

CHATBOT SONG RECOMMENDER SYSTEM BASED ON EMOTION DETECTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering
with Specialization in Artificial Intelligence

By

BHUMIREDDY LAVANYA (Reg.No - 40731014)

GAJJELA SAI CHARAN (Reg.No - 40731025)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by
AICTE JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BHUMIREDDY LAVANYA** (Reg.No - 40731014) and **GAJJELA SAI CHARAN** (Reg.No - 40731025) who carried out the Project entitled "CHATBOT SONG RECOMMENDER SYSTEM BASED ON EMOTION DETECTION" under my supervision from November 2023 to April 2024.

B. Bala Sai Gayathri 12/04/24

Internal Guide

Ms.B. BALA SAI GAYATHRI, M.E.,

Dr. L. Lakshmanan

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 13-04-2024

Nalini
13/4/24
Internal Examiner

Praveen
13/4/24
External Examiner

DECLARATION

I, **Bhumireddy Lavanya** (Reg.No – 40731014), hereby declare that the Project Report entitled “**CHATBOT SONG RECOMMENDER SYSTEM BASED ON EMOTION DETECTION**” done by me under the guidance of **Ms.B. Bala Sai Gayathri M.E.**, at Sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of **Bachelor of Engineering degree in Computer Science and Engineering with Specialization in Artificial Intelligence**

DATE: 13-04-2024

PLACE: Chennai

B.Lavanya
SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.B.Bala Sai Gayathri M.E.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Music is an integral part of our lives. A song and emotion-based recommendation system permits the users to listen to music based on their emotions. The Chatbot Music Generator by Emotion Detection is a novel system that combines the power of chatbot technology and emotion detection algorithms to generate music tailored to the emotional state of the user. This innovative approach leverages natural language processing techniques and machine learning algorithms to detect the user's emotions from text inputs and translates them into a corresponding musical composition. A chatbot music generator is an innovative tool that uses artificial intelligence to create music based on human emotions. It is designed to analyze text input, such as chat messages, and generate music that reflects the emotional state of the user. The system uses complex algorithms and machine learning techniques to understand the nuances. However, since social media platforms like Facebook and Instagram have a huge influence on the music charts worldwide, users are exposed solely to mainstream music, and recommendations on music streaming platforms are not very personalized. In this era of technological advancement, recommending songs based on a person's emotions is much needed as it will improve users' listening experiences and help them relieve stress by listening to soothing music according to the emotion perceived. In this paper, we have implemented a chatbot that interacts with the user, analyses the emotions of chats, and recommends a song playlist based on the user's emotions. The objective of our application is to identify the emotion perceived by the user, and once the emotion is identified, a list of songs is suggested based on the emotion.

TABLE OF CONTENTS

CHAPTE R NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 Background and Motivation	2
	1.2 Problem Statement	2
	1.3 Objective and Scope	3
2	LITERATURE SURVEY	4
	2.1 Overview Of Existing Research And Literature	4
	2.2 Inferences From Literature Survey	6
	2.3 Open Problems In Existing System	7
3	REQUIREMENT ANALYSIS	10
	3.1 Feasibility Studies of The Project	10
	3.1.1 Economic Feasibility	10
	3.2.2 Technical Feasibility	11
	3.2.3 Social Feasibility	11
	3.2 Requirements Specification	12
	3.2.1 Hardware Requirements	12
	3.2.2 Software Requirements	12
4	DESCRIPTION OF PROPOSED SYSTEM	19
	4.1 Selected Methodology or Process Model	20
	4.1.1 Data Collection and Pre-processing	21
	4.1.2 Evaluation Metrics	22
	4.2 Architecture of Proposed System	22
	4.3 Description of Software for implementation and testing plan of the proposed System	24

4.4	Project Management plan	26
4.5	Finacial Report on Estimated Costing	27
5	IMPLEMENTATION DETAILS	29
5.1	Development and Deployment Setup	29
5.2	Algorithms	30
5.2.1	Convolutional Neural Network(CNN)	30
5.2.2	Long Short-Term Memory(LSTM)	30
5.3	Testing	32
6	OUTCOMES AND DISCUSSION	33
6.1	Potential Outcomes	33
6.2	Discussion Points	33
6.3	Constraints	34
7	RESULTS AND DISCUSSION	35
8	CONCLUSION	37
8.1	Conclusion	37
8.2	Recommendations for future research	38
	REFERENCES	38
	APPENDIX	39
A.	SOURCE CODE	39
B.	SCREEN SHOTS	48
C	RESEARCH PAPER	54

LIST OF FIGURES

CHAPTER NO.	FIGURE NAME	PAGE NO
3.1	Structure of Artificial Neural Network	14
3.2	Structure of Long Short-Term Memory	15
3.3	Structure of Convolutional Neural Network	16
4.1	LSTM Model development flow chart process	20
4.2	Architecture Diagram	22
4.3	Flow Diagram	25
7.1	Plot of no. of songs vs Decade	36
7.2	Songs generated for sad mood	36
B.1	Dataset Used	46
B.2	Subplot of decade and count	46
B.3	Univariate Analysis	47
B.4	Univariate Analysis for numerical data	47
B.5	Bivariate Analysis	48
B.6	Boxplot	48
B.7	User interacting website	49
B.8	Entering the emotion	49
B.9	Song Generation	50
B.10	Entering alternate emotion	50
B.11	Song Generation for happy mood	51

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
API	Application Programming Interface
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Natural language processing is the ability of a computer program to understand human language and has a lot of applications in today's world. The most popular application is the recognition of emotions from text. Emotion Detection and Recognition from Text analysis is a recent field of research that is closely associated with sentiment analysis. Sentiment analysis is a branch of Natural Language Processing (NLP) that seeks to detect positive, neutral, or negative emotions in text. It is dedicated to the exploration of subjective opinions or feelings collected from various sources about a particular subject, whereas Emotion Analysis is a process of detecting and recognizing different types of feelings through the expression of texts, such as anger, fear, disgust, happiness, sadness, and surprise. Detecting a person's emotions is a difficult task but detecting emotions using text written by a person is even more difficult as a person can express their emotions in any form.

The context-dependence of emotions inside the text is one of the most difficult aspects of determining emotion. For example, consider the phrase "Shut up!" It has an element of anger without using the word "anger" or any of its counterparts. Emotion detection is a process in project. Recognizing these emotions from a text plays a vital role in our application. In our project, the process of emotion detection of the user is done with supervised machine learning algorithms, namely Support Vector Machine (SVM), Linear Support Vector Machine (LSVM), Random Forest, and Decision Tree. The proposed model will detect emotions What the user needs The Chatbot module of the application makes use of Deep Learning techniques for its implementation. We created a simple, retrieval-based chatbot that employs pre-programmed input patterns and responses. To classify the category of a user's message and obtain an appropriate response from the chatbot, we utilized a special recurrent neural network called Long Short-Term Memory.

1.1 BACKGROUND AND MOTIVATION

Music recommendation systems are a dime a dozen these days. They typically use listening history or user preferences to suggest new songs. However, these methods don't always capture the user's current emotional state. This is where chatbot song recommender systems with emotion detection come in. This is a relatively new approach that leverages the power of chatbots to understand a user's current mood through conversation. Music is a deeply personal experience. By understanding a user's emotions, the system can recommend songs that resonate with their current mood, creating a more personalized and enriching listening experience. Struggling to pick the right song can be frustrating. Emotion detection helps the system take the guesswork out of music selection, allowing users to find the perfect soundtrack for their current state of mind. Music can be a powerful tool for managing emotions. The recommender system can suggest uplifting songs when the user is feeling down or calming melodies during stressful times.

1.2 PROBLEM STATEMENT

Traditional music recommender systems often fail to capture the user's current emotional state. This leads to Users receive songs that don't match their current mood, leading to frustration and a disconnect with the system The inability to account for emotions hinders the creation of a truly personalized music experts. Users waste time browsing through irrelevant music, leading to decision fatigue. This problem is particularly relevant for users who actively seek music to match their current mood. Struggle to choose songs that reflect their emotional state. Desire a more personalized and emotionally driven music listening experience.

By developing a chatbot song recommender system with emotion detection, we can address these issues and create a system that provides Songs that resonate with the user's current mood. Recommendations tailored to user's unique emotional state. Reduced time spent searching for the right music.

1.3 OBJECTIVE AND SCOPE

Objective

Chatbot song recommender systems with emotion detection aim to revolutionize music selection. By analyzing a user's emotional state through conversation, these systems move beyond generic recommendations and tailor suggestions to the user's current mood. This personalized approach tackles decision fatigue and ensures users find music that truly resonates with them. Ultimately, the objective is to create a seamless and emotionally intelligent music experience, allowing users to leverage music for both enjoyment and emotional well-being.

Scope

A Chatbot Song Recommender System based on emotion detection holds significant potential for enhancing user experience in music streaming platforms. By integrating sentiment analysis algorithms, the system can assess users' emotional states and recommend songs that align with their mood, preferences, and activities. This innovative approach not only personalizes the music listening experience but also enables users to discover new tracks tailored to their current emotional needs. The scope of this project includes implementing machine learning models for emotion detection, integrating a user-friendly chatbot interface, and incorporating a vast music database for recommendations. Through natural language processing techniques, the chatbot will engage users in interactive conversations to understand their emotional context and provide relevant song suggestions. Additionally, the system will continuously learn and improve its recommendations based on user feedback and interaction patterns. The project aims to enhance user engagement, satisfaction, and retention within music streaming platforms by delivering a personalized and emotionally resonant music discovery experience.

CHAPTER 2

LITERATURE SURVEY

Several studies on chatbot song recommender system based on emotion detection have been conducted in recent years. This section summarizes all the previous work on facial expression recognition.

2.1 OVERVIEW OF EXISTING RESEARCH AND LITERATURE

Nikhil et al. [1] use algorithms and technologies which include Haar cascade, Canny edge, Blob detection for the process of emotion detection. The system captures pictures of the user and according to that mood gets detected. Inputs like face and emotions are taken from the picture, and the system also provides a chat box to give responses. The proposed system in the paper presents a new approach for building desktop applications for chat bots using text and gestures. The system can make a conversation through the chatting application. The system will send some links, web pages or information depending on the response from the user. The system detects smiles and stress. When a smile is detected by the system, jokes pop-ups will be shown on the screen, and when stress is detected, inspirational quotes pop-ups will be shown on the screen. Also, happy songs are played when a smile is detected. And similarly, inspirational songs are played when stress is detected.

Jae Sik Lee et al. [2] have used the concept of context reasoning wherein the context data is utilized to understand the user's situation. They propose a music recommendation system that comprises the ability of context reasoning in this paper. Their proposed system contains modules such as Intention Module, Mood Module and Recommendation Module, each of which provide a unique functionality to the system and play a vital role for the system's performance. Context reasoning is done by the Intention Module with the help of environmental context data and concludes whether the user is interested in listening to music or not. Next, the type of music that is deemed to be most appropriate to the user's context is determined by the Mood Module. Lastly, the

music is recommended to the user by the Recommendation Module.

Renuka R. Londhe et al. in [3] have studied the concept of recognizing facial expressions by considering the various properties that are associated with a person's face. Whenever there is a change in the facial expression, changes can be noticed in the curvatures on the face as well as features of the face such as nose, lips, eyebrows, and mouth area. And accordingly, there will be changes in the intensity of the corresponding pixels of the images. These features are then classified into six expressions which include anger, disgust, fear, happiness, sadness and surprise with the help of artificial neural network. The Scaled Conjugate Gradient back-propagation algorithm is used to train and test the two layered feed forward neural network. They acquired a 92.2 % recognition rate. Here, they have made use of the JAFFE database which consists of seven expressions for analysis through the computer.

Dolly Reney et al. in their paper [4] address the importance of face and emotion identification in the field of security and how it helps give solutions to the different challenges faced. Database plays a major role when comparing the facial attributes and sound Mel frequency components, when it comes to whichever face and emotion identification system. The database is created for which facial characteristics are computed and these are then stored in the database. Various algorithms are used in order to analyze the face and emotion with the help of the aforementioned database. The implementation of the process of recognizing the person's face and the emotion being expressed by him uses an effective method for the creation of a database comprising the facial expressions and emotion. They have used the Viola-Jones algorithm for the face identification process and the face and emotion identification is evaluated by the KNN classifier.

Ai Thanh Ho et al. in their paper [5], introduce an Emotion-based Movie Recommender System (E-MRS) which is intended to solve the problem that the conventional system of user profile does not take into consideration how important user's emotions are and how they affect user's choices, which the recommender systems are unable to understand and capture the constantly

changing preferences of user. According to the paper, the objective of EMRS is to give the users a list of suggestions that are customized using a combination of collaborative filtering and content-based techniques. Here the user's emotions as well as his preferences are considered when providing a recommendation, also other similar user opinions are considered. The design of the proposed system, its implementation along with its evaluation procedure is also discussed. In order to relate emotions to movies, the users have to answer a questionnaire about what movies or which categories of movies they liked to watch according to each emotion. Furthermore, the system captures user emotions by asking them to use 3 colors to decorate their avatar.

2.2 INFERENCES FROM LITERATURE SURVEY

After going through the previous works, we inferred some points. Nikhil et al paper, their system has a chatbot that responds to user emotions, sending links or web pages. It detects smiles, jokes and playing happy songs, and detects stress, showing inspirational quotes and playing motivational songs. Jae Sik Lee paper, they use context reasoning to understand the user's situation and propose a music recommendation system with three modules: Intention, Mood, and Recommendation.

