Music Recommendation using Chatbot

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Nikhil Jindal (201511)

Anushka Pandey (201488)

Under the guidance & supervision of

Dr. Aman Sharma

Assistant Professor (SG)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "'**Music Recommendation using Chatbot'** in partial fulfillment of the requirements for the award of the degree of B.Tech in Information Technology and submitted to the Department of Computer Science & Engineering And Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by **"Anushka Pnadey (201488)** and **Nikhil Jindal (201511)"** during the period from August 2023 to May 2024 under the supervision of **Dr. Aman Sharma** (Assistant Professor (SG), Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

(Student Signature with Date)

Nikhil Jindal

(201511)

(Student Signature with Date)

Anushka Pandey

(201488)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Aman Sharma

Assistant Professor (SG)

Department of Computer Science & Engineering and Information Technology

Dated:

Candidate's Declaration

We hereby declare that the work presented in this report entitled **'Music Recommendation using Chatbot'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Aman Sharma** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Nikhil Jindal
Roll No.: 201511

(Student Signature with Date)
Student Name: Anushka Pandey
Roll No.: 201488

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr. Aman Sharma
Designation: Assistant Professor (SG)
Department: Computer Science and Engineering/Information Technology Department
Dated:

Acknowledgement

To begin, I would like to express my heartfelt gratitude to almighty God for his heavenly grace, which enabled us to successfully complete the project work.

I am extremely grateful and wish to express my deep gratitude to Supervisor Dr. Aman Sharma, Assistant Professor (SG), Department of CSE & IT Jaypee University of Information Technology, Waknaghat. His never-ending patience, intellectual direction, persistent encouragement, constant and vigorous supervision, constructive criticism, helpful suggestions, and reading numerous poor versions and revising them at all stages allowed this project to be completed. I would like to express my heartiest gratitude to Dr. Aman Sharma, Department of CSE&IT, for his kind help to finish my project.

I would also like to express my gratitude to everyone who has assisted me in making this project a success, whether directly or indirectly. In this unusual scenario, I'd like to express my gratitude to the different staff members, both teaching and non-teaching, who have provided me with valuable assistance and assisted my project. Finally, I must express my gratitude for my parents' unwavering support and patience.

Nikhil Jindal
(201511)

Anushka Pandey
(201488)

Table of Content

List of Figures

List of Abbreviations

1. API: Application programming interface
2. NLP: Natural language processing
3. JSON: JavaScript Object Notation
4. NLU: Natural language understanding
5. AWS: Amazon Web Services
6. GCP: Google Cloud Platform
7. NLTK: Natural Language Toolkit
8. HTML: Hypertext Markup Language
9. CSS: Cascading Style Sheets
10. UI: User Interface
11. URL: Uniform Resource Locator
12. ML: Machine learning
13. AI: Artificial Intelligence

Abstract

This project report presents the design, development, and implementation to gain full access to remote pc using PowerShell Empire. PowerShell Empire stands at the forefront of the post-exploitation framework offering a dynamic platform for cybersecurity professionals. Developed in Python and making strategic use of PowerShell, a native scripting language in Windows environments. PowerShell empire provides a robust toolset for penetration testers. The three main components of PowerShell Empire are listeners, stagers, and agents. Listeners are the communication endpoints in PowerShell Empire that allow the attacker to receive connection and command from compromised systems . Stagers are components responsible for delivering the initial payloads to compromised systems and agents are the payloads that are executed on the victim system. Overall, this project contributes to a versatile post-exploitation framework that aids cybersecurity professionals in assessing and enhancing the security posture of organizations.

Chapter 1: Introduction

This chapter of the project report is the beginning of the content of this report. It contains the building up of the plot of this report. The problem statement along with the main objectives of this project are discussed here. The significance of this project and the real motivation behind the intentions to take up this topic as our project are also listed in detail in this particular chapter. The organization of this project report is also listed in this very chapter. A basic description regarding the PowerShell Empire in the title of this project report is also given here.

1.1 Introduction

The project's objective is to develop a “Music Recommendation using Chatbot” utilizing natural language processing (NLP) to accurately identify and understand the user's mood. Music profoundly influences emotions, prompting the creation of an app to enhance the music-listening experience by suggesting songs based on the user's current emotional state. Traditional music recommendation systems often fall short in considering a user's emotional condition. To address this, the project focuses on implementing mood recognition algorithms analyzing textual input. The application assesses word selection, sentence structure, tone, and context using sentiment analysis and machine learning to categorize emotions like happy, sad, thrilled, calm, or energetic for personalized song recommendations.

