



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: VII Month of publication: July 2023

DOI: <https://doi.org/10.22214/ijraset.2023.54740>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Music Recommendation System using Python

Mr. Chirag Desai¹, Shubham Bhadra², Mehul Parekh³
K J Somaiya College of Engineering, Mumbai, Maharashtra, India

Abstract: *In contrast to the past, the availability of digital music has increased thanks to online music streaming services that can be accessed from mobile phones. It becomes tedious to sort through all of the songs and results in information overload. Many people consider music to be an integral part of their lives and place great value on it. When a person is joyful, depressed, or emotional, he prefers to listen to music to unwind his mind. Users frequently use search engines to find songs of interest to them, but as technology has advanced, other approaches to searching have been adopted. As a result, creating a music recommendation system that can browse song albums automatically and suggest appropriate songs to users is quite advantageous. Utilising such system based on the user's mood, it can anticipate and then present the appropriate songs to its users. Our work is unique in that the developed recommender system is based on the user's mood. There are two known methods for creating content-based music recommendation systems. The first methodology, which makes use of a powerful classification algorithm, is quite popular, while the second one uses deep learning algorithms to augment the performance of the recommender system.*

Keywords: *Deep Learning, Classification, Recommendation System, Machine Learning*

I. INTRODUCTION

Music has an essential role in augmenting an individual's life because it is a source of amusement for song listeners and buffs, and it can even be therapeutic at times. We want to build a music recommendation system based on user sentiment. With the rise of internet streaming services, music recommendation systems have seen a surge in use in recent years. Many kinds of recommender systems are employed nowadays, therefore deciding on a better technique to enhance the fundamental neighbourhood model of our particular application is a key component of our research. There have been very few pieces of literature that apply for a deep learning framework in music recommendation, particularly on audio. Our system consists of three modules: User authentication, music classification, and a recommendation module that includes mood-based and searchbased recommendations. A musicbot is supplied in which the user may enter their mood [calm, sad, energetic, joyful] and the system will provide the finest songs to them. The Recommendation Module recommends songs to the user by mapping their feelings to the mood type of the song and taking into consideration their personal tastes. As a result of ever-increasing discoveries in multimedia and technology, several music players with features such as volume modification, genre classification, and suggestion have been developed in today's globe.

Although these functions meet the user's fundamental needs, the user must actively browse through the song's library and select songs based on the present behaviour and mood. That is an individual's requirement; a user suffers sporadically from the urge and desire to browse through his library, depending on the mood and emotions. Because this activity was time-consuming and an individual frequently struggled to find an adequate list of songs, we decided to create a mood-based music recommendation system.

II. LITERATURE REVIEW

This section explains all of the research papers we took into consideration before proceeding with our project.

The project mentioned in the paper (1) aims to record the emotion displayed on a person's face. The web camera UI on computer systems is used by a music player to record human emotion. The software takes a user's image and uses image processing techniques to extract features from a target person's face in an effort to determine what emotion they are attempting to convey. The project is developed to augment the user's mood by playing music that fits their needs while simultaneously recording their image.

The user's recommended song list is the form of music recommendation. In this paper (2), we investigate an emotion-aware PMRS that can suggest songs from various genres based on the user's preference and current emotion. The user's data includes both the user's music listening history and personal data about them, including their age, gender, location, environment, when the data was collected, their emotions, etc. Every user of the system whose data is maintained by the PMRS is stored in the data repository. The PMRS employs the weighted feature extraction (WFE) methodology and the DCNN methodology based on the two types of data that are available. The DCNN method categorises the music data based on the song's metadata and audio signals.

This paper's methodology (3) focuses on a number of song characteristics. In this case, the user interface enables users to play songs while also making song recommendations based on the currently playing song. Each song pair's audio features and lyrics score are given to the algorithm. The algorithms used are KNN regression and artificial neural networks. The algorithm forecasts the degree of similarity between two songs. The user is suggested songs that have the highest similarity ratings to the song that is currently playing.

The music recommendation system that automatically scans music libraries and recommends appropriate songs to users was discussed in the paper cited in the previous sentence (4). Using the features of the music they have already heard, the music provider will anticipate their customers' needs and then provide them with the appropriate songs. The research creates a framework for music suggestions that can make suggestions based on how similar audio signal features are. Both the convolutional neural network (CNN) and the recurrent neural network (RNN) are used in this study.