Renuka R. Londhe paper, they analyzed facial expressions by examining changes in face features like nose, lips, and eyebrows. Used artificial neural networks to classify six emotions: anger, disgust, fear, happiness, sadness, and surprise. Dolly Reney paper, Face and emotion identification in security is vital. Databases store facial attributes and sound data. Algorithms analyze faces and emotions. Creating databases of facial expressions and emotions is effective. Ai Thanh Ho paper, Introducing Emotion-based Movie Recommender System (E-MRS), it considers user emotions and preferences for personalized movie recommendations. Uses collaborative filtering and content-based techniques.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

A feasibility study for a chatbot song recommender system based on emotion detection involves several crucial considerations. First and foremost, it's essential to define the project's scope and objectives. Determine the specific functions and goals of the chatbot, such as identifying and recommending songs based on users' emotional states. From a technical perspective, assess the feasibility of the project. Examine data sources for emotion detection, choose the appropriate chatbot development platform, and decide on the method for detecting emotions. Additionally, consider the availability of a comprehensive song database for recommendations. Ensure data availability is feasible. This includes assessing whether the necessary emotional data for training and testing your models is readily accessible, and if not, understanding the effort and cost of data collection and processing. Consider user adoption and acceptance. Assess whether users are likely to embrace an emotion-based song recommender chatbot by conducting surveys or focus groups to gather feedback. There are several existing systems that incorporate chatbot technology and music generation to create personalized music based on user input. However, these systems often face certain drawbacks that limit their effectiveness and user experience.

Creating a chatbot song recommender system based on emotion detection is an exciting and evolving field. There are several open problems and challenges that exist within this domain, including:

1. **Emotion Detection Accuracy:** Improving the accuracy of emotion detection is an ongoing challenge. Emotions are complex and often nuanced, and current methods may struggle with accurately detecting subtle emotional states.
2. **Multi-Modal Emotion Detection:** Integrating multiple data sources (text, voice, facial expressions) for a more comprehensive understanding of a user's emotional state is a challenging problem. Combining these modalities effectively and accurately is an open research area.
3. **Context Understanding:** Enhancing the system's ability to understand the context of a conversation is crucial. Understanding the conversation's history

and context can improve song recommendations based on emotional state.

4. **Personalization and Diversity:** Developing methods to recommend songs that are not only emotionally appropriate but also personalized to individual preferences is a significant challenge. Balancing recommendations that fit the user's mood and introducing diversity can be tricky.

5. **Real-time Emotion Tracking:** Most emotion detection methods work on static input data, but in a real-time chat, emotions can change rapidly. Designing systems that can track and respond to changing emotions in real-time is an open problem.

6. **Data Privacy and Ethics:** Ensuring the ethical use of user data, especially when it comes to emotion detection, is a growing concern. Developing methods for privacy-preserving emotion detection is essential.

7. **Cross-Cultural and Multilingual Emotion Detection:** Emotions and music preferences can vary significantly across cultures and languages. Extending emotion detection and recommendation systems to be culturally and linguistically sensitive is a challenge.

8. **Limited Labeled Data:** Training emotion detection models typically requires a large amount of labeled data, and obtaining such data for specific domains or languages can be challenging. Developing techniques for training models with limited data is an open problem.

9. **Evaluation Metrics:** Defining appropriate evaluation metrics for chatbot song recommender systems is an ongoing challenge. Existing metrics may not capture the user's satisfaction accurately.

10. **User Feedback Integration:** Designing systems that can learn and adapt based on user feedback is an open problem. Users' feedback can be valuable for improving the system's recommendations.

11. **Handling Mixed Emotions:** Users often experience mixed emotions, and current systems may struggle to recommend appropriate content for such situations. Developing algorithms to handle mixed emotional states is a challenge.

12. **Explainability and Transparency:** Ensuring that users can understand why a particular song was recommended based on their emotional state is important

for user trust. Developing transparent and explainable recommendation algorithms is a current area of research.

13. Robustness and Security: Ensuring that the system is robust against adversarial attacks or manipulation is crucial. Security issues related to emotion detection, data integrity, and user privacy need to be addressed.

14. Integration with Music Licensing and Copyright: Handling music recommendations in a legal and ethical manner, respecting copyright and licensing agreements, is an open problem in the deployment of chatbot song recommender systems.

Solving these open problems will contribute to the development of more effective and user-friendly chatbot song recommender systems based on emotion detection. Researchers and developers in the field should address these challenges to create systems that provide meaningful and enjoyable user experiences.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES OF THE PROJECT

A feasibility study for a chatbot song recommender system based on emotion detection involves several crucial considerations. First and foremost, it's essential to define the project's scope and objectives. Determine the specific functions and goals of the chatbot, such as identifying and recommending songs based on users' emotional states. From a technical perspective, assess the feasibility of the project. Examine data sources for emotion detection, choose the appropriate chatbot development platform, and decide on the method for detecting emotions. Additionally, consider the availability of a comprehensive song database for recommendations. Ensure data availability is feasible. This includes assessing whether the necessary emotional data for training and testing your models is readily accessible, and if not, understanding the effort and cost of data collection and processing. Consider user adoption and acceptance. Assess whether users are likely to embrace an emotion-based song recommender chatbot by conducting surveys or focus groups to gather feedback. In this case three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

3.1.1 Economic Feasibility

- **Cost of Development:** The development cost will involve expenses related to hiring developers skilled in natural language processing (NLP), machine learning, and emotion detection algorithms. Additionally, there may be costs associated with acquiring or licensing emotion detection APIs or datasets.

3.1.2 Technical Feasibility

- **Emotion Detection Algorithms:** Assess the availability and accuracy of emotion detection algorithms suitable for processing text input from users. Consider factors such as language nuances and cultural differences that may affect the performance of these algorithms.
- **Integration with Music Databases:** Evaluate the technical feasibility of integrating the chatbot with music databases to access a wide range of songs and metadata. This may involve API integration with music streaming platforms or building a custom database.
- **Scalability and Performance:** Ensure that the system can handle a potentially large user base and provide real-time recommendations based on detected emotions. Scalability considerations should be made for both the emotion detection module and the recommendation engine.

3.1.3 Social Feasibility

- **User Acceptance:** Gauge user acceptance through surveys, focus groups, or beta testing. Understand users' attitudes towards sharing emotional states with a chatbot and receiving song recommendations based on those emotions.
- **Privacy and Ethical Considerations:** Address privacy concerns related to the collection and use of emotional data. Implement robust privacy policies, data encryption measures, and user consent mechanisms to protect user privacy and build trust.
- **Cultural Sensitivity:** Ensure that the chatbot's recommendations are culturally sensitive and avoid reinforcing stereotypes or biases related to emotions and music preferences.
- **User Experience (UX):** Design the chatbot interface to provide a seamless and engaging user experience. Consider factors such as language tone, interface design, and feedback mechanisms to enhance user satisfaction

3.2 REQUIREMENTS SPECIFICATION

The Hardware and Software Requirements are described

3.2.1 Hardware Requirements

- RAM : 4GB or above 4GB
- Processor of Frequency : 1.5GHz or above
- Processor : Intel Pentium 4 or higher

3.2.2 Software Requirements

- Operating System : Windows 8 and above
- Languages : Python
- Tools Used : Jupiter Notebook, Visual Studio

Python

Python is a scripting language that is high-level, interpreted, interactive, and object-oriented. Python is intended to be extremely readable. It commonly employs English terms rather than punctuation, and it has fewer syntactical structures than other languages.

Python Features

- **Easy-to-learn** - Python has a small number of keywords, a basic structure, and a well-defined syntax. This enables the pupil to swiftly learn the language.
- **Easy-to-read** - Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** - Python's source code is relatively simple to maintain.
- **A broad standard library** - The majority of Python's library is portable

and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** - Python features an interactive mode that enables for interactive testing and debugging of code snippets.
- **Portable** - Python can operate on a broad range of hardware devices and has the same interface across them all.
- **Extendable** - The Python interpreter may be extended using low-level modules. These modules allow programmers to enhance or adapt their tools to make them more efficient.
- **Databases** - Python provides interfaces to all major commercial databases.
- **GUI Programming** - Python can construct and port GUI programmes to numerous system calls, libraries, and windows systems, including Windows MFC, Macintosh, and Unix's X Window system.
- **Scalable** - Python provides a better structure and support for large programs than shell scripting.
- **Libraries:** Python contains vast source of premodule libraries with code which helps to perform a task in less amount of time by importing the libraries.
- **User Friendly:** Python syntax is so easy to use and can learn in short period of time

Deep Learning

Deep learning is a type of machine learning that uses algorithms meant to function in a manner similar to the human brain. While the original goal for AI was broadly to make machines able to do things that would otherwise require human intelligence, the idea has been refined in the decades since. François Chollet, AI researcher at Google and creator of the machine learning software library Keras, says: “Intelligence is not a skill in itself, it’s not about what you can do, but how well and how efficiently you can learn new things.

Artificial Neural Network

Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions. The term "Artificial neural network" refers to a biologically inspired sub- field of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

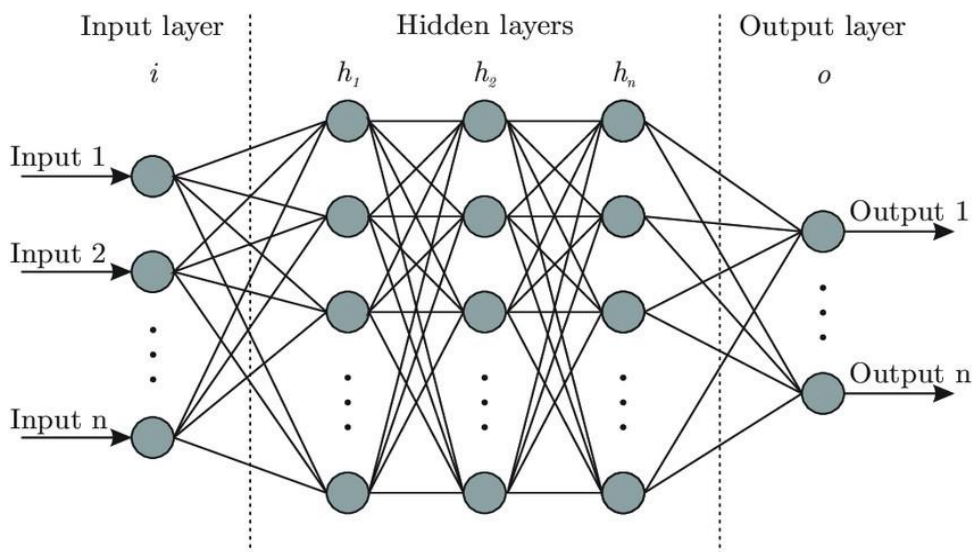


Fig. 3.1: Structure of Artificial Neural Network

Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field. In this post, you will get insight into LSTMs using the words of research scientists that developed the methods and applied them to new and important problems. There are few that are better at clearly and precisely articulating both the promise of LSTMs and how they work than the experts that developed them.

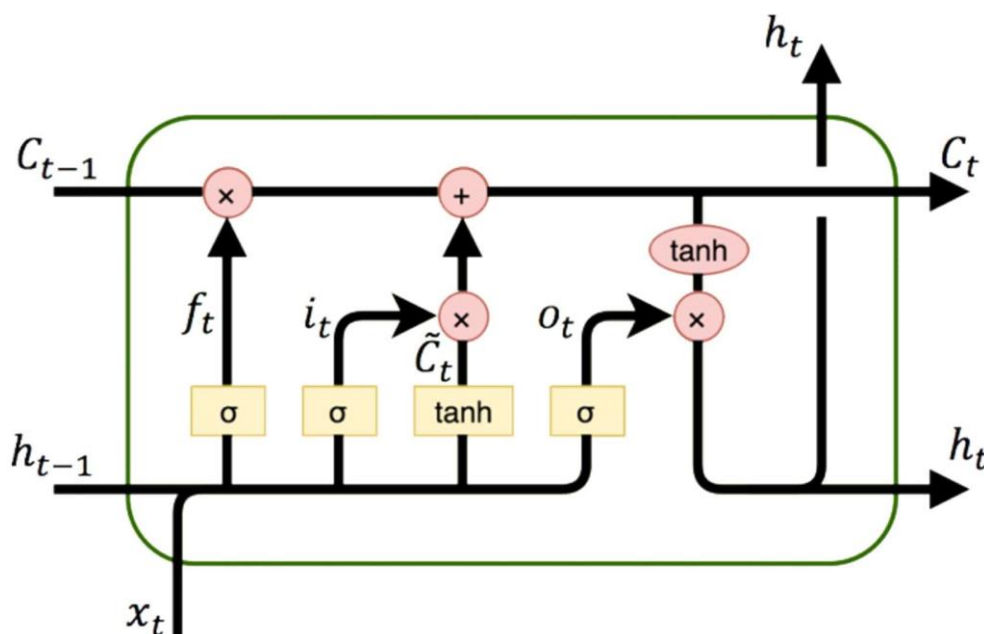


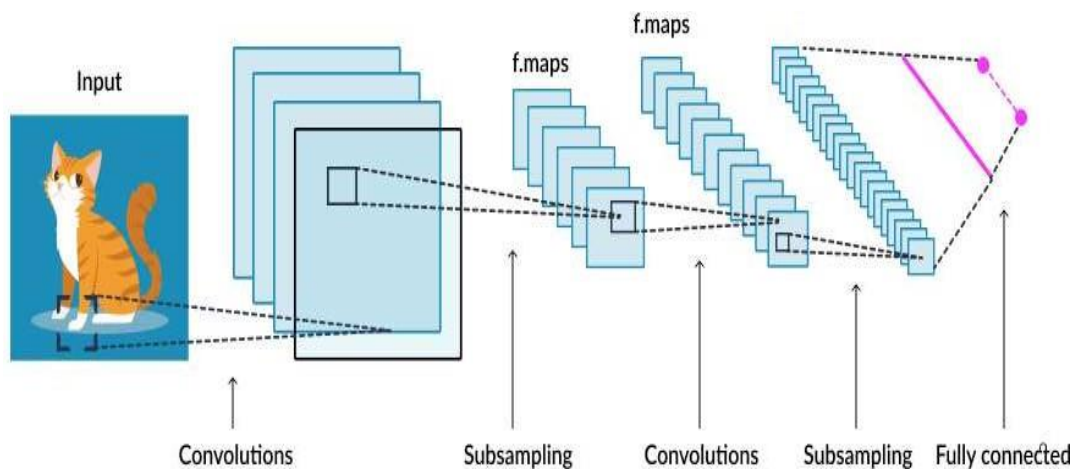
Fig. 3.2: Structure of Long Short-Term Memory