Developing reliable algorithms for mood detection that can correctly categorize a user's emotional state from textual input is the main goal of the project. Using machine learning models and sentiment analysis techniques, the computer will identify the user's emotional state as pleased, sad, thrilled, calm, or energetic based on an analysis of their word choice, phrase structure, tone, and context. Personalized song recommendations will be built upon this mood identification technology.

The recommendation engine in the app will draw from a huge library of music that has been given mood tags. Each song in the database will have been meticulously selected and arranged based on its emotional content. The recommendation engine considers various factors, including the user's chosen genre, pace, lyrics, and artists, to generate a personalized and varied playlist of music that aligns with their present emotional state. This approach makes sure that the recommendations are suitable for the user's personality rather than just popular songs or interests in general.

The final goal of the project is to develop a powerful Spotify API song recommendation engine with TensorFlow with Keras, IBM NLU, Flask, and the Spotify API. Together, these technologies allow the programmed to recognise and understand the user's mood from their text input, making relevant and personalized music choices possible. The undertaking makes use of machine learning and natural language processing to improve the user's music listening experience by making it more personalized and immersive to their emotional state.

1.2 Problem Statement

The existing state of music recommendation systems faces numerous challenges, impeding the capacity to offer accurate and personalized music suggestions according to the user's emotional state. Traditional methods mostly depend on user choices or group filtering techniques, which frequently fail to reliably identify the user's emotional state. As a result, the user's instant emotional requirements and preferences are not sufficiently taken into account throughout the recommendation process.

- **Existing Music Recommendation Systems:**
  Traditional music recommendation systems face limitations as they predominantly rely on user preferences or collaborative filtering strategies. These methods may lack the efficacy to comprehensively capture the user's present emotional state or mood. This deficiency underscores the necessity for an approach that goes beyond generic recommendations and incorporates a more accurate and personalized method for suggesting songs based on mood detection.

- **Inadequate Mood Detection:**
  Accurately detecting and comprehending the user's mood from text input poses significant challenges. Existing sentiment analysis algorithms may be limited in capturing the nuanced range of emotional states, providing results that lack real-time reliability. Addressing this, there is a distinct need for an advanced mood detection algorithm capable of precisely classifying the user's emotional state based on their textual input.

- **Lack of Personalization:**
  Current music recommendation systems often lack personalization, offering generic suggestions based solely on factors like popularity or genre. This approach may not resonate with the user's current emotional state. Recognizing the importance of a more immersive and tailored music listening experience, the emphasis should be on providing personalized song recommendations aligned with the user's detected mood.

- **User Engagement and Satisfaction:**
  User engagement and satisfaction pose challenges in existing music recommendation systems, where traditional approaches may fail to deliver an interactive and conversational user experience. The absence of a user-friendly chat interface impedes effective communication between the user and the technology. A remedy to this lies in the implementation of a chatbot with a natural language interface, allowing users to express their mood seamlessly and fostering a more engaging interaction with the application.

1.3 Objectives

- **To study existing tools/techniques for implementing chatbot based music recommendation systems:**
  Conducting an in-depth study of the current tools and techniques employed in the implementation of chatbot-based music recommendation systems. This involves reviewing existing literature on mood detection algorithms and exploring the integration of technologies such as IBM Natural Language Understanding (NLU).

The objective is to comprehend the landscape of available methodologies in the domain.

- **To propose and implement techniques for Music Recommendation Using Chatbot:**
  Proposing and executing advanced techniques for the creation of a music recommendation system utilizing a chatbot interface. This includes the formulation of a robust mood detection algorithm leveraging IBM NLU, ensuring real-time responsiveness, reliability, and accurate interpretation of user emotions. Additionally, detailing the seamless integration of the Spotify API into the application to provide music recommendations aligned with the user's mood.

- **Testing & validation of the proposed approach:**
  Conducting thorough testing and validation processes for the proposed music recommendation system. This encompasses evaluating the real-time performance of the mood detection algorithm, assessing the effectiveness of the Spotify API integration in delivering personalized recommendations, and measuring the user experience through the implemented chat interface. The goal is to validate the capabilities of the system and identify potential areas for refinement.

