and interpretation of data.
101003/CS6-22T.1-PO6	MEDIUM	Responsibilities relevant to the professional engineering practice by identifying the problem.
101003/CS6-22T.1-PO7	MEDIUM	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
101003/CS6-22T.1-PO8	HIGH	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
101003/CS6-22T.1-PO9	MEDIUM	Identify technically and economically feasible problems by working as a team.
101003/CS6-22T.1-PO10	MEDIUM	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
101003/CS6-22T.1-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
101003/CS6-22T.1-PO12	HIGH	Identify technically and economically feasible problems for long term learning.
101003/CS6-22T.1-PO1	MEDIUM	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
101003/CS6-22T.1-PO2	MEDIUM	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
101003/CS6-22T.1-PO3	MEDIUM	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
101003/CS6-22T.2-PO1	HIGH	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.2-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
101003/CS6-22T.2-PO3	HIGH	Design solutions for complex engineering problems and design based on the relevant literature.
101003/CS6-22T.2-PO4	HIGH	Use research-based knowledge including design of experiments based on relevant literature.
101003/CS6-22T.2-PO5	HIGH	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
101003/CS6-22T.2-PO6	MEDIUM	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
101003/CS6-22T.2-PO8	HIGH	Apply ethical principles and commit to professional ethics based on the relevant literature.
101003/CS6-22T.2-PO9	MEDIUM	Identify and survey the relevant literature as a team.
101003/CS6-22T.2-PO10	HIGH	Identify and survey the relevant literature for a good communication to the engineering fraternity.
101003/CS6-22T.2-PO11	MEDIUM	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
101003/CS6-22T.2-PO12	HIGH	Identify and survey the relevant literature for independent and lifelong learning.
101003/CS6-22T.2-PSO1	MEDIUM	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
101003/CS6-22T.2-PSO2	MEDIUM	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
101003/CS6-22T.2-PSO3	MEDIUM	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
101003/CS6-22T.3-PO1	HIGH	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
101003/CS6-22T.3-PO2	HIGH	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.3-PO3	HIGH	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
101003/CS6-22T.3-PO4	HIGH	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6-22T.3-PO5	HIGH	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
101003/CS6-22T.3-PO6	MEDIUM	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
101003/CS6-22T.3-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6-22T.3-PO8	HIGH	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
101003/CS6-22T.3-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6-22T.3-PO10	MEDIUM	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
101003/CS6-22T.3-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
101003/CS6-22T.3-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6-22T.3-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
101003/CS6-22T.4-PO1	MEDIUM	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6-22T.4-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.4-PO3	MEDIUM	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
101003/CS6-22T.4-PO4	MEDIUM	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
101003/CS6-22T.4-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
101003/CS6-22T.4-PO8	HIGH	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS6-22T.4-PO9	HIGH	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6-22T.4-PO10	HIGH	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
101003/CS6-22T.4-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
101003/CS6-22T.4-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
101003/CS6-22T.4-PSO1	MEDIUM	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
101003/CS6-22T.4-PSO2	MEDIUM	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
101003/CS6-22T.4-PSO3	MEDIUM	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
101003/CS6-22T.5-PO1	HIGH	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6-22T.5-PO2	HIGH	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.5-PO3	HIGH	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
101003/CS6-22T.5-PO4	MEDIUM	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6-22T.5-PO5	MEDIUM	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO6	MEDIUM	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO7	MEDIUM	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO8	HIGH	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO9	MEDIUM	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO11	MEDIUM	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PO12	HIGH	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PSO1	MEDIUM	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

MAPPING	LOW/MEDIUM/HIGH	JUSTIFICATION
101003/CS6-22T.5-PSO2	MEDIUM	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
101003/CS6-22T.5-PSO3	MEDIUM	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.