The difficulties with MRS are studied in this paper (5). Even though modern MRSs greatly aid users in locating interesting music in these vast catalogues, MRS research still faces many obstacles. MRS research becomes a significant undertaking, and related publications are quite scarce, when it comes to developing, incorporating, and evaluating recommendation strategies that integrate data beyond straightforward user-item interactions or content-based descriptors but delve deeply into the very essence of listener needs, preferences, and intentions. This is partially because current MRS approaches, which are frequently built around the fundamental idea of user-item interactions or, occasionally, content-based item selection, do not adequately take into account the fact that users' musical references and needs depend heavily on a wide range of variables. The three main issues that we think the research field of music recommender systems is currently facing are resolving the cold start problem, automatic playlist continuation, and properly evaluating music recommender systems. Both situation-aware MRS, which holistically models contextual and environmental aspects of the music consumption process and infers listeners' needs and intentions, and psychologically inspired MRS, which considers factors like listeners' emotions and personalities in the recommendation process.

It takes an extended period to sort through all of this digital music, and it makes you information-tired. Therefore, developing a system for automatically searching through music libraries and suggesting appropriate songs to users is very helpful. Using a music recommender system, a music provider can anticipate and then present the appropriate songs to their users based on the traits of the music that has already been heard. Research is being done in this paper (6) to create a music recommendation system that can make suggestions based on the similarity of features on audio signals. In this study, similarity distance is used to determine whether two features are similar and convolutional recurrent neural networks (CRNN) are used to extract features. According to the study's findings, users favour recommendations that take music genres into account over those that are solely based on similarity.

In this research paper (7), Convolutional recurrent neural networks are used in given method to classify music into a variety of genres, including Classical, Electronic, Folk 1000, Rock, Jazz, Hip-Hop, and songs with instruments. This study used two models, the CNN and CRNN models, to transform auditory input into spectrogram images. Both models received an F1 score of 72% and 75%. The cosine similarity model was used to determine how similar the songs were, which assisted in genre clustering. Less cosine similarity was scored for songs that were comparable to one another.

The research presented in this paper (8) makes use of a recommender system that receives the user's profile as an input and processes it to provide the first set of recommendations based on user's mood. This is done after the first recommendation phase, which sorts the artists in the first output and performs content-based filtering of the songs, and after this phase, it provides songs that fit the user's mood. Through an interactive drop-down list, users can access the three components of the web system—input, visualisation view, and recommendation panels—using a web browser. Users enter the names of artists to create their profiles. Based on given mood data associated with profile singers, the technology places a user icon in a previously computed visual mood environment and suggests new artists.

In this work (9), they conducted research on a song recommendation system that makes music suggestions to users based on their usage patterns and song history. This model is distinctive in that it incorporates the user-produced output into its subsequent predictions. Three factors are used to determine a user's behaviour: their posts and tweets on social media apps, their likes of those posts, and their uploaded photos to social media sites. The Microsoft Azure Cloud platform is used to develop the recommendation engine. The design also makes advantage of PySpark for processing large amounts of data, and it trains the data using several machine learning techniques like naive bayes, svm, and random forest. It achieves an accuracy of around 74%.

A facial mood recognition system that suggests music based on the mood the camera detects is covered in the research paper (10). The support vector machine module receives the live picture processed as part of the system architecture. It extracts the songs from the database and deduces the sentiment from the visuals. The face is recognised using the Eigenfaces method. Gabor Wavelets, Direct Cosines Transform, and Local Binary Patterns are the methods used for local feature extraction.

III. METHODOLOGY

A. Dataset

We exhaustively investigated multiple platforms for eligible datasets during our data exploration process, including Kaggle, Github, and Google datasets. We also attempted to link the Spotify API to get relevant data from Spotify. We were fortunate to come across a github user named **crystalvch** who had datasets that completely met our criteria. The dataset featured a file especially dedicated to music moods, which proved useful in training our mood recommendation model. Within the dataset, we also discovered another file containing a collection of songs, which we used to test and evaluate our recommendation engine.

Attributes	Description
Acoustiness	A scale from 0 to 1 indicating if the track is acoustic.
Danceability	On a scale of 0 to 1, how suitable the track is for dancing.
Energy	Energy is a 0-to-1 scale that represents a perceptual assessment of intensity and activity.
Instrumentalness	It indicates whether or not the audio contains vocals on a scale of 0 to 1.
Liveness	It describes the presence of the audience on a scale of 0 to 1.
Loudness	The total volume of a track measured in decibels (dB).
Speechiness	It specifies the number of words uttered in the song. As a result, it ranges from 0 to 1.
Valence	A scale from 0.0 to 1.0 that describes the melodic positivity given by a track.
Tempo	The estimated overall beats per minute (BPM) tempo of a track