Convolutional Neural Network

A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals. CNN are

very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces. This characteristic that makes convolutional neural network so robust for computer vision. CNN can run directly on a underdone image and do not need any preprocessing. A convolutional neural network is a feed forward neural network, seldom with up to 20. The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer. CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes. With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces. The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

Fig: 3.3: Structure of Convolutional Neural Network



Tone Analyzer

An evaluation tool Language, emotional, and social tones are the three forms of conversation tones. There are five distinct emotional tones: anger, disgust, fear, pleasure, and sadness. There are three exclusive linguistic tones: analytical, confident, and tentative. There are five predominant social traits (big five personality traits): openness, conscientiousness, extraversion, agreeableness, and emotional variety. What sort of learning has taken place? the use of system studying, a version created to expect the tone of recent texts primarily based on conversations. system gaining knowledge of the use of the help Vector system (SVM). around 30% of samples are related to more than one tone, so they decided no longer to apply multi class category, but as a substitute multi-label classification. The model becomes educated independently for every tone using the one-vs-relaxation paradigm. the very last tones have been decided by way of figuring out the ones expected .

IBM Emotional API

Frontend is going to show us how the UI seems like however every time person talk with chatbot it will ultimately displays on the frontend. And the server aspect there are specially Api's would be operating to discover the emotion of the textual content of person through the usage of IBM Emotional API and after the detection of the emotion it'll suggest the music via last.FM API. conversation is initiated by means of the person The IBM Emotional API is used to research the conversation's emotional content

Keras

TensorFlow is an open-source library for numerical computation and machine learning developed by Google. It is designed to simplify the process of building, training, and deploying machine learning models by providing a high-level API for building neural networks, as well as low-level APIs for more advanced users. TensorFlow supports a wide range of platforms, from desktops to

clusters of GPUs and TPUs, and can be used for a variety of tasks, including image and speech recognition, natural language processing, and recommendation systems. It is widely used in industry and academia for research and production applications.

Keras acts as a high-level interface on top of powerful deep learning frameworks like TensorFlow, PyTorch, or JAX. This simplifies the process of building complex neural networks by providing a user-friendly and consistent API. Keras is built with modular components that can be easily combined to create complex models. It also integrates well with other libraries, making it a versatile tool for deep learning projects.

Advantages of Keras

Keras has several advantages that make it a popular choice for deep learning.

- **Simplicity** - Keras is very easy and simple to use. It is a user-friendly API with easy-to-learn and code features.
- **Backend support** - Keras does not operate with low-level computations. So, it supports the use of backends.
- **Pre-trained models** - Keras provides numerous pre-trained models.
- **Fast experimentation** - Keras is built to simplify the tasks of users.
- **Great community and calibre documentation** - Keras has a large supportive community.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The proposed system combines all three existing systems and integrates them into a single system that recommends songs to users based on the emotional tone of their conversation with the chatbot. After the current song has finished playing, similar songs will be played, or the user can play the next song. This system developed is a blend of a emotional based Chatbot, ML algorithms to detect the emotional tone in the text and the Spotify music player, the system has a major implementation of the API functionality in these modules, API simplifies our execution process and lets us focus more on the functionality and simpler system design is obtained.

A simple webpage is developed for user interaction and a Spotify web player which displays the recommended songs.

Song recommendation systems are used by various music providing platforms such as Spotify, SoundCloud, YouTube, iTunes, and others to recommend songs based on previous songs, top tracks and artists played by the user.

These platforms use various algorithms to recommend songs.

The first thing one notices on Spotify is the home screen, which is where the recommendations begin, and the home screen is controlled by an AI system known as Bandits for Recommendations. For example, Spotify's algorithm is always finding new ways to understand the kind of music one listens to, from the songs that are always on repeat to the favorite genre that one can't get enough of. Bart is the reason why Spotify satisfies their users with a good Spotify playlist.

Soundcloud is another example, where suggested tracks are chosen by an algorithm that gives suggestions through a network of SoundCloud relations and interactions (for example a user liked a track, a user followed another user, a track is reposted etc.).

Track recommendations can be related to tracks in this network both directly and indirectly.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

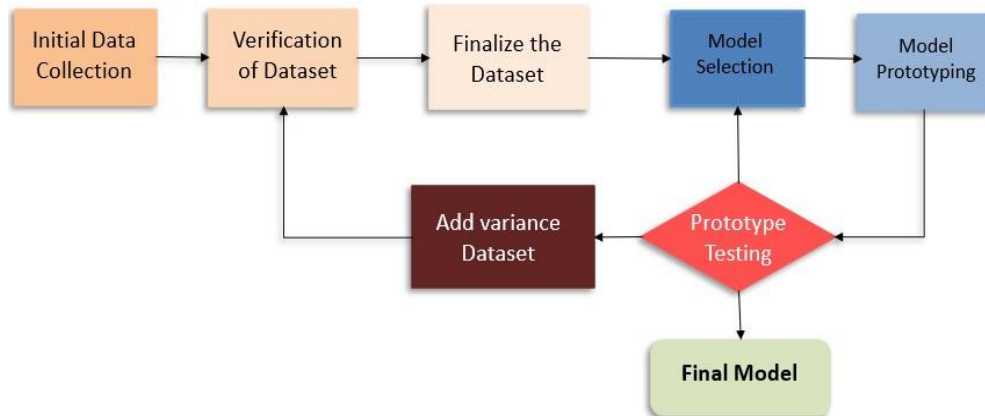


Fig: 4.1: LSTM model development flow process

4.1.1 Data Collection and Pre-processing

Developing an LSTM (Long Short-Term Memory) model for a Chatbot Song Recommender System based on emotion detection is a complex process. Here's a high-level flow process of how you might go about it:

1. Data Collection and Preprocessing:

- Gather a diverse dataset of text inputs that represent different user emotions. These could be user messages or descriptions of emotions.
- Clean and preprocess the text data, including tokenization and removing any irrelevant information.

2. Labeling Emotions:

- Annotate the data with emotion labels, such as happy, sad, excited, etc. These labels will be used to train the model to recognize emotion.

3. Embedding Layer:

- Create word embeddings to represent words in a numerical format. Word embeddings capture the semantic meaning of words and are used as inputs to the LSTM model.

4. LSTM Model Architecture:

- Design the LSTM model architecture. The input layer will consist of word embeddings.
- Configure the LSTM layers. You can have multiple LSTM layers to capture sequential information effectively.
- The output layer should be designed to predict the emotional class (e.g., happy, sad, etc.).

5. Training:

- Split your labeled dataset into training, validation, and testing sets.
- Train the LSTM model using the training data. Use loss functions and optimization algorithms suitable for emotion classification, such as categorical cross-entropy and Adam optimizer.
- Monitor the model's performance on the validation set and implement early stopping to prevent overfitting.

6. Evaluation:

- Evaluate the LSTM model's performance using the test dataset. Common evaluation metrics include accuracy, precision, recall, and F1-score.
- Analyze the confusion matrix to understand where the model may be misclassifying emotions.

7. Integration with Chatbot:

- Once your LSTM model is trained and evaluated successfully, integrate it into the chatbot module. This model will be responsible for detecting the user's emotions during the conversation.

8. Real-time Emotion Detection:

- Configure the chatbot to perform real-time emotion detection as the user interacts with it. This may involve processing and classifying user messages as they come in.

9. Song Recommendation Integration:

- After determining the user's emotion, connect the LSTM model to the song recommendation system. The emotion detected will be used to suggest relevant songs from the database.

10. Testing and User Feedback:

- Thoroughly test the integrated chatbot and song recommender system. Ensure that it provides accurate recommendations based on detected emotions.

- Collect user feedback to make iterative improvements.

4.1.2 *Evaluation Metrics*

The accuracy of the models is evaluated using various evaluation metrics such as accuracy, precision, recall, and F1-score. The accuracy of the models is the proportion of correctly classified physical activities. The precision is the ratio of correctly classified physical activities to the total number of physical activities classified as positive. The recall is the ratio of correctly classified physical activities to the total number of actual positive physical activities. The F1-score is the harmonic mean of precision and recall.

4.2 ARCHITECTURE OF PROPOSED SYSTEM

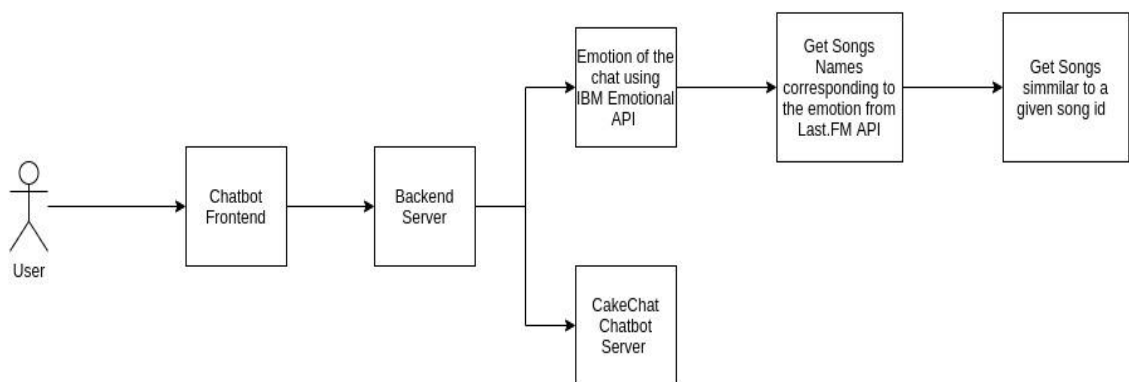


Fig. 4.2: Architecture Diagram

The figure totally showed us how this project would look like. Basically, frontend is going to show us how the Ui look like but whenever user talk with chatbot it will eventually reflects on the frontend. AND the server side there are mainly two Api's would be working to detect the emotion of the text of user by using IBM Emotional API and after the detection of the emotion it will recommend the song through Last.FM API. Conversation is initiated by the user The IBM Emotional API is used to analyze the conversation's emotional content The Cake chat Chatbot will reply to your conversation by utilizing the Last.fm songs API, the app retrieves the top songs based on the Emotion it perceives After listening to a song for some time, a similar song will be recommended to the user using the Last. Fm. API.

Algorithms: -

SVD (Singular Value Decomposition) Algorithm:

SVD could also be a matrix factorization technique that is typically wont to diminish the quantity of feature of a dataset by reducing the matrix from N space to K space where $K < N$. For the point of the guidance framework be that as it may, we are just interested about by the matrix factorization part keeping same dimensional. The matrix factorization is completed on the user item ratings matrix built. Each item is often represented with a q vector. Similarly, each client are often represented by a p vector such the inner product of these 2 vectors is the expected rating

Nearest Neighborhood Model:

The Nearest Neighborhood model involves collecting data from numerous clients at that point making forecasts according to the similarity measures between users and between items. This might be grouped into client-based and item-based models. In item-based model, it's expected that songs that are frequently listened to together by certain clients will in general be indistinguishable and are bound to be listened together in future additionally by another client. According to client-based similarity model, clients who have similar listening histories, i.e., have listened in to similar songs inside the past

will in general have comparative interests and may most likely hear similar songs in future as well.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED SYSTEM

We present a brief description of the software components and tools that will be used in the design, development, and evaluation of the proposed Chatbot Music Recommendation System with Emotion Detection model in this section. The software stack is required for the development of the system, deployment, and validation.

A programming language and framework:

Python (Version X.X):

Python will be the major programming language used throughout the project. It provides an extensive ecosystem of libraries and frameworks.

Keras (Version X.X) with TensorFlow (Version X.X):

TensorFlow and Keras will be used to create and train machine learning models, which will include emotion detection and recommendation algorithms.

Libraries for Natural Language Processing:

Text processing and word embeddings are performed using NLTK, spaCy, and Gensim.

Framework for the web:

[Flask/Django] (Select one): We will utilise [Flask/Django] to create the chatbot system's web interface.

Database Management:

Relational Database Management MySQL (Version X.X):

MySQL will be utilised to store user profile, chat, and music recommendation

system. **MongoDB (Version X.X) NoSQL Database**

MongoDB will be used as a backup data storage for unstructured data and chat logs.

Development Environment IDE (Integrated Development Environment):

Visual Studio Code:

The visual studio code IDE for coding, debugging, and version management will aid in project development.

Version Control and Collaboration Version Control:

Git (Version X.X) :

Git will be used for tracking changes in the source code and collaborative development.

Repository Hosting:

GitHub (Or GitLab) The project repository will be hosted on GitHub (or GitLab) for team collaboration and version control.

