RAJIV GANDHI INSTITUTE OF TECHNOLOGY

GOVERNMENT ENGINEERING COLLEGE

KOTTAYAM-686 501

DEPARTMENT OF COMPUTER APPLICATIONS

20MCA2045 - MINI PROJECT REPORT

DECEMBER 2023

MUSIC RECOMMENDATION SYSTEM

Submitted By

ZAHEER

(KTE22MCA-2057)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRUVANANTHAPURAM

ACKNOWLEDGMENT

I want to express my gratitude to everyone who has supported me throughout the endeavour. First and foremost, I give thanks to God Almighty for His mercy and blessings, for without His unexpected direction, this would still be only a dream.

I sincerely thank **Dr. PRINCE A**, Principal of the Rajiv Gandhi Institute of Technology, Kottayam, for providing the environment in which this research could be completed.

I owe a huge debt of gratitude to **Dr. Reena Murali** Professor and Head of the department and **Dr. Vineetha S** Associate Professor Head of the department in charge of Computer Applications, for granting permission and making available all of the facilities needed to complete the project properly.

I am grateful to **Prof. Sajithamol TS**, Assistant Professor, the Department of Computer Applications, for her helpful criticism on my thesis.

I also express my sincere thanks to the Project co-ordinator **Dr. Sangeetha Jose**, Associate Professor, Department of Computer Applications for the constructive suggestions and inspirations throughout the project.

Finally, I'd like to take this chance to express my gratitude to the Department of Computer Applications' entire teaching and technical team.

ZAHEER

ABSTRACT

As streaming platforms have become more and more popular in recent years and music consumption has increased, music recommendation has become an increasingly relevant issue.Music applications are attempting to improve their recommendation systems in order to offer their users the best possible listening experience and keep them on their platform.

In this proposed system,l develop a prototype in recommendation of dynamic music recommendation system based on human emotions. Based on each human listening pattern, the songs for each emotions are trained using machine learning techniques. Based on the contents in input,respective songs for the specific mood would be played to hold the users.In this approach, the application gets connected with human feelings thus giving a personal touch to the users.Therefore my projected system concentrate on identifying the human feelings for developing emotion based music player using machine learning techniques.

Keywords: Music recommendation Content-based Collaborative approach Hybrid approach

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Definition
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbors
VScode	Visual Studio Code
IDE	Integrated Development Enviornment

Chapter 1

INTRODUCTION

1.1 Need for the Project

This project, titled **"Music recommendation system"** is pivotal for both users and music platforms due to their ability to revolutionize the way people discover and engage with music. Firstly, these systems cater to the ever-growing need for personalization in the digital music landscape. By analyzing user preferences, listening habits, and contextual cues, recommendation algorithms curate a tailored selection of music that resonates with individual tastes. This personalized approach enhances user satisfaction, fostering deeper connections between listeners and the vast pool of available music.

The system address the challenge of content overload and aid in content discovery. With an exponentially expanding catalog of songs and artists, users often face difficulty in navigating and finding music that suits their preferences. Recommendation systems streamline this process by surfacing relevant and diverse content, introducing users to new tracks, genres, or artists they might not have encountered otherwise. This not only enriches the user experience but also supports lesser-known musicians by offering them visibility to a wider audience.

Lastly, music recommendation systems hold immense strategic value for music platforms. They play a pivotal role in user retention and engagement, crucial metrics for any digital service. By keeping users engaged through personalized recommendations, platforms can prolong user sessions, increase interactions, and foster a loyal user base. Moreover, the insights garnered from user data aid in refining the platform's content strategy, optimizing recommendations, and shaping marketing initiatives, thereby contributing to long-term growth and competitiveness in the market

1.2 Objective

The goal of this project was to learn about machine learning and its fundamental concepts, as well as numerous data mining approaches and algorithms. Another goal was to become familiar with a variety of machine learning algorithms and how to use them. Learning algorithms alone does not make you an engineer; the true challenge is determining which method is best for a certain project. In terms of results, the major goal was to establish a framework for consumers to use in order to assist them find the ideal tunes for them. This project seeks to discover the correlation and similarity between different songs, and then construct a recommendation system framework that suggests new music for your Spotify playlist based on that information. The primary objectives of this project to provide music to users based on their individual tastes and preferences ,keep users engaged with the music platform by offering a variety of content and playlists and support the business model, such as increasing sub- scription rates, ad revenue, or music sales.