Table 1.: Dataset Description

B. Implementation

1) Technologies Used

- a) *Web Technologies:* HTML, JavaScript, jQuery, Django, SQLite: In our project, we extensively utilized a range of web technologies to develop a user-friendly and interactive interface. HTML (Hypertext Markup Language) formed the foundation of our web pages, allowing us to structure and present content effectively. JavaScript, a powerful scripting language, enabled us to add dynamic functionality and interactivity to our web application. We employed jQuery, a popular JavaScript library, to simplify DOM manipulation, handle events, and make AJAX requests. For the backend development, we chose Django, a high-level Python web framework. Django provided a robust and scalable foundation for building our web application, offering features such as URL routing, model-viewcontroller (MVC) architecture, form handling, and user authentication. We also utilized SQLite, a lightweight and reliable relational database management system, integrated with Django to store and manage our application's data efficiently.
- b) *Programming Languages:* Python: Python served as our primary programming language throughout the project. Its simplicity, readability, and vast ecosystem of libraries and frameworks made it an ideal choice for various tasks. We leveraged Python's extensive standard library and rich thirdparty modules to accomplish a wide range of functionalities, from data preprocessing and model training to web development and database management.
- c) *Modules Used:* NumPy, Pandas, scikit-learn, TensorFlow, pickle, Matplotlib: To facilitate data manipulation and analysis, we relied on several essential Python modules. NumPy provided powerful arrays and mathematical functions, enabling efficient numerical computations and data handling. Pandas, built on top of NumPy, offered versatile data structures and data analysis tools, simplifying data preprocessing and manipulation tasks. For machine learning tasks, scikit-learn (sklearn) proved invaluable. It provided a comprehensive set of algorithms for classification, regression, clustering, and dimensionality reduction, along with tools for model evaluation and selection. TensorFlow, a popular deep learning framework, empowered us to build and train neural networks for more complex modeling tasks. We utilized pickle, a Python module, for serializing and deserializing Python objects, allowing us to save trained models and reload them when needed. Finally, Matplotlib enabled us to create visually appealing plots and charts to visualize data, model performance, and other relevant insights.

2) Methodology

a) Mood Based Recommendation Model

To assure the quality and applicability of the data for training, it was crucial to carry out data cleaning and preprocessing activities prior to training our model. For this, we used the 'datamood.csv' dataset, which had the relevant data on song moods. The cleaning procedure included a number of processes, such as resolving missing numbers, getting rid of duplicates, and dealing with outliers if any were found. In order to ensure that the input features were on a similar scale and to eliminate bias throughout the training process, we also performed feature scaling. Along with that, we encoded the output features, transforming the categorical mood labels into numerical representations appropriate for our model's training. After the data cleaning stage, we moved on to training our model. We divided the cleaned dataset into training and testing sets, allocating training sets 80% of the data. In order to stack many layers of neural networks, we used a sequential model, a kind of deep learning model. This approach is ideal for processing sequential input, such as songs in our example.

We trained the data into four batches in order to train our model effectively. We were able to process the training data in smaller batches and iteratively alter the model's parameters because each batch only comprised a fraction of the training data. To indicate how many times the model would iterate over the data, we set the number of epochs for each batch to 300.

We assessed the performance of our model after finishing the training for all four batches. By correctly classifying songs according to their moods, we were able to reach an accuracy of 82.30% on the testing data. This accuracy score evaluated the efficacy of our model's generalisation to new data.

We converted our model into a pickle file so that it could be easily down-loaded and used on our website. We can serialise Python objects, including trained models, into a binary format that can be saved and loaded later using the Pickle Python module. We could quickly load and use our trained model within our web application by saving it as a pickle file, ensuring seamless interaction between the recommendation system and our web application.

b) Cosine Based Similarity Model

The 'spotify songs.csv' dataset is loaded into a pandas DataFrame for exploration during our data preprocessing stage. Following our comprehension of the structure and content of the dataset, we concentrate on a particular genre based on a chosen track. By filtering the original DataFrame to only contain songs from the chosen genre, we generate a new DataFrame, df1. To assure data quality, we delete any missing values in df1. Creating the feature columns required for recommendation is our next step. We list the required features for calculating song similarity in the feature cols list. We will train a recommendation model using these characteristics. We utilize the MinMaxScaler from scikit-learn to normalize the feature values, ensuring that all features are on a comparable scale and preparing the data for training. The normalized data are then used to create a cosine similarity matrix, which will be used to produce song recommendations.

Next, we create the function generate recommendation, which accepts a song title and a similarity model as inputs. From the pre-computed indices series, we retrieve the index for the supplied song title. We compare the given song to all other songs using the selected similarity model, producing a list of song similarity scores. We then sort the similarity scores in descending order and choose the top 10 tracks, excluding the given song itself. We retrieve the track names of the most highly suggested songs from the dataframe.

To create YouTube links for the suggested songs, we define the function youtube_link. It accepts a search term (song title) as input, formats it, and then uses urllib to search YouTube. We extract the video ID from the search results and use it to create the YouTube link for the most frequently recommended song.