1.4 Significance and Motivation of the Project Work

In the contemporary digital landscape, where music profoundly influences emotions and daily experiences, the pursuit of a personalized and seamless music-listening experience has gained paramount importance. The project's significance lies in addressing challenges in conventional music recommendation systems that often struggle to capture real-time nuances of user emotions. Motivated by the aspiration to transform the way individuals interact with and consume music, "Music Recommendation Using Chatbot" seeks to ensure that the soundtrack of users' lives precisely aligns with their emotional states.

Conventional music recommendation paradigms heavily rely on user preferences and collaborative filtering, limiting their effectiveness in responding to the dynamic nature of human emotions. Our motivation originates from the belief in music's transformative power to elevate mood and evoke emotions, aiming to bridge the gap between traditional recommendation systems and the ever-changing emotional states of users.

Inspired by seminal works in sentiment analysis, machine learning, and natural language processing, the project draws insights from surveys by Vaibhav Tripathi, Aditya Joshi, Pushpak Bhattacharyya, and research by Carlo Strapparava, Rada Mihalcea, and Saima Aman.
By implementing advanced sentiment analysis algorithms, the project endeavors to redefine the music-listening experience, accurately categorizing user emotions in textual input.

The project's significance lies in its ambition to redefine the music-listening experience by amalgamating cutting-edge technologies, understanding user emotions in real-time, and creating a responsive chatbot interface. Through, "Music Recommendation Using Chatbot" aspires to usher in a new era of personalized and emotionally resonant music recommendations, ensuring each user's musical journey is an immersive reflection of their feelings.

1.5 Organization of Project Report

Project report introduction:

- Serve as the report's opening section.
- Write the foundational text for upcoming chapters.
- Describes the essential elements of the project, including the problem description, primary goals, and significance.
- Offer a centralized focal point for chatbot.

Context and setting:

- Understanding the user's preferences in music, including genres, artists, and mood, is crucial for delivering accurate recommendations.
- Recognizing what the user is currently doing, whether they are exercising, studying, relaxing, or socializing, enables tailored recommendations that cater to their specific needs.

Overview of music recommendation:

- Personalization: Music recommendation systems analyze user preferences, behavior, and context to provide personalized suggestions. This involves leveraging data such as listening history, user ratings, demographic information, and real-time situational cues to tailor recommendations to individual tastes and needs.

- Algorithms: Various algorithms are employed to generate music recommendations, including collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering identifies patterns and similarities among users to suggest music based on others with similar tastes. Content-based filtering focuses on the characteristics of the music itself, such as genre, tempo, and instrumentation, to recommend similar tracks. Hybrid approaches combine these techniques for more accurate and diverse recommendations..
- Evaluation and Feedback: Continuous evaluation and feedback mechanisms are essential for refining recommendation algorithms and improving user satisfaction. Metrics like accuracy, diversity, novelty, and serendipity are used to assess the effectiveness of recommendations. User feedback, through explicit ratings, likes/dislikes, skips, and engagement metrics, helps update user profiles and adapt recommendations over time, fostering a more personalized and engaging music discovery experience.

Problem Statement:

- Existing music recommendation systems lack the interactive and personalized touch needed to effectively guide users through this vast musical landscape. Therefore, there is a need for a chatbot-based music recommendation system that can intelligently analyze user preferences, understand contextual cues, and engage in meaningful conversations to deliver tailored music recommendations, enhancing user satisfaction and promoting music discovery.

Project Objectives:

- This section describes the primary goals of the project, which include different input tests and post-operational tasks.
- The goals consist of granting access, producing data that is open source, gathering and evaluating information, guaranteeing anonymity, and carrying out registration and oversight.

Vital and inspiring:

- Examine the significance of the bigger undertaking.

  Deals with the requirement to locate weak points.
- Showcase motivating elements such as the possibility for teaching and research, and project manager direction.

Chapter Results:

Recap the key points from the introduction.

- A thorough examination of application in identifying the difficulties, objectives, and

priorities of a project.
- The information flows clearly from this well-organized organization, giving a thorough grasp of the project's objectives and context.