.

1.3 Scope of the project

This recommendation system focuses on analyzing the inherent features and attributes of items, such as music tracks, articles, or products, to make personalized suggestions. Its scope involves the extraction and evaluation of item-specific characteristics, such as metadata, text, audio features, or visual content. Using this information, the system creates profiles or representations of items and matches them with user preferences or historical interactions. The scope extends to developing algorithms that compute similarities between items based on their content, facilitating accurate and relevant recommendations, while also encompassing ongoing improvement through user feedback and adapting to evolving content characteristics.

Chapter 2

LITERATURE SURVAY

2.1 Literature survey

Thi Thuan To and Sutheera Puntheeranurak proposed Music Recommendation Systems: Overview and Challenges.They have used User-Based Collaborative Filtering technique and acheived an accuracy of 85 % with 21000 songs as dataset.

M. Sunitha, T. Adilakshmi and Mir Zahed Ali proposed Item-Based Collaborative Filtering.They have used Item-Based Collaborative Filtering technique and acheived an accuracy of 89.9 % with 2000 songs as dataset.

Professor Reda Alhaj proposed Content-Based Music Filtering .They have used Content-Based Music Filtering technique and acheived an accuracy of 93.5 % with 48000 songs as dataset.

Muhammad Umair Hassan, Numan Zafar proposed Hybrid Music Recommendation System .They have used Content-Based Music Filtering technique and acheived an accuracy of 96.9 % with 2,814 songs as dataset.

Music lyrics are also considered to be one of emotional presentation because they include some kinds of implicit thinking, thus we can fully understand emotions and their associated thinking in each song (Nunes and Jannach, 2017; Tintarev and Masthoff, 2008). Cano et al. (2017) mentioned that there is a strong relation between the user mood and listening to the music

M. Sunitha, T. Adilakshmi and Mehar Unissa proposed Deep Learning-Based Music Recommendation System .They have used Deep Learning technique and acheived an accuracy of 99.9 % with 34000 songs as dataset.

Y. Sri Lalitha, Y. Gayathri, Yamsani Nitish, Devarakonda Naveen, Bhagavathula Krishna Vamsi and Mekala Nithish Kumar proposed Study of Transfer Learning Models for Disposition-Based Music Recommendation Systems (27 September 2023)The main objective of the project is to develop an application capable of changing the user's mood by using music as a catalyst with the help of the user's facial features.

Sl No.	Title	Author	Model	Accuracy	No. of Dataset used
1	User-Based Collaborative Filtering Recommendation System	Thi Thuan To and Sutheera Puntheeranurak	User-Based Collaborative Filtering	85.5%	21,00
2	Enhancing Item-Based Collaborative Filtering	M. Sunitha, T. Adilakshmi and Mir Zahed Ali	Item-Based Collaborative Filtering	89.9%	2000
3	Content-Based Music Filtering	Professor Reda Alhajj	Content-Based Filtering	93.5%	48,00
4	Hybrid Music Recommendation System	Muhammad Umair Hassan, Numan Zafar	Hybrid filtering	96.90%	2,814
5	Deep Learning-Based Music Recommendation System	M. Sunitha, T. Adilakshmi and Mehar Unissa-Tahsin Rahman, and Riasat Khan et al	Deep Learning	99%	3400

Table 2.1: Literature survey

Chapter 3

PROPOSED SYSTEM

3.0.1 Proposed System

The proposed system filtering leverages intrinsic features of songs like genre, tempo, mood, and artist specifics to offer highly personalized recommendations. This method excels in delivering tailored suggestions based on the specific characteristics of music pieces, enabling users to discover new tracks that closely align with their preferences. By focusing on these content attributes, it ensures a more transparent recommendation process, clarifying to users why certain songs are suggested based on similarities in musical elements.

Additionally, content-based music filtering can mitigate the cold start problem by analyzing the inherent features of songs, allowing it to provide recommendations even to new users or for newly released tracks without relying heavily on prior user interaction data

Chapter 4

MATERIALS AND METHODS

4.1 Tools

1. **Jupyter Notebook:** The study was conducted using the Jupyter Notebook, providing an interactive and collaborative platform for developing and executing Python code. This environment facilitated seamless experimentation with deep learning models and the implementation of the proposed classification system.