We use the generate recommendation function on the song "Memories - Dillon Francis Remix" to create a list of suggested tracks using the cosine similarity model. We extract the artist name and track name for each suggested song from df1 and combine them as 'artist name track name'. We save the recommended song information, including the song name and YouTube URL, in a dictionary called song dict. Throughout the implementation, we handle certain exceptions that could arise while creating YouTube links. Finally, we print the song dict dictionary, which includes the suggested song titles and the corresponding YouTube links.

c) Website

For building the website, we used django as our main technology for implementing the web development. Using HTML and Bootstrap 5 CSS, the website theme was made like a music website theme. The model which we made, was then applied on the main dataset i.e 'spotify songs.csv' to get the moods of all testing songs which were going to be implemented in our website. We created the function for recommending the songs based on the user's mood which was given as input in the form.

We also created a function that uses cosine similarity to recommend similar songs based on the user’s previously searched song. The SQLite database contained the csv which were going to be used , For our genre based classification of songs , the songs were classified by filtering the database which included our dataset . The songs which were recommended contained the respective youtube links which were generated using the search keyword functionality of the youtube to give us the best results and provide the link . The recommendation system was exclusive to the user’s which were only logged in . So proper login , registration and logout functionalities were implemented with proper email , password and username validation .

IV. RESULT ANALYSIS

A. Mood Recommendation System:

During the training of our dataset , as mentioned above we ran our model for 300 Epochs 4 times and with each round of 300 epochs we got better results and the accuracy graph as well as the loss graph showed a straight line which gave us an indication to stop after 4th Round . Final Accuracy : 82.30%