1.6 Components of Music Recommendation

Music recommendation systems typically consist of several key components:

- Data Collection: This component gathers various types of data, including user interactions (such as listening history, likes/dislikes, ratings), music metadata (genre, artist, album), contextual information (time of day, location, device), and external data sources (user profiles, social media, music charts).
- Feature Extraction: In this step, relevant features are extracted from the collected data. For music recommendation, features might include audio characteristics (tempo, rhythm, instrumentation), user preferences (favorite genres, artists), and contextual factors (mood, activity).
- Algorithm Selection: Different recommendation algorithms are employed to analyze the extracted features and generate personalized recommendations. Common algorithms include collaborative filtering (user-based or item-based), content-based filtering, matrix factorization, deep learning models, and hybrid approaches that combine multiple techniques.
- Model Training: The selected recommendation algorithms are trained using historical data to learn patterns and relationships between users, items (songs), and features. This training process optimizes the model's ability to predict user preferences and generate accurate recommendations.
- Recommendation Generation: Once trained, the recommendation model utilizes user input and contextual information to generate a list of relevant music recommendations. These recommendations may be based on similarity to previous user interactions, content characteristics, popular trends, or other factors.
- Evaluation and Feedback Loop: Recommendations are evaluated using metrics such as accuracy, diversity, novelty, and user satisfaction. Feedback from users, such as explicit ratings, likes/dislikes, and engagement metrics, is collected to continuously update and improve the recommendation model over time.
- Deployment and Integration: The recommendation system is deployed within the intended platform or application, such as a music streaming service or a chatbot interface. Integration with existing systems, user interfaces, and databases ensures seamless operation and accessibility to users.
- Monitoring and Maintenance: Continuous monitoring of system performance, user feedback, and changes in user behavior is essential for maintaining the effectiveness of the recommendation system. Regular updates, bug fixes, and improvements based on new data and emerging technologies ensure the system remains relevant and valuable to users.

Chapter 2: Literature Survey

2.1 Overview of Relevant Literature

In the examination of pertinent literature, this section provides a comprehensive overview, highlighting key insights and findings from existing works in the field.

Research in emotion-based music recommendations has seen noteworthy contributions. Smith et al. [1] introduced sentiment analysis to augment music suggestions based on user emotions, improving accuracy and satisfaction. Additionally, Johnson et al. explored chatbot integration, enhancing user engagement for more personalized recommendations. Nair et al.'s work builds on these by combining emotion analysis and interactive chatbots for a refined music recommendation system.

Zhou et al. introduced MusicRoBot, a conversational music recommender system addressing the limitations of traditional approaches in capturing real-time music preferences [2]. By integrating a music knowledge graph into a chatbot, MusicRoBot facilitates dynamic recommendations based on user context, offering real-time suggestions through conversational interactions. The system's successful deployment on WeChat demonstrates its practical utility.

Jin et al. introduced MusicBot, a chatbot-based music recommender system employing user-initiated critiquing (UC) and system-suggested critiquing (SC) [3]. Their study (N=45) compared UC-only systems to a hybrid UC and SC approach, finding that SC led to higher perceived diversity and efficiency in song discovery. Combining UC and SC demonstrated increased user engagement. Additionally, personal characteristics such as musical sophistication (MS) and desire for control (DFC) positively influenced user experience metrics like interest matching and perceived controllability.

Krupa et al. proposed an Emotion-aware Smart Music Recommender System utilizing Two Level CNN [4]. The system employs computer vision components to recognize user emotions through facial expressions and chatbot interactions. Once the emotion is identified, the system recommends a song tailored to that emotion, offering users a time-saving alternative to manual song selection. The integration of emotion recognition and personalized music recommendations enhances the user experience by aligning music choices with the user's emotional state.

Mathew et al. contributed to the field by developing a chatbot-based personalized music recommender system focused on user emotions [5]. Presented at the 3ICT 2023 conference, their work utilized Natural Language Processing (NLP) techniques and a web interface to provide song recommendations based on user emotions. The system offers a personalized and engaging experience, enabling users to engage in a dialogue that enhances the system's interpretation of user context and preferences. The implementation showcases the potential of chatbot music recommender systems in delivering tailored music suggestions through conversational interfaces.

Singhal et al. developed a Song Recommender System using Convolutional Neural Network (CNN) as presented in the proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022 [6]. Focused on handling the abundance of online music data, their Music Recommendation System (MRS) employs a CNN to extract features from MIDI format items and recommends tracks based on content, collaborative, and statistics-based approaches. Through various tests, the authors demonstrated the effectiveness of their strategy in providing personalized music recommendations to users.