2. **Streamlit:**Streamlit is an open-source Python library that enables developers to create web applications for data exploration and machine learning prototyping with minimal effort. It simplifies the process of building interactive web apps by allowing users to convert Python scripts into web apps quickly.

3. **VS Code:**Visual Studio Code is a popular, free, and open-source code editor developed by Microsoft. It's highly extensible, customizable, and designed to cater to various programming languages and development workflows. Here are some key features

4. **CPU: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz:** The computational power for this study was provided by an Intel(R) Core(TM) i7-7700HQ CPU running at 2.80GHz.

4.2 System architecture

The architecture of a music recommendation system revolves around extracting intricate features from music tracks, encompassing metadata like genre, tempo, mood, and audio signals using signal processing or pre-trained models. This data, alongside user preferences and listening history, undergoes feature representation and vectorization, converting it into numerical representations. Utilizing similarity measures like cosine similarity or Pearson correlation, the system calculates the likeness between these representations, matching track features to user profiles. The recommendations generated are based on the highest similarity scores, offering users tracks that align with their established preferences and the intrinsic characteristics of the music itself, ensuring personalized and contextually relevant suggestions.

The project involves a sequential process of data collection, training, testing,classification and model evaluation.

Data Collection Gathering user listening habits, track information, and preferences.Accumulating a comprehensive dataset reflecting diverse user tastes and music attributes..

Feature Extraction Extracting relevant features from users' listening patterns (time, frequency, genres liked/disliked). Extracting track features such as genre, tempo, artist, popularity, etc.

Training Training various recommendation algorithms (collaborative filtering, content-based, hybrid) analogous to training CNN models. Fine-tuning models to capture intricate user preferences and music characteristics

Testing Once trained, the models undergo evaluation using a separate testing dataset.

This dataset is disjoint from the training data, ensuring the models generalize well to unseen examples and do not overfit to specific training patterns.

Classification During the testing phase, each music in the testing dataset is passed through every trained CNN model. The models then produce predicted class labels for each music based on the learned features during the training phase.

Model Evaluation The performance of each CNN model is assessed using various metrics, such as accuracy, precision, recall, and F1-score. These metrics serve to quantify the models' effectiveness in correctly classifying music into their respective classes. The evaluation process provides insights into the strengths and weaknesses of each CNN architecture.

4.2.1 Block diagram

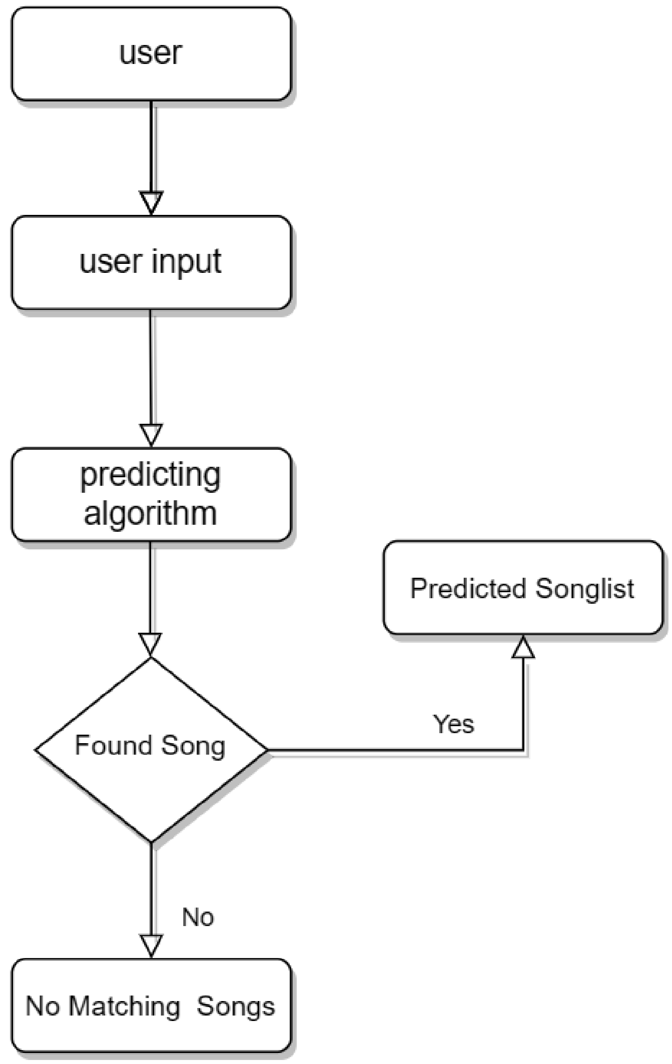


Figure 4.1: Block diagram

4.3 Algorithm / Pseudo code of the Project Problem

The algorithm used for the Project Problems is:

4.3.1 Cosine Similarity

The metric cosine similarity assesses how similar two or more vectors are. The cosine similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector. Cosine Similarity is a statistic that determines how similar two or more vectors are. The cosine similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector. The metric cosine similarity assesses how similar two or more vectors are. The cosine similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector. The metric cosine similarity assesses how similar two or more vectors are. The cosine similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector. Cosine Similarity is a statistic that determines how similar two or more vectors are. The cosine similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector The metric cosine similarity assesses how similar two or more vectors are. The cosine

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Figure 4.2: cosine similarity

similarity is the cosine of the angle between vectors. In most cases, the vectors are non-zero and belong to the inner product space. The cosine similarity is formally described as the difference between the dot product of vectors and the product of the euclidean norms or magnitude of each vector. The range of cosine similarity values is limited to 0 to 1. Similarity is calculated using the cosine of the angle between the two non-zero vectors A and B.The characteristics of each item are used in content-based filtering to locate items that are comparable. We may propose an item based on how similar it is to all other things in the dataset by assigning a score to how similar each item is. We utilise the properties (loudness, pace, etc.) of each song in a Spotify playlist to calculate the average score of the entire playlist. Then we suggest a song that has a comparable score to the playlist but isn't on it. Assume the two vectors intersect at a 90-degree angle. In such case, the cosine similarity will be 0, indicating that the two vectors are orthogonal or perpendicular

Chapter 5

RESULT AND ANALYSIS

5.1 Results

In the system, First user inputs the song which he/she wants; once the required song is inputted by the user, that ten similar songs are recommended to him. Initially, the process takes into consideration by taking three main features, that is Title, Artist, and Top Genre, which is done by taking Angular distance and Euclidean distance. For this, we have taken the class Count Vectorizer and method cosine similarity. Count vectorizer is stored in an object which is used to count the number of terms that appeared in a particular feature; after that, structured data is used by cosine similarity to find the similarity score. Before the data is processed by the count vectorizer class, since we are using multiple parameters/ features to find the similarity score, a function is created to merge the contents of all the rows of the specified features. In case any NaN values are found, they are replaced with an empty string.

Chapter 6

CONCLUSION

6.1 Conclusion

In conclusion, this study delved into the application of deep learning for the classification of bitter gourd at different developmental stages, employing VGG16 and InceptionV3 architectures. The achieved accuracies of 87% for VGG16 and 92% for InceptionV3 on a dataset of 1379 images underscore the efficacy of deep learning in automating the identification of crop stages.

The comparative analysis revealed that InceptionV3, with its inception modules capturing features at various scales, outperformed VGG16 in bitter gourd classification. This highlights the importance of selecting an appropriate deep learning architecture tailored to the intricacies of the specific agricultural task.

The practical implications of this research are significant, contributing to the advancement of precision agriculture. The accurate classification of bitter gourd at different stages provides valuable insights for farmers, enabling informed decision-making in crop management. The potential for automation in agriculture is further emphasized, with deep learning models offering a reliable and efficient means of assessing crop quality.

Looking ahead, future work involves refining the models, exploring diverse datasets, and adapting the approach for real-world agricultural scenarios. The continuous evolution of deep learning techniques in agriculture remains pivotal for addressing the challenges of scalability, robustness, and adaptability in the field.

This study marks a step forward in the integration of technology into agriculture, showcasing the potential for sophisticated neural networks to contribute to the optimization of crop management practices. The findings presented here lay the groundwork for future advancements in automated crop stage identification and underscore the ongoing synergy between technology and agriculture.

Chapter 7

FUTEURE SCOPE

7.1 Future Enhancement

The future of music recommendation systems holds exciting possibilities driven by advancements in AI, data analytics, and user-centric personalization. Innovations in deep learning and natural language processing may enhance systems' understanding of nuanced user preferences, incorporating context and sentiment analysis from user interactions beyond traditional metrics. Integration of multimodal data, like combining audio analysis with visual or textual information, could create richer user profiles for more precise recommendations. Embracing decentralized and privacy-preserving approaches, like federated learning, may empower systems to learn from diverse data sources while preserving user privacy.