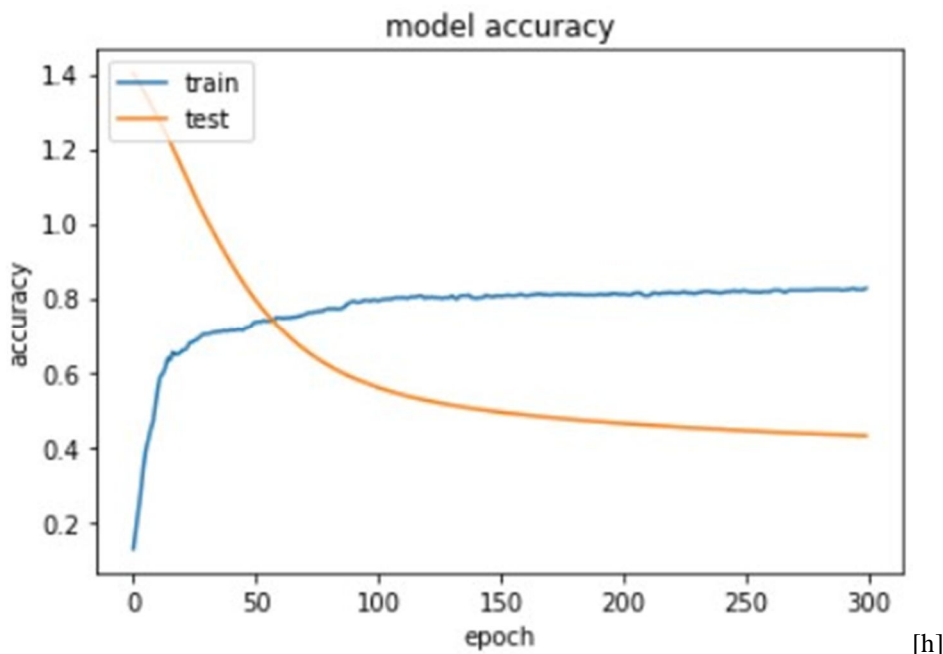


Figure 1.: First 100 Epochs

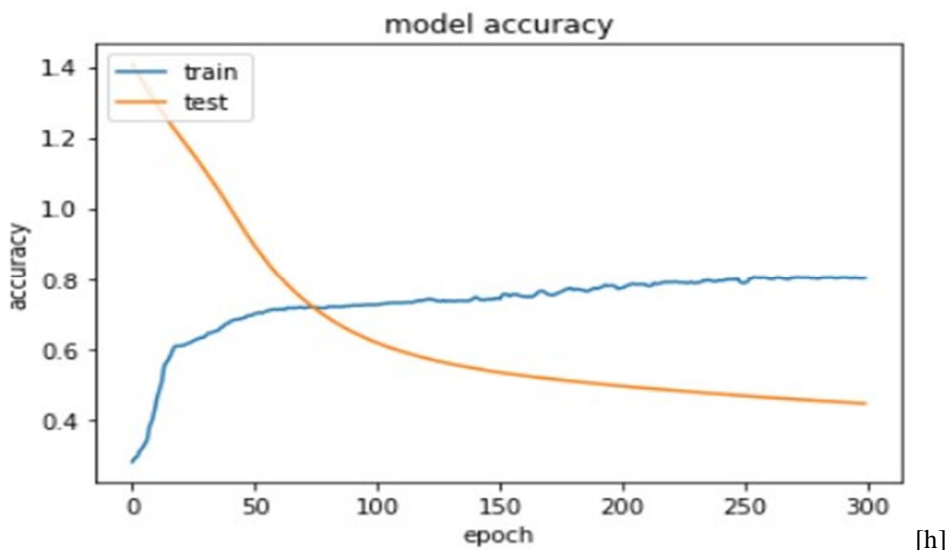


Figure 2.: Second 100 Epochs

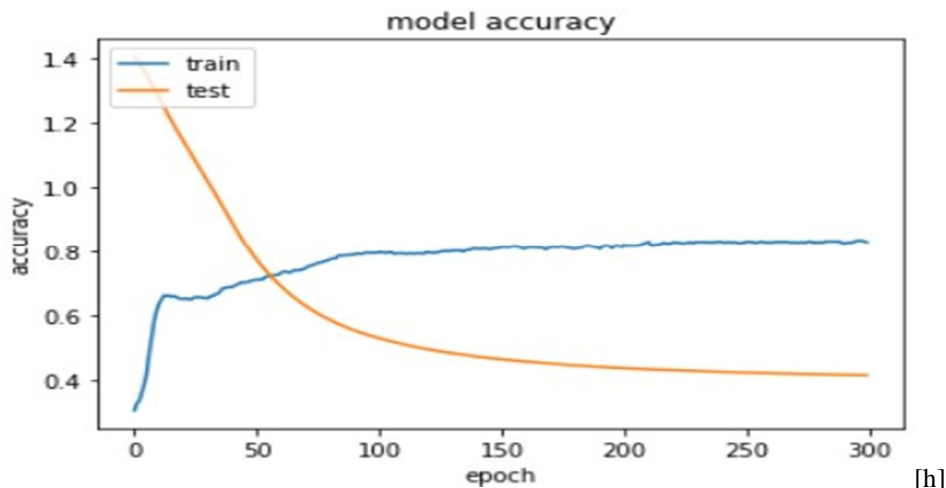


Figure 3.: Third 100 Epochs

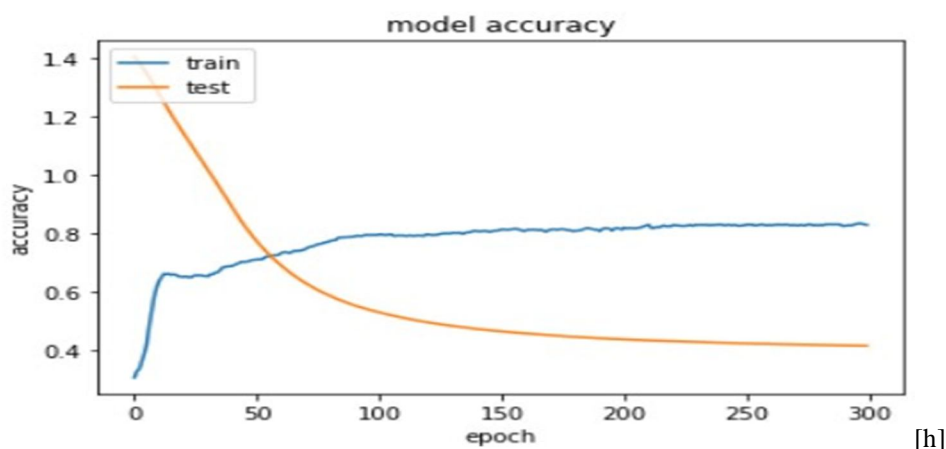


Figure 4.: Fourth 100 Epochs

Song name	Happy	Energetic	Sad	Calm	Mood
Dawn - Hardwell Edit	33.0142	4.7653	60.1599	2.0606	Sad
Leaving You - Thomas Newsc	39.3321	59.1189	1.5489	0	Energetic
Unleash (Life In Color Anthen	62.3542	14.8723	22.7729	0.0006	Happy
It Won't Stop (feat. Chris Bro	47.6861	47.4077	4.9059	0.0003	Happy
Everybody Is In The Place - R	16.5547	82.8257	0.6183	0.0013	Energetic
Hey Brother	56.0924	36.4358	7.4715	0.0002	Happy
Booyah - Radio Edit	33.469	64.7441	1.7868	0.0001	Energetic
Wasted	50.5999	41.7319	7.6673	0.0009	Happy
Many Ways - Radio Edit	59.5234	29.6314	10.8449	0.0003	Happy
City Of Lights - Official Radio	33.8213	63.9737	2.2046	0.0004	Energetic
Closer - Sultan & Ned Shepar	62.8756	27.5526	9.5717	0	Happy
Sweet Surrender - Radio Edit	55.0121	37.9807	7.0068	0.0003	Happy
Only For You - Maor Levi Ren	52.3146	42.9072	4.7781	0.0001	Happy