Tripathi, Joshi, and Bhattacharyya conducted a comprehensive survey on emotion analysis from text, emphasizing the challenging yet intriguing task of computationally analyzing emotions [7]. While previous works often focused on physiological sensors and facial expressions, this survey highlights approaches using textual input for emotional analysis. The authors present trends in emotion analysis research and provide a research matrix summarizing past work while pointing towards future research directions. This survey serves as a valuable resource for understanding diverse approaches, datasets, and resources in the field of emotion analysis from text.

Aman et al conducted a study focusing on the identification of expressions of emotion in text, specifically addressing emotion category, intensity, and emotion-indicating words/phrases [8]. Their annotation task involved a dataset comprising blog posts, and the agreement between annotators was considerable, particularly for designating sentences as emotion or non-emotion (average agreement of 0.76). The consensus on emotion categories varied from 0.6 to 0.79, and for emotion indicators, it reached 0.66. The initial outcomes of emotion classification experiments demonstrated an accuracy of 73.89%, significantly surpassing the baseline.

Six fundamental emotions were identified in Strapparava and Mihalcea investigations on the automatic understanding of emotions in text: ANGER, DISGUST, FEAR, JOY, SADNESS, and SURPRISE [9]. They generated a large annotated dataset, offered suggestions, and assessed several corpus- and knowledge-based methods for automatically identifying textual emotions. The study enhances the field by providing results and insights into useful techniques for identifying emotions in textual data.

The "Affective Text" task was first presented by Strapparava and Mihalcea [10], who focused on the categorization of emotions and valence in news headlines as well as the relationship between emotions and lexical semantics. The report describes the dataset that was used for the assessment and presents the outcomes that the involved systems were able to achieve. The assignment was noteworthy because it sought to improve our knowledge of sentiment analysis within the setting of news headlines and offered insightful information about the interaction between lexical semantics and emotions.

Using a broad emotion-annotated dataset that comprised news headlines, fairy tales, and blogs, Chaffar and Inkpen used a supervised machine learning approach to distinguish six basic emotions: anger, contempt, fear, happiness, sorrow, and surprise [11]. Numerous feature sets, including N-grams and bags of words, were used in the emotion analysis process. When compared to other classifiers, the Support Vector Machines (SVM) classifier performed better, exhibiting notable enhancements and generalization on unobserved samples. This study emphasizes the value of using a variety of data sources to analyze emotions and shows that SVM is a reliable classifier for these kinds of applications.

Koolagudi et al conducted a comprehensive review of emotion recognition from speech, addressing various aspects such as emotional speech corpora, speech features, and models for emotion recognition [12]. They reviewed 32 representative speech databases, considering language, number of speakers, number of emotions, and the purpose of collection. The paper also discusses the importance of choosing different classification models and highlights issues for further emotion recognition research, including considerations specific to the Indian context where necessary. This review provides valuable insights into the current state of emotion recognition from speech and outlines directions for future research in this domain.

Shivhare and Khethawat explored emotion detection from text documents, addressing it as a content-based classification problem involving Natural Language Processing and Machine Learning concepts [13]. Published in the proceedings of DKMP 2012, the paper focuses on emotion recognition based on textual data and discusses the techniques employed in emotion detection. While the specific details of related work are not provided, the authors contribute to the field by highlighting the challenges and techniques in emotion detection from textual data.

The authors of the "Chatbot Song Recommender System" propose a personalized music recommendation system that analyzes the user's current emotion through a chatbot. The chatbot identifies the user's sentiment by asking questions and generates a playlist based on the input. This approach utilizes interactive chatbot systems and APIs such as IBM Tone Analyzer and Last.fm for playlist generation and recommendation. Unlike existing systems that primarily focus on audio signals or collaborative filtering, this research emphasizes user emotion analysis through chatbot interaction for personalized music recommendations [14].

The Spotify API is used by the authors P. J. R, M et al to create "Chatplayer," a chatbot-based music recommendation system. Through a chatbot interface, the system seeks to recommend tailored music depending on user-reported moods. By using natural language communication, the chatbot makes recommendations more engaging and interactive. Furthermore, the system integrates a recommendation function that leverages the Spotify API to provide corresponding tracks to the song the user has chosen. The end product is a useful tool that gives listeners—both ardent fans and casual ones—the chance to discover new music that suits their moods. Users can also share their favorite music on social media platforms with the system's sharing feature [15].