Collaborations between music platforms and AI research could lead to more seamless and intuitive interfaces, offering interactive recommendation experiences, and enabling users to co-create playlists or discover music collaboratively. Additionally, leveraging emerging technologies like augmented reality (AR) or virtual reality (VR) might transform how users interact with and experience recommended music, immersing them in dynamic, personalized musical journeys.

Bibliography

[1] Makarand Velankar,Parag Kulkarni ,”Music Recommendation Systems: Overview and Challenges,” 23 September 2022

[2] L. Breiman.”Random forest,” Machine Learning,vol.45,2001.

[3] Vikas Sindhwani Prem Melville, ”Recommender Systems,” in Encyclopedia of Machine Learning. Yorktown Heights, USA: IBM T. J. Watson Research Center, 2010.

[4] Aiolli, F. (2012). A preliminary study on a recommender system for the million songs dataset challenge. PREFERENCE LEARNING: PROBLEMS AND APPLICATIONS IN AI

[5] D. Paul and S. Kundu, ”A Survey of Music Recommendation Systems with a Proposed Music Recommendation System,” in Advances in Intelligent Systems and Computing, vol. 937, 1st online: 17 July 2019.

[6] M. Sunitha, T. Adilakshmi, and M. Z. Ali, ”Enhancing Item-Based Collaborative Filtering for Music Recommendation System,” in Smart Innovation, Systems and Technologies, vol. 224, 1st online: 14 July 2021.

[7] N. Sheth, R. Agrawal, and S. K. Vishwakarma, ”A Content-Based Music Recommendation System Using RapidMiner,” in Lecture Notes in Electrical Engineering, vol. 862, 1st online: 14 June 2022.

[8] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang. “Deep Learning Based Improved Classification System for Designing Content Based Filtering.” *IEEE Access,* vol. 6, 2018, pp. 67940-67950. doi: 10.1109/ACCESS.2018.2879324.

[9] M. Sunitha, T. Adilakshmi, and M. Unissa, ”Hybrid Deep Learning-Based Music Recommendation System,” in Lecture Notes on Data Engineering and Communications Technologies, vol. 117, 1st online: 22 May 2022.

[10] C. H. P. D. Wishwanath et al., ”A Personalized and Context Aware Music Recommendation System,” in Lecture Notes in Computer Science, vol. 12195, 1st online: 10 July 2020.

Chapter 8

Annexure

8.1 Sample Code

8.1.1 app.py

```
import pickle
import streamlit as st
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

CLIENT_ID = "70a9fb89662f4dac8d07321b259eaad7"
CLIENT_SECRET = "4d6710460d764fbbb8d8753dc094d131"



# Initialize the Spotify client
client_credentials_manager = SpotifyClientCredentials(client_id=
    CLIENT_ID, client_secret=CLIENT_SECRET)
sp = spotipy.Spotify(client_credentials_manager=
    client_credentials_manager)

def get_song_album_cover_url(song_name, artist_name):
```

```
    search_query = f"track:{song_name} artist:{artist_name}"
    results = sp.search(q=search_query, type="track")

    if results and results["tracks"]["items"]:
        track = results["tracks"]["items"][0]
        album_cover_url = track["album"]["images"][0]["url"]
        print(album_cover_url)
        return album_cover_url



  else:
        return "https://i.postimg.cc/0QNxYz4V/social.png"


def recommend(song):
    index = music[music['song'] == song].index[0]
    distances = sorted(list(enumerate(similarity[index])),
  reverse=True, key=lambda x: x[1])
    recommended_music_names = []
    recommended_music_posters = []
    for i in distances[1:6]:
        # fetch the movie poster
        artist = music.iloc[i[0]].artist
        print(artist)
        print(music.iloc[i[0]].song)
        recommended_music_posters.append(get_song_album_cover_url
```

```
    (music.iloc[i[0]].song, artist))
            recommended_music_names.append(music.iloc[i[0]].song)

    return recommended_music_names,recommended_music_posters

st.header('Music Recommendation System')
music = pickle.load(open('df.pkl','rb'))
similarity = pickle.load(open('similarity.pkl','rb'))

music_list = music['song'].values
selected_movie = st.selectbox(
    "Type or select a song from the dropdown",
    music_list
)

if st.button('Show '):
    recommended_music_names,recommended_music_posters = recommend
    (selected_movie)
    col1, col2, col3, col4, col5= st.columns(5)
    with col1:

        st.text(recommended_music_names[0])
        st.image(recommended_music_posters[0])
        st.markdown(f''' <a href=https://wynk.in/music/song/never
    -know-why/sm_A10328E0003700101N?qnever+know+><button style="
    background-color:#F05654;display: inline-block;
  padding: .02px 40px;
```

```
font-size: 15px;
cursor: pointer;
text-align: center;
text-decoration: none;
outline: none;
color: #fff;
background-color: #F05654;
border: none;
border-radius: 10px;
background-color: #F05654;
">PLAY</button></a> ''', unsafe_allow_html=True)

  with col2:
      st.text(recommended_music_names[1])
      st.image(recommended_music_posters[1])
      st.markdown(f''' <a href=https://wynk.in/music/song/rock-me/sm_A10328E0006212650C?qrock+me><button style="background-color:#F05654;display: inline-block;
padding: .02px 40px;
font-size: 15px;
cursor: pointer;
text-align: center;
text-decoration: none;
outline: none;
color: #fff;
background-color: #F05654;
border: none;
```