Typhoon - Original Mix	53.3894	38.4324	8.1775	0.0007	Happy
Tera Ban Jaunga	10.3095	0.17	89.4649	0.0555	Sad
Shayad	9.2793	0.1107	90.5723	0.0377	Sad
Haan Main Galat	60.5125	30.3153	9.1719	0.0003	Happy
Shayad (From "Love Aaj Kal")	16.4018	0.2495	83.3436	0.0051	Sad
Kabira	21.381	0.4245	78.1901	0.0044	Sad
Enna Sona	8.4828	0.0532	91.4565	0.0074	Sad
Makhna	60.5888	32.3558	7.0553	0.0001	Happy
Genda Phool (feat. Payal Dev	48.2459	49.6657	2.0885	0	Energetic
Tum Hi Ho	45.1223	2.9545	51.9211	0.0022	Sad
Munna Badnaam Hua	68.0258	21.1029	10.8712	0.0001	Happy
Kala Chashma	45.9434	52.8263	1.2303	0	Energetic
Chogada (From "Loveyatri")	60.0105	35.7687	4.2209	0	Happy
Lut Gaye	64.9006	24.0416	11.0577	0.0002	Happy

Figure 5.: Percentage Probabilities for songs using mood recommendation model

B. Cosine Similarity

```
print("Recommended Songs:")  
s=generate_recommendation('Memories - Dillon Francis Remix',cosine).values  
print(s)
```

Recommended Songs:

```
1  
['Sugar' 'Year 3000' 'Secreto Perfecto' 'home ft. WALK THE MOON'  
'Walk Like an Egyptian' 'Higher' 'Act My Age' 'Galway Girl'  
'Blah Blah Blah (feat. 3OH!3)' 'Too Close']
```