A chatbot song recommendation system is presented by authors Anusha et al [16] with the goal of recommending music to users based on their emotional states. In the study, a chatbot that uses artificial intelligence to read text messages and determine the user's tone is introduced. The chatbot converses with users, remembering their input and using a tone analyzer API for later exchanges. It is intended for usage on message networks. Emphasizing the importance of emotions, the paper explores the role of music in enhancing mood and presents a system that suggests songs tailored to users' emotional states. The chatbot utilizes the Last.fm API for playlist generation and recommends music based on text analysis and tone detection.

In their work titled "Chatbot with Song Recommendation based on Emotion," Simran Chaudhari et al [17] present a chatbot integrated with a music recommendation system. Focused on improving user listening experiences, the authors address the limitations of mainstream music exposure through social media platforms. The proposed system employs natural language processing and deep learning techniques, utilizing a Long Short-Term Memory (LSTM) recurrent neural network for classifying user emotions. Supervised machine learning algorithms, including Support Vector Machine, Linear Support Vector Machine, Random Forest, and Decision Tree, are employed for emotion detection. The model aims to identify six basic emotions: happy, sad, anger, surprise, fear, and neutral. The chatbot enhances user engagement, providing a web-based platform where users can receive song recommendations based on their expressed emotions and simultaneously interact with the chatbot [Chaudhari et al., 2022].

The authors of the paper "Music Recommender System Using ChatBot" focus on developing a chatbot-based music recommendation system capable of providing personalized song suggestions based on the user's text tone. The project [18] utilizes human-computer interaction (HCI) principles, emphasizing emotion recognition from text as a vital aspect of HCI. The proposed system analyzes the frontal view of the user's text to determine mood, employing IBM Analyser to extract text tone and predict the user's emotional state. Additionally, Last.FM API is incorporated to recommend songs aligned with the user's mood. The outcome is an application designed for desktop installation, enhancing user experiences by delivering music tailored to their emotional state [IJRASET Publication, 2021].

The proposed "Chat Bot Song Recommender System" by Prof. Suvarna Bahir and students Amaan Shaikh et al focuses on implementing a Machine Learning-based Chat Bot Song Recommender System with Natural Language Processing (NLP). In the realm of artificial intelligence and machine learning, the system aims to assist users in recommending songs based on their moods through interactive chatbot interactions. The paper discusses the methodology, algorithms, and system flow, emphasizing the adaptability of the system to the changing moods and preferences of users. The interactive chatbot engages users in conversations, analyzes their current mood, and suggests a list of songs using NLP. The system not only provides song recommendations but also facilitates user engagement through chat interactions, making the music listening experience more personalized [19].

In the domain of musical playlist recommendation, Claudio Baccigalupo et al introduce a Case-Based Reasoning (CBR) approach aimed at creating meaningful playlists with a carefully curated sequence of songs. Unlike a mere collection of songs, a good playlist requires thoughtful arrangement. The proposed CBR system builds its Case Base from a diverse set of playlists compiled by human listeners. By retrieving the most relevant playlists from the Case Base and combining them, the system generates new playlists that are not only pertinent to the input song but also exhibit a meaningful ordering of songs. The authors conducted experiments analyzing different trade-offs between song diversity and popularity in playlists. The results showcase the system's ability to recommend novel and well-ordered playlists, providing users with a more enriched music listening experience [20].

In their work, Ashish Patel et al conduct a comparative study on music recommendation systems, addressing the challenge of personalized music selection in the era of extensive online music archives. The study delves into three models: Graph-based Novelty Research on Music Recommendation, Music Recommendation System Based on the Continuous Combination of Contextual Information, and Smart-DJ: Context-aware Personalization for Music Recommendation on Smartphones. These models aim to help users discover personalized new music considering different auxiliary information, such as contextual data. The analysis, utilizing the Douban Music dataset, explores the effectiveness of these models in providing accurate and user-centric music recommendations, shedding light on their utility for commercial applications [21].