8.2 Project Screenshots

8.2.1 keyword seclection

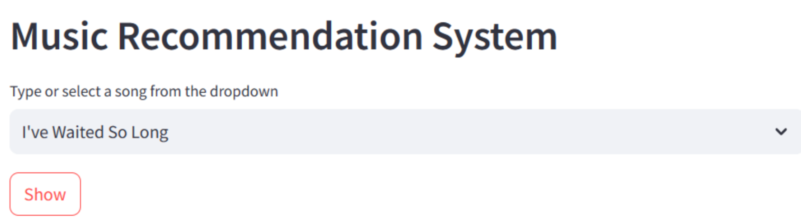
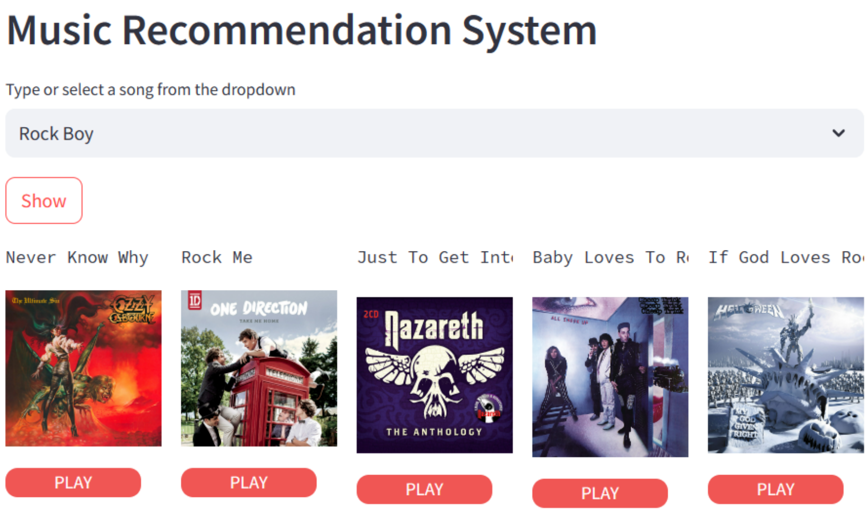