Testing and Quality Assurance Unit Testing Framework:

PyTest (Version X.X)

PyTest will be employed for testing individual components and functions within the system.

Continuous Integration (CI):

We will set up Jenkins for automating the integration and testing of the system during development.

Additional Tools Emotion Detection API:

We integrate an emotion detection API like Microsoft Azure Emotion API/Google Cloud Natural Language API for emotion analysis.

Music APIs:

Integration with music APIs like Spotify for real-time music recommendations.

4.4 PROJECT MANAGEMENT PLAN

Article IEEE Std. 1058-1998 [23] IEEE Standard for Software Project Management Plans, IEEE Std. 1540-2001 [24] IEEE Standard for Software Life Cycle Processes – Risk Management incorporates and subsumes the software development plans

- **Project Planning and Management Module** - This module will include the overall planning and management of the project, including setting project goals and timelines, creating and assigning tasks, tracking progress, and communicating with team members.
- **Data Collection and Pre-processing Module** – emotion detection through natural language processing dataset collection for this project is implemented and the emotion of an user has been detected with the input from chatbot given by user.
- **ConvLSTM Model Module** - This module will implement the ConvLSTM model for song generation and emotion detection. It will include designing the model architecture, training the model on the preprocessed data, and evaluating the performance of the model on a validation dataset.
- **Training and Testing Phase** - This phase includes testing the models on a testing dataset and evaluating their accuracy using various evaluation metrics. The testing phase is expected to take two weeks.

The project management plan with modules for chatbot song recommender system will allow for a well-organized and structured approach to developing the system, ensuring that each component is carefully designed, implemented, and tested before integrating them into the final product. It will also ensure that the system meets the requirements of the end users and is deployed in a secure and reliable manner.

4.5 FINANCIAL REPORT ON ESTIMATED COSTING

Deep Learning techniques such as LSTM and CNN has greatly improved the accuracy of systems. However, developing a HAR system using LSTM and CNN can be a complex and resource-intensive process that requires careful planning and budgeting. In this financial report, we will provide an estimate of the cost of developing a HAR system using LSTM and CNN

- **Cost Estimate - Hardware and Software Costs:** The hardware and software cost for developing a HAR system using LSTM can be significant. This includes the cost of purchasing and maintaining high-performance GPUs, cloud computing services, and specialized software such as TensorFlow or Keras.
- **Data Acquisition and Pre-processing Costs -** Collecting, annotating, and preprocessing the data required for training and testing the LSTM models can be a significant expense. Additionally, the cost of storing and managing large datasets can be significant.
- **Model Development Costs -** Developing and optimizing the LSTM models can be a time-consuming and resource-intensive process. This may involve hiring machine learning experts or outsourcing the development work to a third-party provider. Additionally, the cost of testing and validating the

models can be significant, as this requires access to large datasets and specialized software tools.

- **Deployment and Maintenance Costs** - Once the models have been developed, they need to be deployed and integrated into the target system. This may involve additional costs for server infrastructure, API development, and ongoing maintenance and support. The cost of ongoing maintenance and support can be significant, especially if the system is deployed in a complex environment with multiple users or if the models need to be updated frequently.

Developing a chatbot song recommender system using LSTM can be a complex and resource-intensive process that requires planning and budgeting. The cost of developing a system can vary greatly depending on the size and complexity of the system, the data sources and size, the programming languages and tools used, and the expertise of the developers. Therefore, it is important to carefully assess the requirements of the project and work with experienced developers and data scientists to ensure the success of the project.

CHAPTER 5

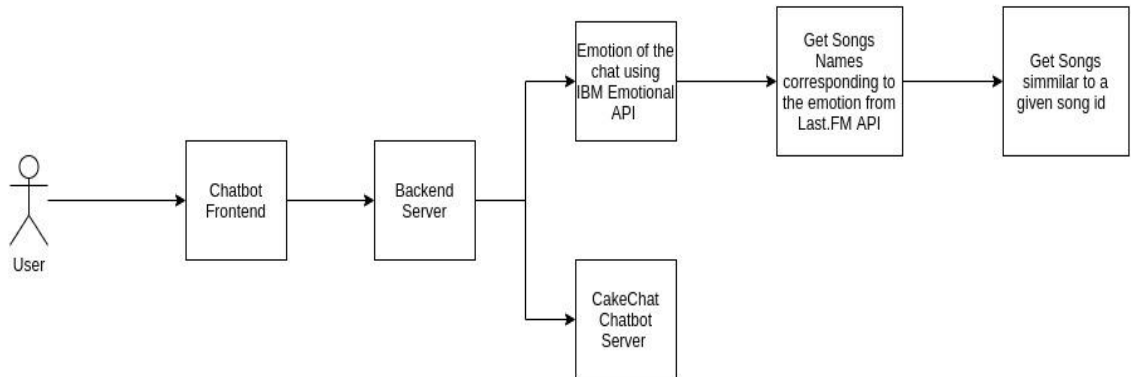
IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

To develop and deploy a chatbot song recommender system based on emotion detection system using Long Short-Term Memory (LSTM), you can follow the steps below:

- **Data Collection and Pre-processing** - Collect data from various sources and pre-process it to make it suitable for machine learning. You can use various techniques for data pre-processing, such as normalization, data augmentation, and feature extraction.
- **Model Architecture Design** - Design the architecture of the LSTM model. This architecture should include convolutional layers for feature extraction from the input data, LSTM layers for modeling time-series data, and fully connected layers for classification.
- **Model Training** - Train the LSTM model using the preprocessed data. You can use various optimization algorithms, such as Adam, RMSprop, or SGD, to minimize the loss function.
- **Model Evaluation** - Evaluate the performance of the trained model using various metrics, such as accuracy, precision, recall, and F1 score. You can also use visualization techniques, such as confusion matrices or ROC curves, to analyze the performance of the model.
- **Deployment** - Deploy the model in a production environment. You can use various frameworks, such as TensorFlow or PyTorch, to create a deployable model. You can also use cloud platforms, such as AWS or Azure, to deploy the model in a scalable and cost-effective way.
- **Continuous Improvement** - Continuously improve the model by retraining it on new data and fine-tuning the hyperparameters. You can also use techniques such as transfer learning to improve the model's performance.

With proper planning and execution, you can develop a robust and accurate human activity recognition system that can be deployed in a real-world environment.



5.2 ALGORITHMS

Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are two types of neural network architectures used in machine learning for various applications, including image recognition, natural language processing, and time series analysis.

5.2.1 Convolutional Neural Network (CNN)

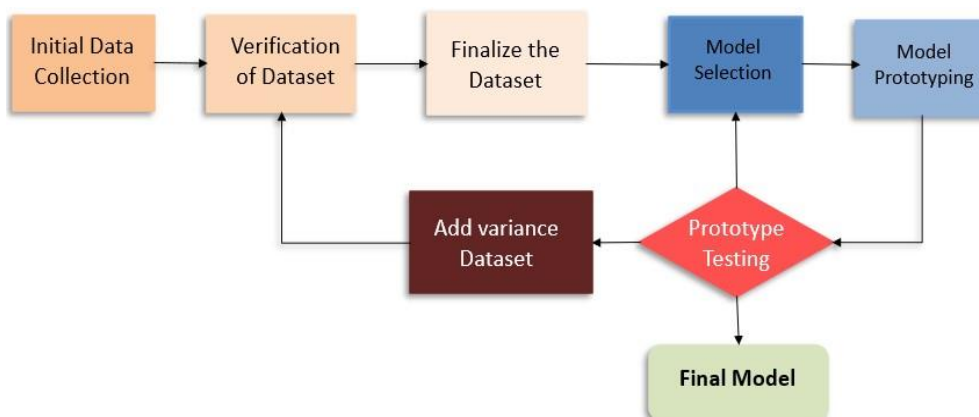
CNN is a deep learning architecture primarily used for image and video recognition. CNNs use a series of convolutional and pooling layers to extract and learn features from input images, followed by fully connected layers for classification. The convolutional layer applies a filter or kernel to the input image, sliding it across the image and computing a dot product at each position. The pooling layer then reduces the spatial size of the image by aggregating the output of the previous layer. This process of convolution and pooling is repeated multiple times to extract higher-level features from the input image.

CNNs are highly effective for image recognition tasks and have achieved state-of-the-art performance on various benchmark datasets.

5.2.2 Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) that is designed to model sequences of data, such as speech, text, and time series. Unlike traditional RNNs, which suffer from the vanishing gradient problem when modelling long sequences, LSTM uses a gated memory mechanism to store and retrieve information over time. Each LSTM cell has three gates - input gate, forget gate, and output gate - that control the flow of information into, out of, and within the cell. The input gate decides which information to update, the forget gate decides which information to discard, and the output gate decides which information to output. The LSTM architecture allows for the modelling of long-term dependencies in the input data and has been widely used for various applications, including speech recognition, language translation, and time series analysis.

Overall, CNN and LSTM are two powerful neural network architectures that have been used in various machine learning applications. While CNN is mainly used for image and video recognition, LSTM is used for modelling sequences of data. Both architectures have achieved state-of-the-art performance on various benchmark datasets and continue to be widely used and researched in the machine learning community.



5.3 TESTING

The following testing techniques can be used to evaluate their performance:

- **Hold-Out Testing** - In this technique, the dataset is divided into two parts: a training set and a testing set. The model is trained on the training set and then evaluated on the testing set to determine its accuracy.
- **Cross-Validation** - As described earlier, this technique can also be used for CNN and LSTM algorithms to evaluate their performance.
- **Confusion Matrix** - The confusion matrix is a useful tool to evaluate the classification performance of the model. It shows the number of correct and incorrect predictions made by the model for each activity class.
- **F1 Score** - The F1 score is a measure of the model's accuracy, calculated as the harmonic mean of precision and recall. It is a useful metric for evaluating the overall performance of the model.

CHAPTER 6

OUTCOMES AND DISCUSSION

6.1 POTENTIAL OUTCOMES:

- **Enhanced User Experience:** By understanding user emotions, the system can recommend songs that are more likely to resonate with their current mood, leading to a more enjoyable and personalized music listening experience.
- **Improved Music Discovery:** The system can introduce users to new music genres and artists that they might not have discovered otherwise, based on their emotional state.
- **Emotional Well-being:** Music can be a powerful tool for managing emotions. The ability to recommend uplifting or calming songs based on emotion detection can potentially contribute to a user's emotional well-being.
- **Reduced Decision Fatigue:** Choosing the right song can be time-consuming. Emotion-based recommendations can help users find the perfect music selection quickly and efficiently.

6.2 DISCUSSION POINTS:

- **Accuracy of Emotion Detection:** Emotion detection is a complex task, and the accuracy of the system's recommendations will depend on how well it can identify user emotions through text or voice chat.
- **Privacy Concerns:** Emotion detection raises privacy concerns. It's important to be transparent about how user data is collected, used, and stored to build trust with users.
- **Limited Scope of Emotions:** Current emotion detection models might not capture the full spectrum of human emotions. The system might struggle with more nuanced emotional states.
- **Over-reliance on Emotion:** While emotions are important, other factors

like genre preference or familiarity might also influence song selection. The system should find a balance between emotional state and user preferences.

- **Integration with Streaming Services:** Integrating with popular music streaming services would be crucial for wider adoption and user convenience.

Overall, chatbot song recommender systems with emotion detection hold promise for a more personalized and emotionally-driven approach to music discovery.

6.3 CONSTRAINTS

There are several constraints that may affect the performance and accuracy of the system. Some of these constraints include:

- **Limited Data Availability** - system require a significant amount of labeled data to train the CNN and LSTM models. Without access to large amounts of high-quality labeled data, the accuracy and performance of the system can be limited.
- **Computational Resource** - LSTM models require significant computational resources for training and inference. Without access to powerful computing resources, training and inference can be slow or infeasible.
- **Model Complexity** – LSTM can be complex, with many layers and parameters. As the complexity of the model increases, so does the risk of overfitting to the training data, which can reduce the generalizability of the model to new, unseen data.
- **Interpretability** - CNN and LSTM models can be challenging to interpret, which can limit the ability of researchers and practitioners to understand the factors that contribute to human activities and improve the accuracy of the system.

CHAPTER 7

RESULTS AND DISCUSSION

The chatbot song recommendation system would aim to provide personalized music recommendations to users based on their preferences and listening history. It would integrate the Last.fm API. The result of the chatbot system would typically involve generating song recommendations based on various factors, such as user preferences, favorite artists, genres, or even current mood.

The debate over a Last.fm API-based chatbot music recommendation system might touch on a few different topics.