Figure 6.: Cosine similarity model

C. Website

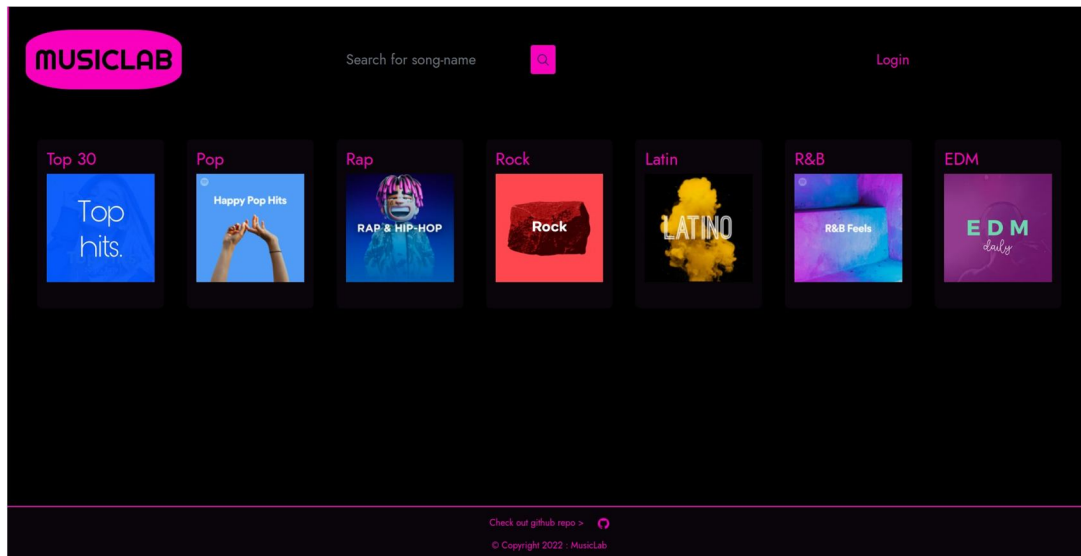


Figure 7.: Landing Page

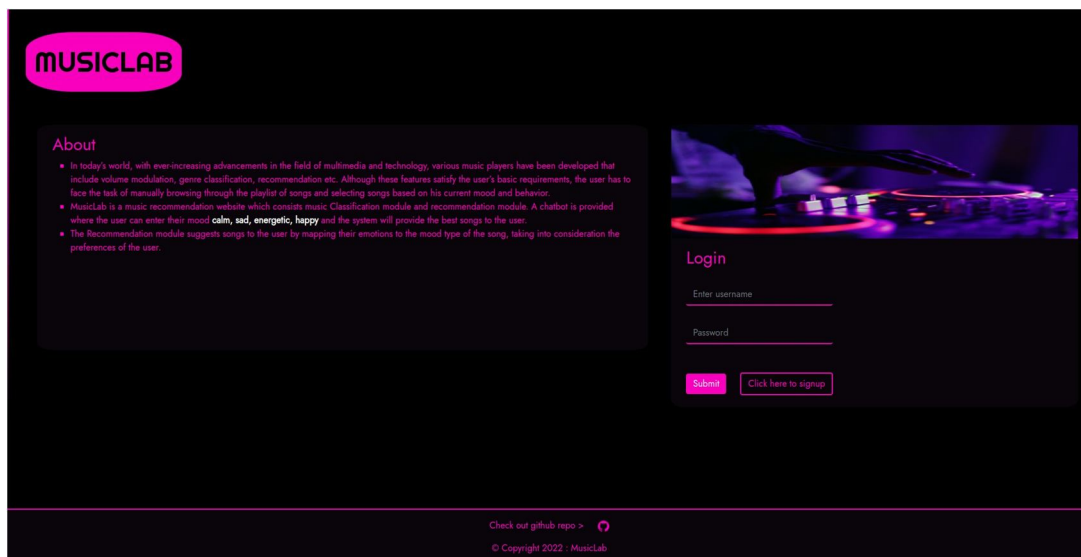


Figure 8.: Login Page

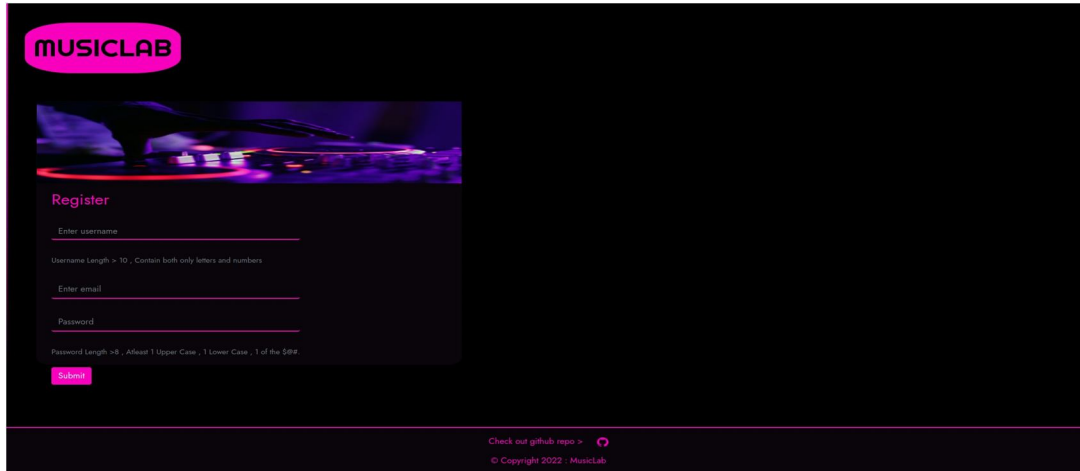


Figure 9.: Registration Page

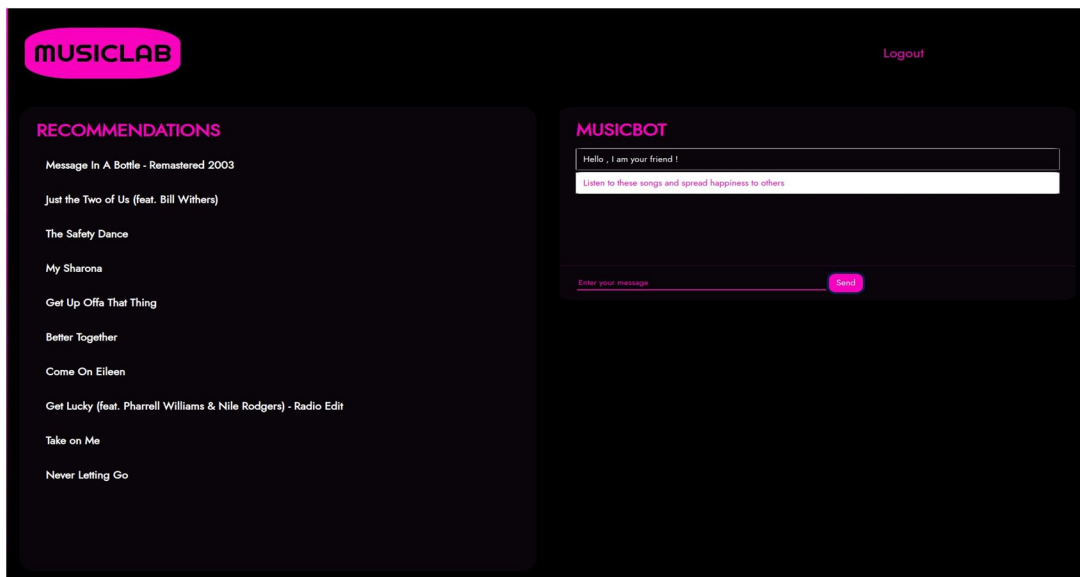


Figure 10.: Recommendation System



Figure 11.: Genre Page

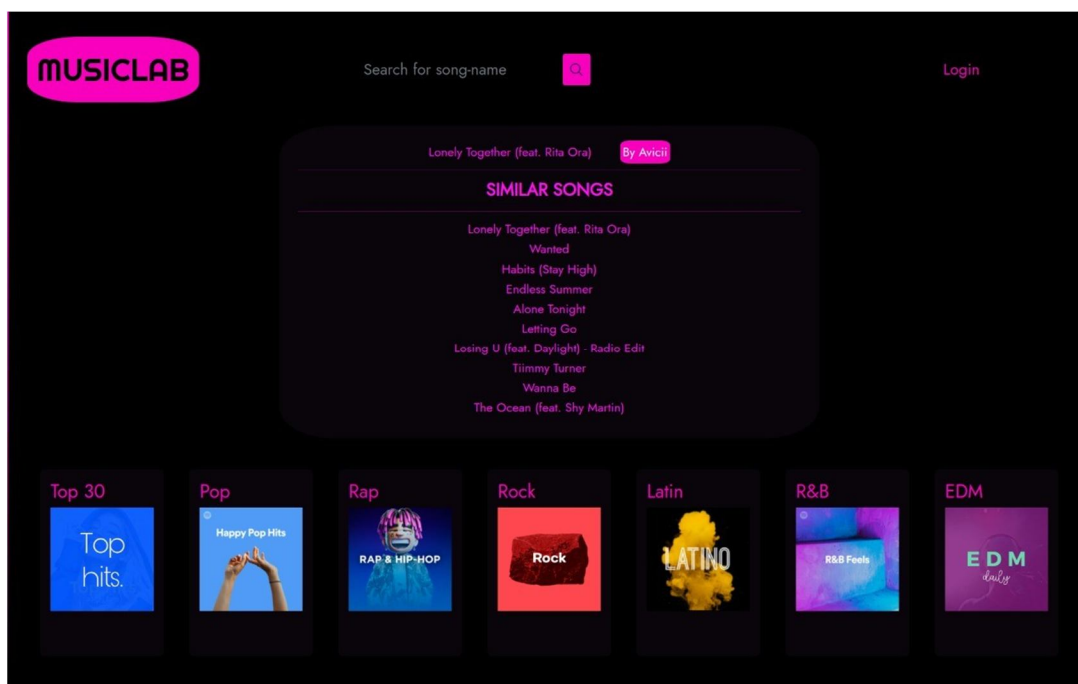


Figure 12.: Search Based Music Recommendation System

REFERENCES

- [1] A. Alrihaili, A. Alsaedi, K. Albalawi and L. Syed, "Music Recommender System for Users Based on Emotion Detection through Facial Features," 2019 12th International Conference on Developments in eSystems Engineering (DeSE), 2019
- [2] Abdul, A.; Chen, J.; Liao, H.-Y.; Chang, S.-H. An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach. Appl. Sci. 2018
- [3] Smt. Namitha S J, 2019, Music Recommendation System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 08, Issue 07 (July2019)
- [4] Raju, Anand R.S, Dr.Sabeenian Gurang, Deepika Kirthika, R Rubeena, Shaik. (2021). AI based Music Recommendation system using Deep Learning Algorithms. IOP Conference Series: Earth and Environmental Science, 2021