Figure 8.1: Keyword seclection

8.2.2 Prediction



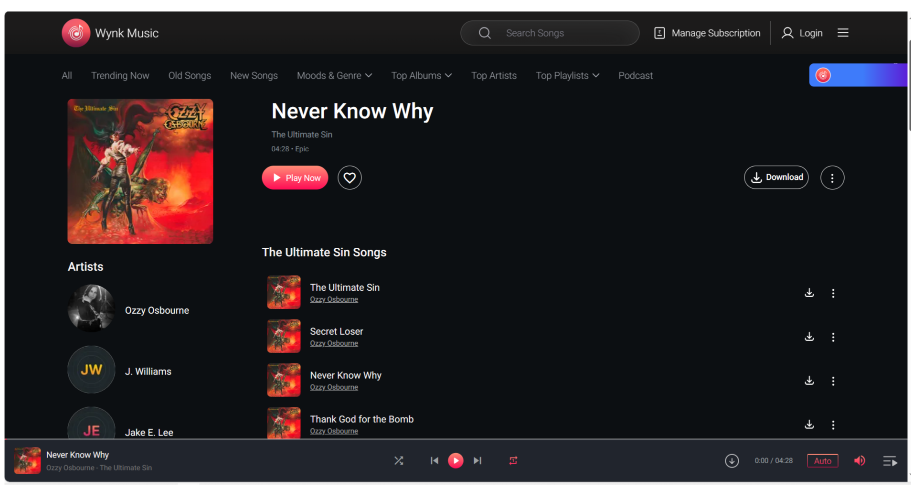


Figure 8.2: Prediction

8.3 GitHub History

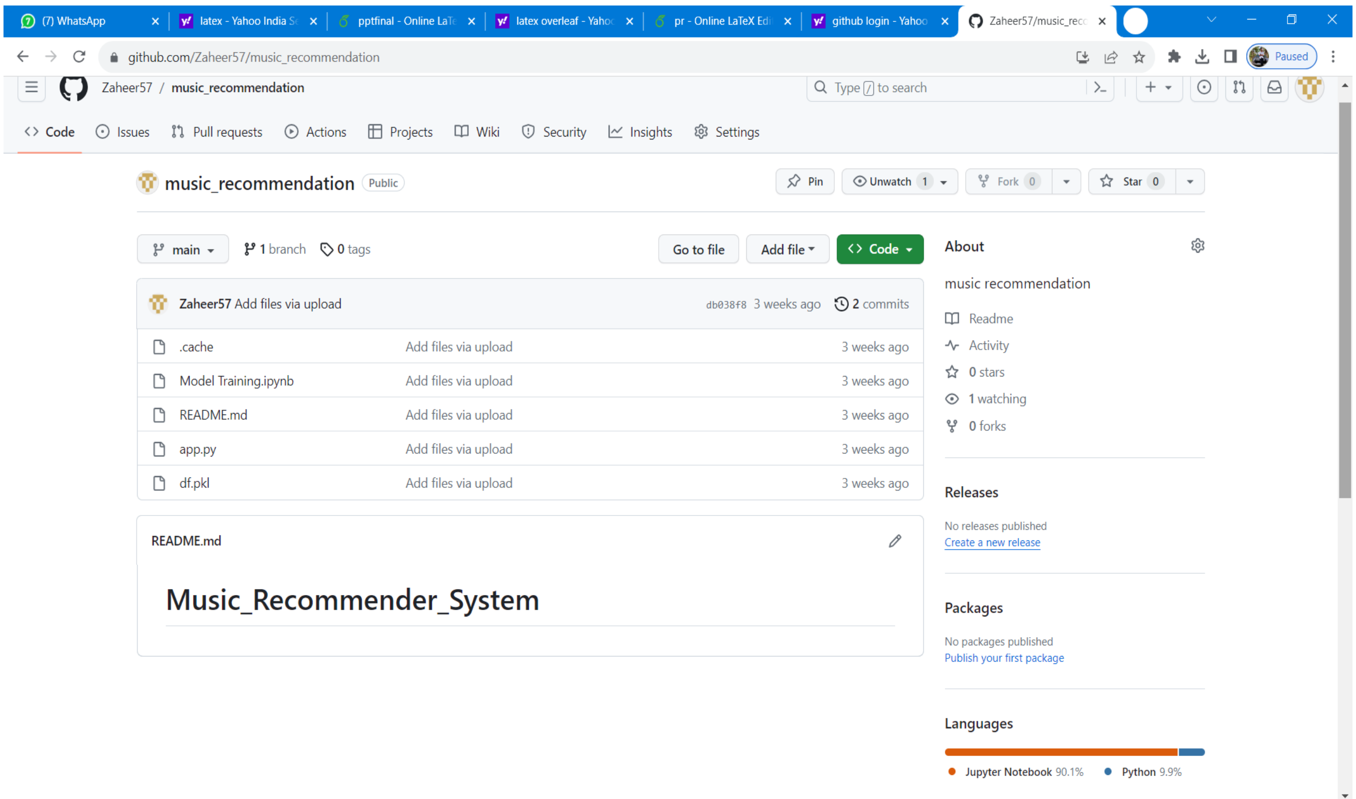


Figure 8.3: GitHub History