- **Recommendation accuracy:** The performance of the system might be evaluated by looking at user feedback and satisfaction with the recommendations.
- **Personalization:** To improve the system's capacity to deliver pertinent recommendations, personalization techniques like collaborative filtering, user profiling, or machine learning algorithms should be investigated.
- **User experience:** The conversational capabilities and user interactivity of the chatbot would also be discussed.
- **Integration with Last.fm API:** This would include subjects including maintaining effective communication between the chatbot and API, dealing with API queries, retrieving and processing data, and handling authentication.
- **Ethical considerations:** Discussion topics would include the effect of the recommendation system on user privacy, data security, and the potential for biased or filter bubble-like recommendations

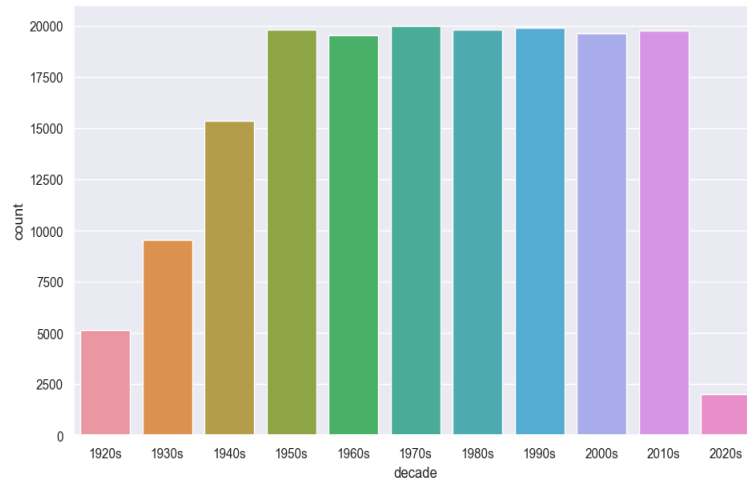


Fig 7.1: Count of songs for decade

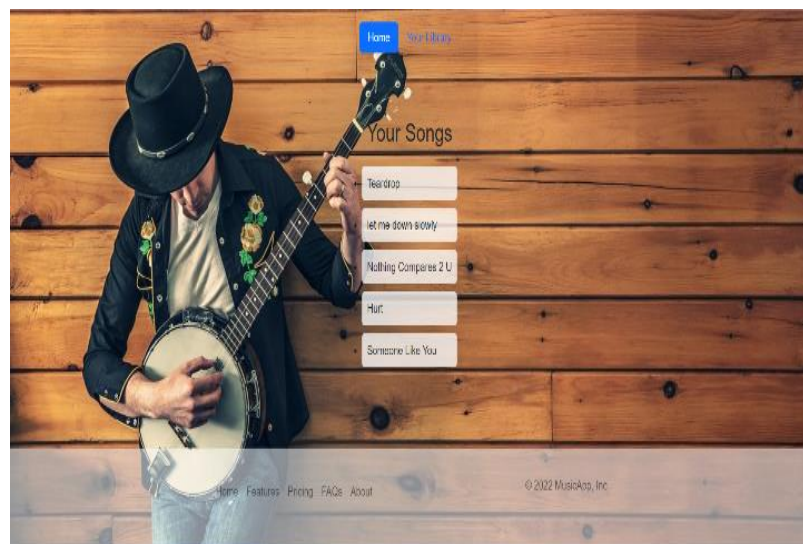


Fig 7.3: Songs Generated for sad mood

CHAPTER 8

CONCLUSION

8.1 CONCLUSION

Our project's subfocus is on the development of the Flask framework and API, with a focus on chatbots, emotional tone analysis, and song recommendation systems. It also discusses the fields of machine learning and artificial intelligence, how they might be combined with chatbots, and how this impacts the chatbot's responsiveness. When paired with song recommendations, emotional analysis can recognize user emotions and play music that speaks to those feelings.

The project was successful in gathering user feedback, assessing the user's emotional state, displaying music to the user, and suggesting related songs to the user based on the songs that were shown. Songs offering platforms like Spotify, SoundCloud, iTunes, YouTube Music, and more will benefit from this initiative for improved.

8.2 RECOMMENDATIONS FOR FUTURE RESEARCH

There are several potential areas of future work for chatbot song recommender system using LSTM algorithms. The initiative has enormous expansion potential. You may either incorporate the project as an additional feature within an application or scale it up and distribute it on services like Heroku. It's running on a localhost now. Docker can be installed on a container by any cloud provider, such as Azure, Heroku, AWS, and others.

However, it may slow down the system and require a lot of memory. To enhance cognitive skills and make the user experience better for both the chatbot and the user, voice or facial recognition for user input data can be used. The consumer may be offered with multiple playlists to select from, instead than just a few music recommendations. larger datasets and more advanced models of emotion recognition as teaching resources can be implemented.

REFERENCES

- [1] Musengamana Jean de Dieu, Peng Liang, Mojtaba Shahin, How Do Developers Search for Architectural Information? An Industrial Survey, March 2022, DOI: doi.org/10.1109/ICSA53651.2022.00014.
- [2] Sheetal Kusal, Shruti Patil, Jyoti Choudrie, Ketan Kotecha, A Review on Text-Based Emotion Detection Techniques, Applications, Datasets, and Future Directions, April 2022, LicenseCC BY 4.0
- [3] Yung-Chun Chang, Yan-Chun Hsing, Emotion-infused deep neural network for emotionally resonant conversation, August 2020, <https://doi.org/10.1016/j.asoc.2021.107861>
- [4] Nalini Chinnu, Vinothkumar S, Venkata Prasanna R, Spotify Genre Recommendation Based On User Emotion Using Deep Learning, July 2022, DOI: doi.org/10.1109/CCICT56684.2022.00081
- [5] Sindhura Kannappan, SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING, April 2023, DOI: doi.org/10.5281/zenodo.7766376
- [6] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, An investigation on the user interaction modes of conversational recommender systems for the music domain, April 2020, DOI: doi.org/10.1007/s11257-019-09250-7.
- [7] Nishtha Kapoor, Arushi Gupta, Gulshan Kumar, Dhruv Aggarwal, MU-SYNC - A Music Recommendation Bot, January 2022, DOI: doi.org/10.46501/IJMTST0801003
- [8] G. V. Padma Raju, "Developing a Chat bot using Machine Learning", International Journal of Recent Technology and Engineering, DOI: [doi.org/Vol. 8, No. 3, PP. 89-92, 2019](https://doi.org/Vol.8, No. 3, PP. 89-92, 2019)
- [9] Enric Plaza, Case-Based Sequential Ordering of Songs for Playlist Recommendation, September 2006, DOI: doi.org/10.1007/11805816_22
- [10] Rula M Ali Farkash, Tengku Zatul Hidayah Tengku Petra, An AI Chatbot for Personalized Music Recommendations Based on User Emotions, DOI: [doi.org/https://doi.org/10.24191/jcrinn.v9i1.427](https://doi.org/10.24191/jcrinn.v9i1.427)
- [11] Anvit Kulkarni, Prasad Mahajan, Gaurav Nimbokar, Narayan Raut, AI Based Song Recommendations System, June 2023, Conference: IDES and

APPENDIX

A. SOURCE CODE

```
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from sklearn.metrics import euclidean_distances
from scipy.spatial.distance import cdist
import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv("data.csv")
genre_data = pd.read_csv('data_by_genres.csv')
year_data = pd.read_csv('data_by_year.csv')

data.shape

genre_data.shape

year_data.shape

print(data.info())

print(genre_data.info())

print(year_data.info())

def get_decade(year):
    period_start = int(year/10) * 10
    decade = '{}s'.format(period_start)
    return decade
data['decade'] = data['year'].apply(get_decade)
sns.set(rc={'figure.figsize':(11 ,6)})
sns.countplot(data['decade'])
```

```

#univariate analysis
plt.figure(figsize=(14,8))
plt.subplot(121)
data['year'].value_counts().plot(kind='pie')
plt.subplot(122)
data['year'].value_counts().plot(kind='pie',autopct='%.2f')
plt.show()

univariate analysis for numerical data
plt.figure(figsize=(14,4))
plt.subplot(121)
data.year.plot(kind='kde')
plt.subplot(122)
data.year.plot(kind='hist')
plt.show()

#bivariate analysis
plt.figure(figsize=(14,8))
plt.subplot(121)
data['mode'].value_counts().plot(kind='bar')
plt.subplot(122)
data['mode'].value_counts().plot(kind='pie',autopct='%.2f')
plt.show()

#boxplot
plt.figure(figsize=(14,4))
sns.boxplot(data['year'])
plt.show()

#check null values
data.isnull().sum()

sound_features = ['acousticness', 'danceability', 'energy', 'instrumentalness', 'liveness', 'valence']
fig = px.line(year_data, x='year', y=sound_features)
fig.show()

top10_genres = genre_data.nlargest(10, 'popularity')
fig = px.bar(top10_genres, x='genres', y=['valence', 'energy', 'danceability', 'acousticness'],
barmode='group')
fig.show()

genre_data.isnull().sum()

```

```

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

cluster_pipeline = Pipeline([('scaler', StandardScaler()), ('kmeans', KMeans(n_clusters=10))])
X = genre_data.select_dtypes(np.number)
cluster_pipeline.fit(X)
genre_data['cluster'] = cluster_pipeline.predict(X)

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

cluster_pipeline = Pipeline([('scaler', StandardScaler()), ('kmeans', KMeans(n_clusters=10))])
X = genre_data.select_dtypes(np.number)
cluster_pipeline.fit(X)
genre_data['cluster'] = cluster_pipeline.predict(X)

#handling outliers and emotion module
percentile25_val=data['valence'].quantile(0.25)
percentile75_val=data['valence'].quantile(0.75)
percentile25_year=data['year'].quantile(0.25)
percentile75_year=data['year'].quantile(0.75)
print("Q2 of val:",percentile25_val)
print("Q4 of val:",percentile75_val)
print("Q2 of year:",percentile25_year)
print("Q4 of year:",percentile75_year)

iqr_val=percentile75_val-percentile25_val
print("IQR of val",iqr_val)
iqr_year=percentile75_year-percentile25_year
print("IQR of year",iqr_year)

upper_limit_val=percentile75_val+1.5*iqr_val
lower_limit_val=percentile25_val-1.5*iqr_val
upper_limit_year=percentile75_year+1.5*iqr_year
lower_limit_year=percentile25_year-1.5*iqr_year

print("upper limit of val",upper_limit_val)
print("lower limit of val",lower_limit_val)
print("upper limit of year",upper_limit_year)
print("lower limit of year",lower_limit_year)

```

```

from sklearn.manifold import TSNE

tsne_pipeline = Pipeline([('scaler', StandardScaler()), ('tsne', TSNE(n_components=2,
verbose=1))])
genre_embedding = tsne_pipeline.fit_transform(X)
projection = pd.DataFrame(columns=['x', 'y'], data=genre_embedding)
projection['genres'] = genre_data['genres']
projection['cluster'] = genre_data['cluster']

fig = px.scatter(
    projection, x='x', y='y', color='cluster', hover_data=['x', 'y', 'genres'])
fig.show()

song_cluster_pipeline = Pipeline([('scaler', StandardScaler()),
                                   ('kmeans', KMeans(n_clusters=20,
                                   verbose=False))
                                   ], verbose=False)

X = data.select_dtypes(np.number)
number_cols = list(X.columns)
song_cluster_pipeline.fit(X)
song_cluster_labels = song_cluster_pipeline.predict(X)
data['cluster_label'] = song_cluster_labels

#Visualizing the Clusters with PCA
#sentiment analysis
from sklearn.decomposition import PCA

pca_pipeline = Pipeline([('scaler', StandardScaler()), ('PCA', PCA(n_components=2))])
song_embedding = pca_pipeline.fit_transform(X)
projection = pd.DataFrame(columns=['x', 'y'], data=song_embedding)
projection['title'] = data['name']
projection['cluster'] = data['cluster_label']

fig = px.scatter(
    projection, x='x', y='y', color='cluster', hover_data=['x', 'y', 'title'])
fig.show()

pip install spotipy

plt.figure(figsize=(16,5))
plt.subplot(121)
sns.distplot(data['valence'])

```

```

plt.subplot(122)
sns.distplot(data['year'])
plt.show()

import spotipy
from spotipy.oauth2 import SpotifyClientCredentials
from collections import defaultdict

#sp =
spotipy.Spotify(auth_manager=SpotifyClientCredentials(client_id=os.environ["SPOTIFY_CLIENT_
ID"],
                                                    #client_secret=os.environ["SPOTIFY_CLIENT_SECRET"]))

def find_song(name, year):
    song_data = defaultdict()
    results = sp.search(q= 'track: {} year: {}'.format(name,year), limit=1)
    if results['tracks']['items'] == []:
        return None

    results = results['tracks']['items'][0]
    track_id = results['id']
    audio_features = sp.audio_features(track_id)[0]

    song_data['name'] = [name]
    song_data['year'] = [year]
    song_data['explicit'] = [int(results['explicit'])]
    song_data['duration_ms'] = [results['duration_ms']]
    song_data['popularity'] = [results['popularity']]

    for key, value in audio_features.items():
        song_data[key] = value

    return pd.DataFrame(song_data)

data.corr()

plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True,cbar=True,linewidth=True,fmt='f')
plt.show()

data.duplicated().sum()

#handling outliers and emotion module
percentile25_val=data['valence'].quantile(0.25)

```

```

percentile75_val=data['valence'].quantile(0.75)
percentile25_year=data['year'].quantile(0.25)
percentile75_year=data['year'].quantile(0.75)
print("Q2 of val:",percentile25_val)
print("Q4 of val:",percentile75_val)
print("Q2 of year:",percentile25_year)
print("Q4 of year:",percentile75_year)

iqr_val=percentile75_val-percentile25_val
print("IQR of val",iqr_val)
iqr_year=percentile75_year-percentile25_year
print("IQR of year",iqr_year)

upper_limit_val=percentile75_val+1.5*iqr_val
lower_limit_val=percentile25_val-1.5*iqr_val
upper_limit_year=percentile75_year+1.5*iqr_year
lower_limit_year=percentile25_year-1.5*iqr_year

print("upper limit of val",upper_limit_val)
print("lower limit of val",lower_limit_val)
print("upper limit of year",upper_limit_year)
print("lower limit of year",lower_limit_year)

data[(data['year']>54) | (data['year']<14)]

capping of outliers
data['valence']=np.where(
data['valence']>upper_limit_val,
upper_limit_val,
np.where(
data['valence']<lower_limit_val,
lower_limit_val,data['valence'])
)

data['valence'].describe()

sns.boxplot(data['valence'])
plt.show()

data['year'].describe()

sns.boxplot(data['year'])
plt.show()