- [5] Schedl M, Zamani H, Chen CW, Deldjoo Y, Elahi M. Current challenges and visions in music recommender systems research. Int J Multi Inform Ret. (2018)
- [6] A. V. Iyer, V. Pasad, S. R. Sankhe and K. Prajapati, "Emotion based mood enhancing music recommendation," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), 2017
- [7] Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks , " 2019 The 4th International Conference on Computer Science and Computational Intelligence (ICCCSI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society,2019
- [8] Ivana Andjelkovic, Denis Parra, John O'Donovan, Moodplay: Interactive music recommendation based on Artists' mood similarity, International Journal of Human-Computer Studies, Volume 121, 2019,
- [9] V. Moscato, A. Picariello and G. Sperl'1, "An Emotional Recommender System for Music," in IEEE Intelligent Systems, vol. 36, no. 5, pp. 57-68, 1 Sept.-Oct. 2021, doi: 10.1109/MIS.2020.3026000.
- [10] Samuvel, D. J., Perumal, B., Elangovan, M. (2020). Music recommendation system based on facial emotion recognition. 3C Tecnolog'ia. Glosas de innovaci'on aplicadas a la pyme. Edici'on Especial, Marzo 2020, 261-271. <http://doi.org/10.17993/3ctecno.2020.specialissue4.261-271>



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)