```

```

X=data.iloc[:, :-1]
y=data.iloc[:, -1]

y.value_counts()

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

X_train.shape,X_test.shape

X_train.sample(5)

X_test.sample(5)

X_train.isnull().mean()*100

data_cat=X_train.select_dtypes(include='object')
data_cat.sample(5)

data_num=X_train.select_dtypes(exclude='object')
data_num.sample(5)

from collections import defaultdict
from sklearn.metrics import euclidean_distances
from scipy.spatial.distance import cdist
import difflib

number_cols = ['valence', 'year', 'acousticness', 'danceability', 'duration_ms', 'energy', 'explicit',
               'instrumentalness', 'key', 'liveness', 'loudness', 'mode', 'popularity', 'speechiness', 'tempo']

def get_song_data(song, spotify_data):

    try:
        song_data = spotify_data[(spotify_data['name'] == song['name'])
                                & (spotify_data['year'] == song['year'])].iloc[0]
        return song_data

    except IndexError:
        return find_song(song['name'], song['year'])

def get_mean_vector(song_list, spotify_data):

```

```

song_vectors = []

for song in song_list:
    song_data = get_song_data(song, spotify_data)
    if song_data is None:
        print('Warning: {} does not exist in Spotify or in database'.format(song['name']))
        continue
    song_vector = song_data[number_cols].values
    song_vectors.append(song_vector)

song_matrix = np.array(list(song_vectors))
return np.mean(song_matrix, axis=0)

def flatten_dict_list(dict_list):

    flattened_dict = defaultdict()
    for key in dict_list[0].keys():
        flattened_dict[key] = []

    for dictionary in dict_list:
        for key, value in dictionary.items():
            flattened_dict[key].append(value)

    return flattened_dict

def recommend_songs( song_list, spotify_data, n_songs=10):

    metadata_cols = ['name', 'year', 'artists']
    song_dict = flatten_dict_list(song_list)

    song_center = get_mean_vector(song_list, spotify_data)
    scaler = song_cluster_pipeline.steps[0][1]
    scaled_data = scaler.transform(spotify_data[number_cols])
    scaled_song_center = scaler.transform(song_center.reshape(1, -1))
    distances = cdist(scaled_song_center, scaled_data, 'cosine')
    index = list(np.argsort(distances)[:n_songs][0])

    rec_songs = spotify_data.iloc[index]
    rec_songs = rec_songs[~rec_songs['name'].isin(song_dict['name'])]
    return rec_songs[metadata_cols].to_dict(orient='records')

import nltk
import random

```

```

# Make sure you have NLTK's VADER sentiment analysis tool and its data downloaded
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Sample song database
song_database = {
    "happy": ["Happy Song 1", "Happy Song 2", "Happy Song 3"],
    "sad": ["Sad Song 1", "Sad Song 2", "Sad Song 3"],
    "angry": ["Angry Song 1", "Angry Song 2", "Angry Song 3"],
    "calm": ["Calm Song 1", "Calm Song 2", "Calm Song 3"],
}

# Initialize the sentiment analyzer
sid = SentimentIntensityAnalyzer()

# Function to recommend a song based on the detected emotion
def recommend_song(text):
    sentiment_scores = sid.polarity_scores(text)

    # Determine the dominant emotion
    emotion = "neutral"
    if sentiment_scores['compound'] >= 0.05:
        emotion = "happy"
    elif sentiment_scores['compound'] <= -0.05:
        emotion = "sad"

    # Recommend a song based on the detected emotion
    if emotion in song_database:
        songs = song_database[emotion]
        recommended_song = random.choice(songs)
        return f"Based on your emotion, I recommend the song: {recommended_song}"
    else:
        return "I couldn't detect a specific emotion. Please try again."

# Chatbot loop
while True:
    user_input = input("You: ")
    if user_input.lower() == "exit":
        break

    response = recommend_song(user_input)
    print("Chatbot:", response)

```

B.SCREENSHOTS

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2973 entries, 0 to 2972
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   mode                   2973 non-null  int64
1   genres                 2973 non-null  object
2   acousticness           2973 non-null  float64
3   danceability           2973 non-null  float64
4   duration_ms            2973 non-null  float64
5   energy                 2973 non-null  float64
6   instrumentalness        2973 non-null  float64
7   liveness               2973 non-null  float64
8   loudness               2973 non-null  float64
9   speechiness            2973 non-null  float64
10  tempo                  2973 non-null  float64
11  valence                2973 non-null  float64
12  popularity              2973 non-null  float64
13  key                    2973 non-null  int64
dtypes: float64(11), int64(2), object(1)
memory usage: 325.3+ KB
None
```

Fig B.1: Data set used

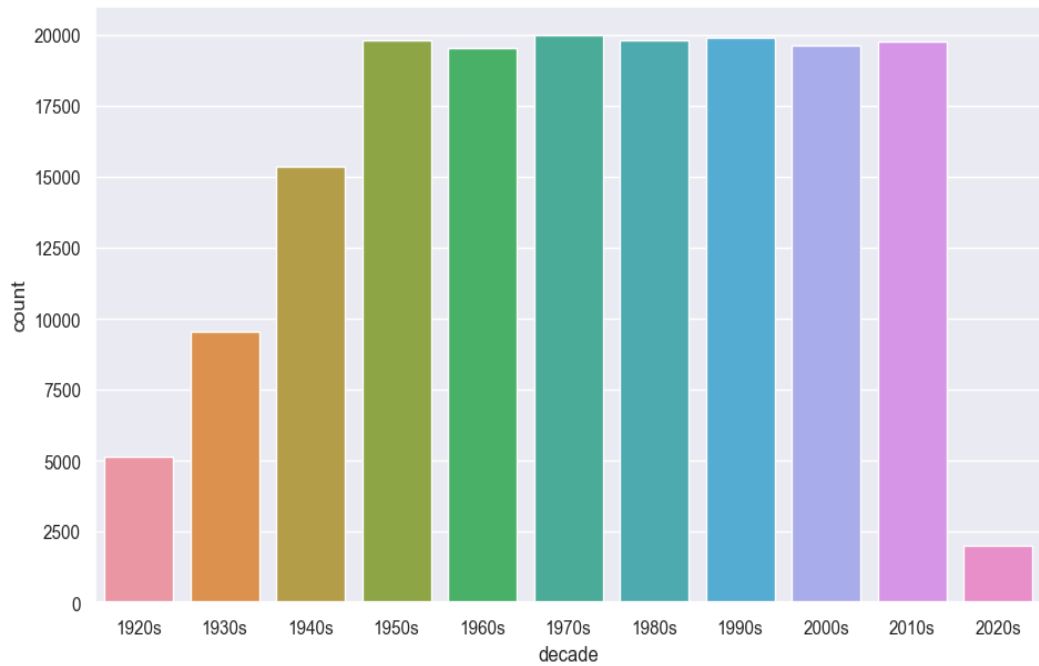


Fig B.2: subplot of decade and count

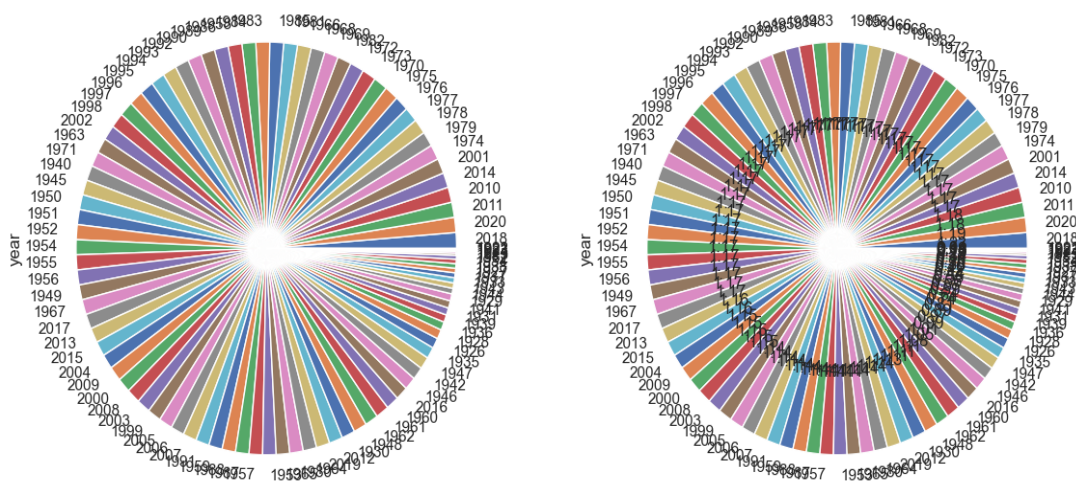


Fig B.3:Univariate Analysis

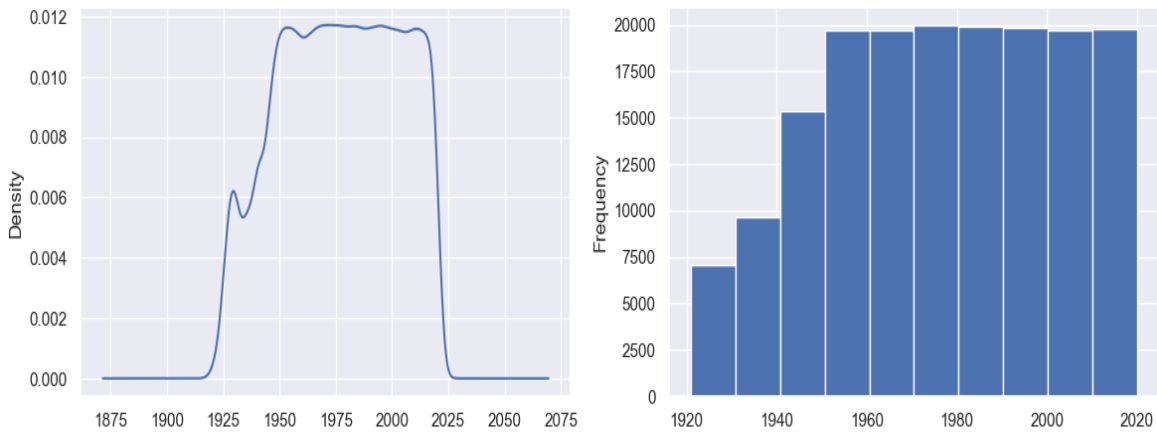


Fig B.4:Univariate analysis for numerical data

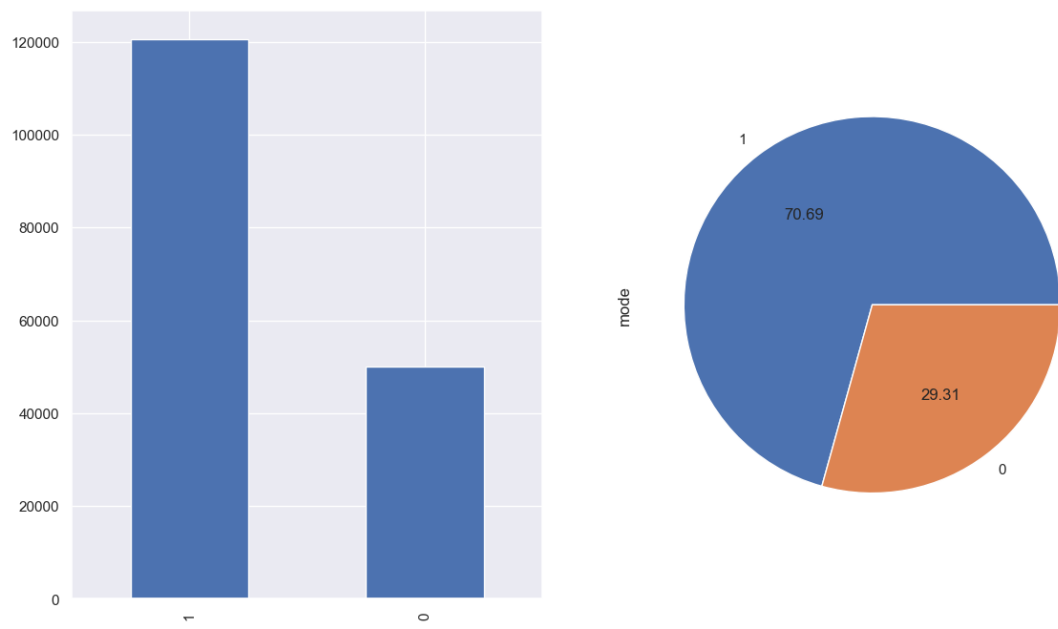


Fig B.5: Bivariate analysis

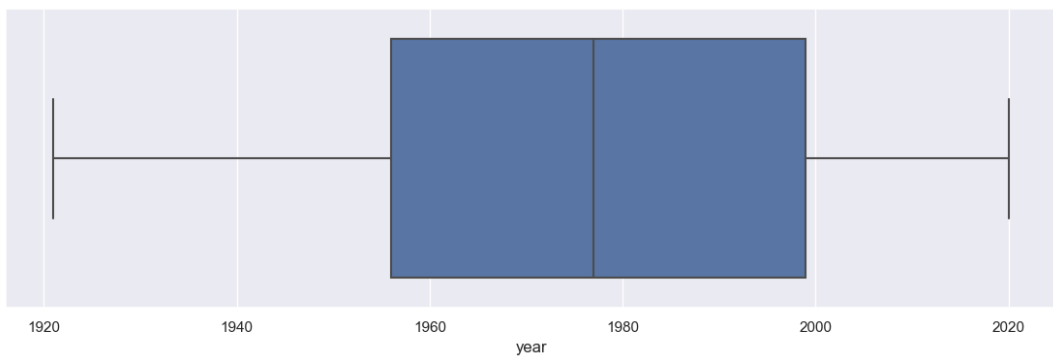


Fig B.6: Boxplot

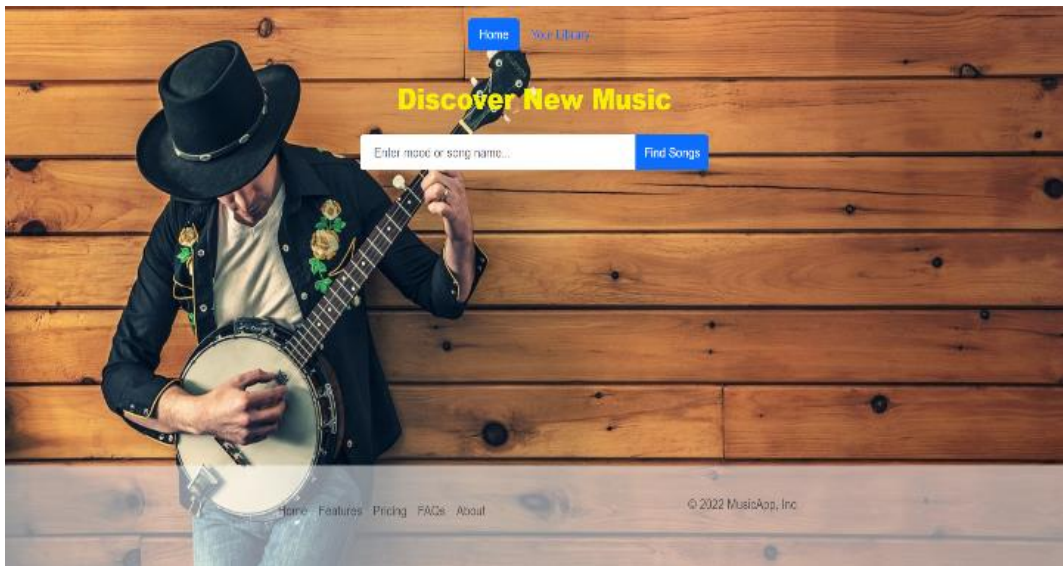


Fig B.7: User interacting website

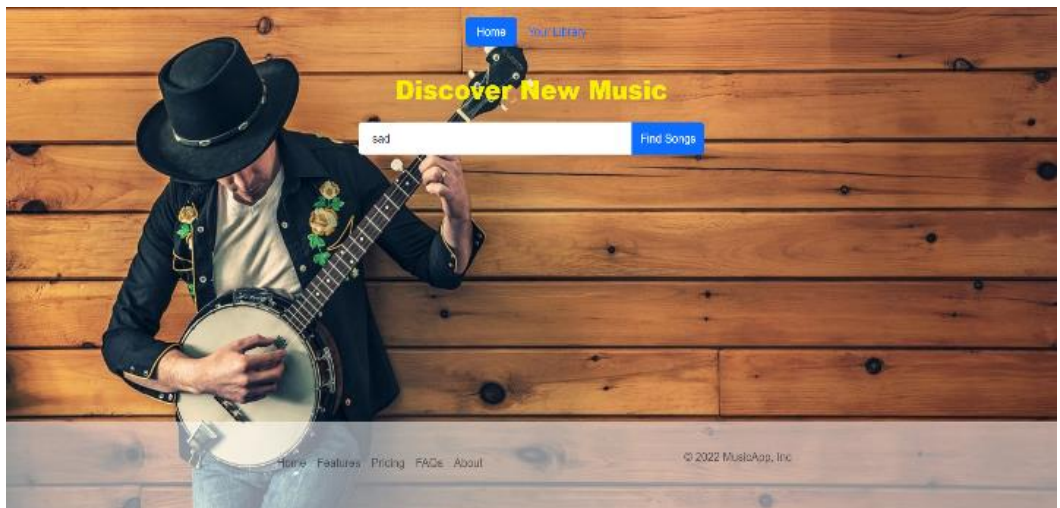


Fig B.8: Entering the emotion



Fig B.9: Song Generation

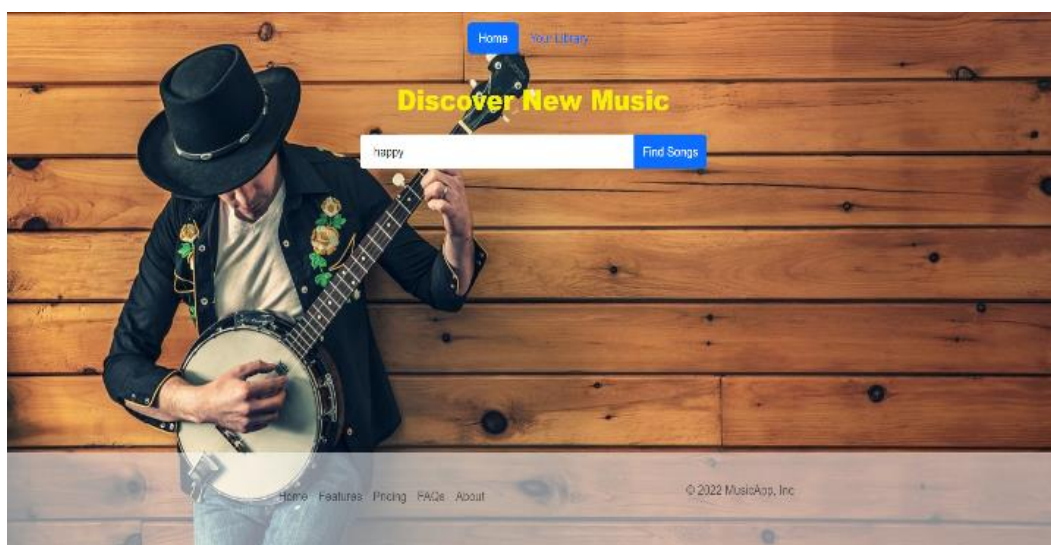


Fig B.10: Entering alternate emotion

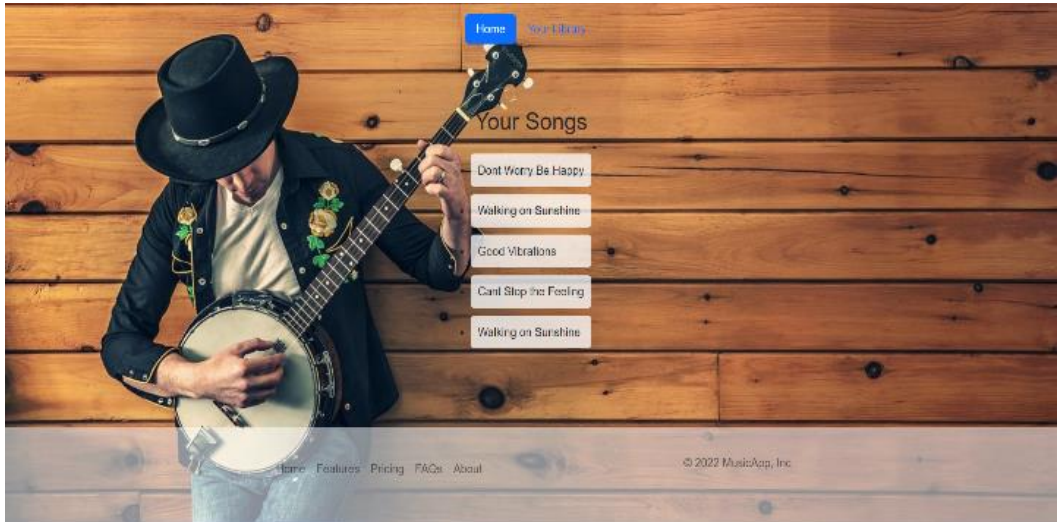


Fig B.11: Song Generation for happy mood

C.RESEARCH PAPER

Chatbot Song Recommender system based on Emotion Detection

Gajjela Sai Charan¹, Lavanya Bhumireddy² and Bala Sai Gayathri³

Author Affiliations

1,2,3 Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India -600119

Author Emails

- a) Corresponding author: gajjelsaicharan2003@gmail.com
- b) lavanyabhumireddy9398@gmail.com
- c) balasaigayathri.cse@sathyabama.ac.in

Abstract: This research project explores the development of a chatbot-based song recommender system that leverages emotion detection from text to personalize music recommendations. The system utilizes natural language processing (NLP) techniques to analyze textual input from users through a chatbot interface. By extracting emotional cues from the text, the system can identify the user's current emotional state and recommend appropriate songs from a Spotify dataset. This approach aims to enhance the music recommendation experience by providing users with songs that resonate with their emotional state, fostering a more personalized and emotionally engaging experience. This project has the potential to revolutionize music recommendation by providing a more personalized and emotionally driven experience. The chatbot-based interface offers a natural and interactive way for users to express their emotions and receive tailored recommendations. This could lead to increased user engagement with music platforms and a deeper understanding of the emotional connection between music and listeners.

Keywords: Chatbot, Song Recommendation, Emotion Detection, NLP, Spotify, Personalized Music

INTRODUCTION

The ability of computer software to interpret human language is recognized as natural language processing, and it has multiple uses in the modern world. Text-to-emotion recognition is the most often used application. Recognizing and acknowledging emotions Text analysis is a relatively modern field of study that shares many similarities with sentiment analysis. Sentiment analysis, a subfield of the study of natural languages (NLP), looks for either positive, neutral, or negative emotions in text.

Emotion analysis, though it is the process of identifying and identifying various types of emotions through the expression of texts, such as anxiety, disdain outrage, joy, sorrow, anger, and shock, emphasizes the study of subjective beliefs or emotions gleaned from various sources regarding a particular topic. Even while it can be challenging to understand someone's experiences, it can be far more challenging to understand the subject matter of their writing due to the wide range of ways in which people choose to express their emotions.

One of the most challenging components of identifying emotion in a text is its context dependence. For instance, think about the term "Shut up!" It contains a component of fury without resorting to the term "anger" or any of its synonyms. In our project, emotion detection is an essential procedure. Acknowledging these feelings from a text contributes to an essential part of our application. Four supervised machine learning methods are used in our project to help determine the user's emotions: Support Vector Machine (SVM), Logistic Regression, Naïve Bayes, Trees. The proposed framework

will identify a few fundamental emotions What is required by the user Deep Learning algorithms are implemented in the application's Chatbot module. We developed a straightforward chatbot that uses configured input patterns and responses, retrieval-based. We used a unique convolutional neural network called Long Short-Term Memory which allows the chatbot to classify user messages and provide relevant responses.

DESCRIPTION

The web application presented in the paper "Chatbot with Song Recommendation based on Emotion" 1 interfaces with users via a chatbot interface, evaluates the emotions of talks, and generates a playlist of songs depending on the user's feelings. Three components comprise the application: a chatbot, an emotion detection module, and a music generation model. A distinct convolutional neural network called Long Short-Term Memory allows the chatbot to classify user messages and provide relevant responses.

The chatbot module uses predefined input patterns and responses. Users' feelings articulated in conversations are detected using SVM, Logistic Regression, Naïve Bayes, Trees. A collection of songs is suggested by the song generating model based on the user's emotions. Deep learning, natural language processing, and other techniques are used in the system, which is implemented as a web application. The application's objective is to identify the feeling the user is experiencing. Once this has been done, a selection of songs is presented according to the mood.

Here's a description of the modules incorporated using the given tools and libraries.

Chatbot Module

Chat bot modules are designed to provide a conversational interface with the user, allowing for natural language exchanges for music recommendations. Rasa or Dialogflow would be good frameworks to use for constructing the chatbot structure and managing discussions.

Chatbot modules' functions include greeting users and explaining their purpose. Feedback is provided during the emotion detection and song choosing processes. It displays music recommendations and allows you to play, skip, or request alternative genres. As needed, engages in additional conversations (for example, offering information about songs, artists, or genres).

Emotion Detection Module

The emotion detection module's purpose is to analyze text input from users with the objective of identifying underlying emotions. Text processing and sentiment analysis can be accomplished with the use of NLP libraries that include TextBlob, NLTK, and spaCy, which were used in the development of this module.

The emotion detecting module's functionalities incorporates preprocessing techniques such as stop word removal, lemmatization, and tokenization. Uses sentiment analysis techniques (such as sentiment scores and emotion labels) to extract emotional information. It takes into consideration extra context-related data like user preferences and history. the maps detected emotions to appropriate musical genres or moods (for example, melancholy to soft music, happiness to lively music).

Song Generation Module

The module's purpose is to determine and provide music recommendations that correlate with observed emotions. The Spotify Web API would enable access to extensive music databases and metadata. Module functionalities comprise the specified music API is queried centered on the mapped emotions

or genres. If available, filters result according to user preferences. Choose a compatible song or playlist for recommendation.

The music recommendation model's underlying methodology, which includes the features evaluated for emotion-based song selection. Assessing the model's performance in acquiring emotional nuances in the input text and mapping them to appropriate tunes. the selection of training data from Spotify's dataset, taking into consideration variables like diversity, genre representation, and relevance to user preferences.

Examine the model's capacity to generalize to different music genres and user preferences. The model associate's emotions sensed by the Emotion Detection Module with the development of emotionally relevant music recommendations. Highlight any difficulties or opportunities for improvement in improving the emotional relevance of the generated song suggestions.

LITERATURE SURVEY

Nikhil et al. [1] employ a variety of methods and technologies, including Blob detection, Canny edge, and Haar cascade, to identify emotions. The system takes user photos and analyzes mood relevant to those images. The image is used to provide inputs such as face and expression, and the system further offers a chat area for responses. The solution that is suggested in the study offers a novel method for creating desktop chatbot applications that combine text and gestures. Via the chat program, the machine is allowed to have a discussion. Depending on the user's response, the software may provide links, webpages, or other content. Smiling and worry are identified by the software. Emoji pop-ups will be displayed on the screen when the algorithm detects a smile.

Jae Sik Lee et al. [2] utilized the theory of deductive reasoning of corpus, which makes use of context data to understand the user's position. They present an integrated context-reasoning music recommendation system in this study. The intention, mood, and recommendation modules—three of the modules in their proposed system—each provide a unique capability and are critical to the system's functioning. By employing contextual reasoning and surrounding background data, the Intention Module determines if the user wishes to listen to music. Then, based on the user's situation, the Mood Module chooses the type of music that is best.

Renuka R. Londhe et al. investigated the idea of facial emotion recognition in [3] by considering the several characteristics connected to an individual's face. Changes in the nose, lips, eyebrows, and mouth area, as well as the facial curvatures, are noticeable whenever there is a shift in the facial expression. Then, these characteristics are divided into six expressions with the use of artificial neural networks, including rage, contempt, astonishment, fear, joy, and grief. Scaled Conjugate Stochastic backward propagation is used to train and test the two-layered feed forward neural network.

In their work [4], Dolly Reney et al. discuss the value of countenance and sentiment recognition in security domain and how to apply it to address a variety of problems. When comparing the sound Mel frequency components and facial features, the database is crucial in determining which face and emotion detection algorithm is used. With the aid of the previously mentioned database, several algorithms are utilized to analyze the face and mood. For the face and emotion recognition and face identification processes, they have employed the Viola-Jones algorithm.

In their paper [5], Ai Thanh Ho et al. introduce a Movie Recommender System (E-MRS) with an emotional focus. This system intends to overcome the shortcomings of recommender systems in understanding and capturing users' ever-changing preferences, as well as the shortcomings of the standard user profile system in evaluating the importance of users' emotions and how they impact their decisions. The report states that EMRS combines content-based and collaborative filtering techniques to give customers a customized list of recommendations. There are additional discussions about the design, implementation, and evaluation procedure of the proposed system. To link emotions to movies, participants are required to answer a survey about the movies they have seen or the kinds of movies they enjoy watching based on their feelings.

ARCHITECTURE

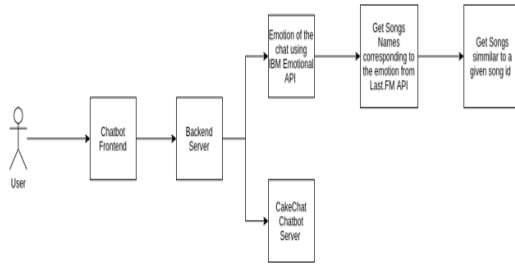


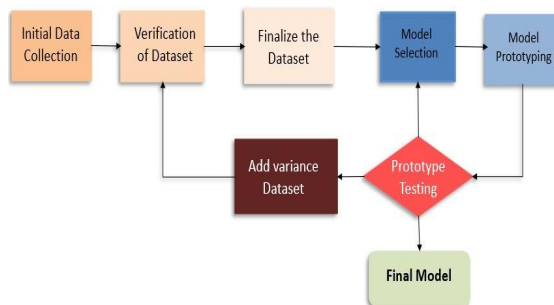
Figure 1

Product Architecture of Music Recommendation system using chatbot.

This figure gave us a clear idea of how this project will seem. In essence, the frontend will display the user interface, but as soon as a user interacts with a chatbot, the frontend will also display that interaction. On the server side, there are primarily two APIs that are used to identify the user's emotion in text by utilizing IBM's Emotional API and Last.FM Application interface to suggest a song depending on the emotion detected. The individual starts the conversation. The emotional content of the discussion is examined using the IBM Emotional API.

The software obtains the famous songs relevant to the feeling it detects by using the Last.fm songs API, which is how the Chatbot will respond to the discussion. Following the hearing of music for a while, the Last.fm API will suggest a comparable song to the user.

METHODOLOGY



The problem statement is straightforward: provide customers with music recommendations from a vast song library. Existing are several live online music recommender systems, each with a unique approach to making music recommendations. Last.fm, for instance, adopts a mechanism wherein similarity scores based on tags, artists, and tracks are considered along with user activity.

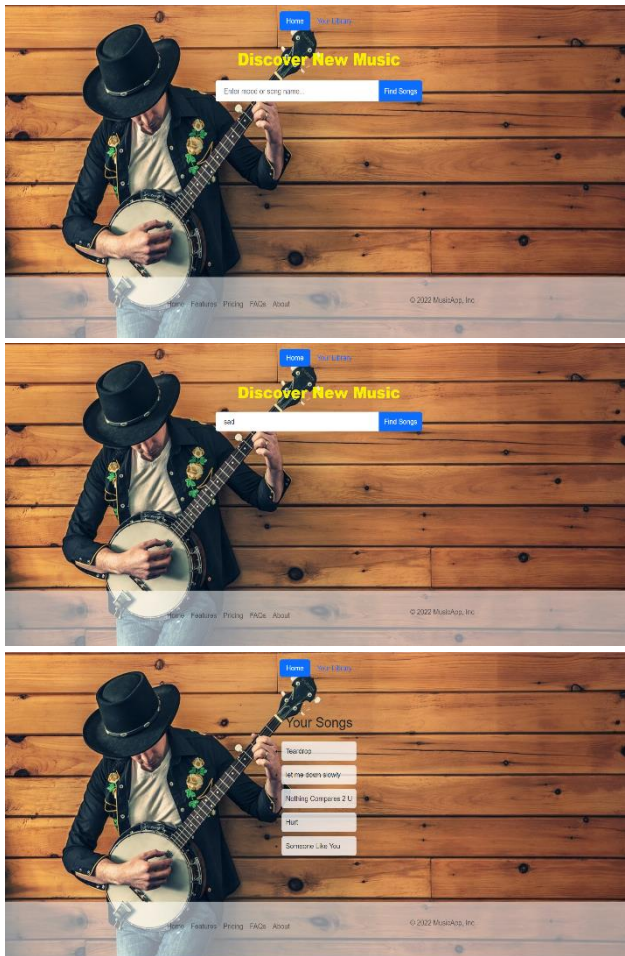
CONCLUSION

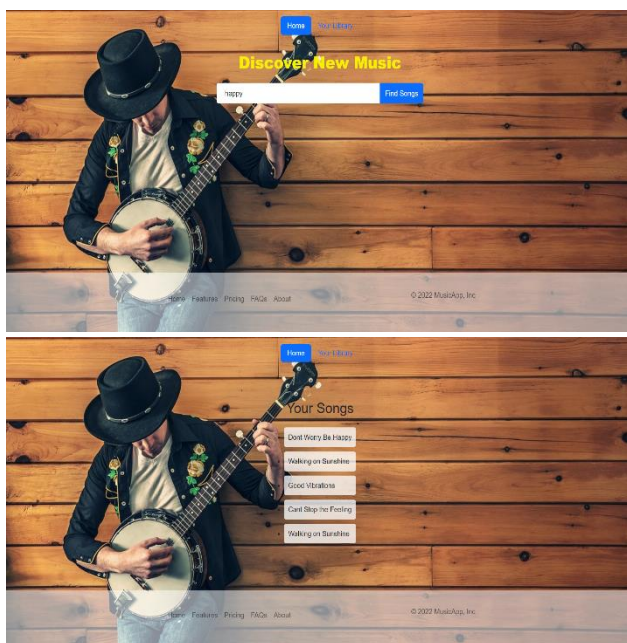
Our project's subfocus is on the development of the Flask framework and API, with a focus on chatbots, emotional tone analysis, and song recommendation systems. It also discusses the fields of machine learning and artificial intelligence, how they might be combined with chatbots, and how this impacts the chatbot's responsiveness. When paired with song recommendations, emotional analysis can recognize user emotions and play music that speaks to those feelings. The project was successful in gathering user feedback, assessing the user's emotional state, displaying music to the user, and suggesting related songs to the user based on the songs that were shown. Songs offering platforms like Spotify, SoundCloud, iTunes, YouTube Music, and more will benefit from this initiative .

FUTURE WORK

The initiative has enormous expansion potential. You may either incorporate the project as an additional feature within an application or scale it up and distribute it on services like Heroku. It's running on a local host now. Docker can be installed on a container by any cloud provider, such as Azure, Heroku, AWS, and others. However, it may slow down the system and require a lot of memory. To enhance cognitive skills and make the user experience better for both the chatbot and the user, voice or facial recognition for user input data can be used. The consumer may be offered multiple playlists to select from, instead than just a few music recommendations. larger datasets and more advanced models of emotion recognition as teaching resources can be implemented.

SCREENSHOTS





REFERENCES

- [1] Musengamana Jean de Dieu, Peng Liang, Mojtaba Shahin, How Do Developers Search for Architectural Information? An Industrial Survey, March 2022, DOI: doi.org/10.1109/ICSA53651.2022.00014.
- [2] Sheetal Kusal, Shruti Patil, Jyoti Choudrie, Ketan Kotecha, A Review on Text-Based Emotion Detection Techniques, Applications, Datasets, and Future Directions, April 2022, LicenseCC BY 4.0
- [3] Yung-Chun Chang, Yan-Chun Hsing, Emotion-infused deep neural network for emotionally resonant conversation, August 2020, <https://doi.org/10.1016/j.asoc.2021.107861>
- [4] Nalini Chinnu, Vinothkumar S, Venkata Prasanna R, Spotify Genre Recommendation Based On User Emotion Using Deep Learning, July 2022, DOI: doi.org/10.1109/CCICT56684.2022.00081
- [5] Sindhura Kannappan, SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING, April 2023, DOI: doi.org/10.5281/zenodo.7766376
- [6] Fedelucio Narducci, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, An investigation on the user interaction modes of conversational recommender systems for the music domain, April 2020, DOI: doi.org/10.1007/s11257-019-09250-7.
- [7] Nishtha Kapoor, Arushi Gupta, Gulshan Kumar, Dhruv Aggarwal, MU-SYNC - A Music Recommendation Bot, January 2022, DOI: doi.org/10.46501/IJMTST0801003
- [8] G. V. Padma Raju, "Developing a Chat bot using Machine Learning", International Journal of Recent Technology and Engineering, DOI: doi.org/Vol. 8, No. 3, PP. 89-92, 2019

- [9] Enric Plaza, Case-Based Sequential Ordering of Songs for Playlist Recommendation, September 2006, DOI: doi.org/10.1007/11805816_22
- [10] Rula M Ali Farkash, Tengku Zatul Hidayah Tengku Petra, An AI Chatbot for Personalized Music Recommendations Based on User Emotions, DOI: [doi.org/https://doi.org/10.24191/jcrinn.v9i1.427](https://doi.org/10.24191/jcrinn.v9i1.427)
- [11] Anvit Kulkarni, Prasad Mahajan, Gaurav Nimbokar, Narayan Raut, AI Based Song Recommendations System, June 2023, Conference: IDES and Association of Computer Electrical Electronics and Communication Engineers (ACEECom)At: Pune
- [12] Christina, Z. K. A. Baizal, Book Recommender System Using Singular Value Decomposition Combined with Slope One Algorithm, August 2022, DOI: doi.org/10.1109/ICoICT55009.2022.9914884
- [13] Yucheng Jin, Li Chen, Wanling Cai, Xianglin Zhao, CRS-Que: A User-Centric Evaluation Framework for Conversational Recommender Systems, November 2023, DOI: doi.org/10.1145/3631534

CERTIFICATES



PANIMALAR ENGINEERING
An Autonomous Institution & Affili
Approved by AICTE, New Delhi, Accredited
Approved by UGC for
Bangalore Trunk Road, Poor



PCTE

This is to certify that Dr./Mr./Ms.....B
.....CSE.....from..Sathyabama Institution of
entitled..CHATBOT SONG RECOMMENDER SYSTEM
..... in 7th
Computing(IConIC) held on 22nd & 23rd M



FACULTY CO-ORDINATOR



CO - CONVENER
Prof. S. VIMALA,M.TECH.,(F



IEEE
MADRAS SECTION



PANIMALAR ENGINEERING

An Autonomous Institution & Affiliated
Approved by AICTE, New Delhi, Accredited
Approved by UGC for
Bangalore Trunk Road, Poon



PECTEA

This is to certify that Dr./Mr./Ms.....B
.....CSE.....from Sathyabama Institution of
entitled CHATBOT SONG RECOMMENDER SYSTEM
..... in 7th
Computing(IConIC) held on 22nd & 23rd M/

Prof. S. Vimala

FACULTY CO-ORDINATOR

Prof. S. Vimala

CO - CONVENER

Prof. S. VIMALA, M.TECH., (P



IEEE
MADRAS SECTION

(



PANIMALAR ENGINEERING

An Autonomous Institution & Affiliated
Approved by AICTE, New Delhi, Accredited
Approved by UGC for PG Studies
Bangalore Trunk Road, Pochampalli



PECTEA

This is to certify that Dr./Mr./Ms.....!
.....CSE.....from..Sathyabama Institution of
entitled...CHATBOT SONG RECOMMENDER SYSTEM
..... in 7th
Computing(IConIC) held on 22nd & 23rd Nov

Gracy K

FACULTY CO-ORDINATOR

S. Vimala

CO - CONVENER

Prof. S. VIMALA, M.TECH.,(