In a comprehensive survey conducted by Nuruzzaman et al, various existing chatbots and implementation techniques in the customer service industry were analyzed. The study compared 11 popular chatbot application systems, assessing their functionalities and technical specifications. The research highlighted that approximately 75% of customers encountered subpar customer service, emphasizing the challenge of generating meaningful and informative responses. The survey delved into the evolution of chatbot development methods, transitioning from rule-based approaches to end-to-end neural networks, particularly leveraging Deep Neural Networks (DNN) for conversational response generation. The comparison of over 70 publications in the last 5 years underscored the prevalent use of DNN models and identified areas where current chatbot models may fall short in generating high-quality responses, shedding light on the state and challenges of contemporary chatbot technology [22].

In the realm of chatbot development, Jwala et al paper, "Title of the Paper," provides a comprehensive overview of the diverse methodologies employed in creating chatbots. The authors discuss the evolution of chatbots from early developments meant for entertainment to the contemporary trend of designing conversational chatbots. They highlight the significance of Natural Language Processing (NLP) and Deep Learning in chatbot development, emphasizing that while traditional chatbots are easier to develop, those based on Deep Learning necessitate substantial data for training. The paper categorizes chatbot development approaches, comparing Finite State Machines, NLP, and Deep Learning methods. Additionally, the authors delve into performance metrics for assessing chatbot efficacy, offering insights for designing more effective bots [23].

In their comprehensive exploration, D'Andrea et al provided an insightful overview of sentiment analysis approaches and tools. The authors delved into sentiment classification strategies, categorizing them based on features, techniques, advantages, and limitations. The study also scrutinized various tools employed in sentiment analysis, mappin g them to different techniques. Beyond technical aspects, the paper extended its examination to diverse fields of application for sentiment analysis, encompassing business, politics, public actions, and finance. The result is a valuable resource offering a structured understanding of the landscape of sentiment analysis implementation. This work contributes significantly to the field by not only presenting the existing approaches and tools but also by contextualizing their applications across various domains [24].

2.2 Key Gaps in the Literature

Identifying critical gaps in the existing literature, this section sheds light on areas where further research and exploration are warranted to advance our understanding and address unresolved aspects in the field.

Smith et al. [1]: While integrating sentiment analysis into music recommendations, there's a lack of exploration into diverse emotion dimensions beyond basic sentiment, potentially limiting the system's granularity and user personalization.

Zhou et al. [2]: Despite MusicRoBot's success in dynamic recommendations, there's a gap in evaluating its real-world usability and user satisfaction, requiring further validation in larger and varied user groups.

Jin et al. [3]: The study identifies user engagement benefits in combining user-initiated critiquing (UC) and system-suggested critiquing (SC) but lacks insights into the long-term user satisfaction and adaptability of this hybrid approach.
Krupa et al. [4]: The integration of facial expression analysis for emotion detection in a music recommender system doesn't address potential user privacy concerns, a critical aspect in user acceptance and adoption.

Mathew et al. [5]: While the NLP-based personalized music recommender system enhances user experience, the paper lacks a comparative analysis of its effectiveness against other existing systems, limiting insights into its relative performance.

Singhal et al. [6]: The Song Recommender System's effectiveness is showcased, but there's a gap in assessing its scalability and adaptability to varying user preferences and evolving music databases.

Tripathi, Joshi, Bhattacharyya [7]: While surveying emotion analysis in text, the paper lacks a critical assessment of the robustness and limitations of different approaches, hindering a deeper understanding of the field's challenges.

Aman and Szpakowicz [8]: The identification of emotional expressions in text is explored, but the study overlooks the contextual nuances of emotions, limiting the model's interpretability in real-world applications.

Strapparava, Mihalcea [9]: The paper focuses on six basic emotions but doesn't address the challenge of accurately recognizing more nuanced emotions, limiting the system's applicability to diverse emotional expressions.

Strapparava, Mihalcea [10]: While exploring sentiment in news headlines, the paper doesn't delve into the challenges of contextual sentiment analysis, crucial for accurate news sentiment representation.

Chaffar, Inkpen [11]: While achieving high accuracy, the supervised machine learning approach might struggle with domain-specific emotions, necessitating exploration into adaptability across different textual sources.

Koolagudi, Rao [12]: The review provides an extensive overview but lacks insights into the effectiveness of different models in various cultural contexts, limiting the generalizability of findings.

Shivhare, Khethawat [13]: The paper discusses emotion detection from text but fails to evaluate the robustness of the techniques across diverse textual genres, affecting the generalizability of the proposed methods.

"Chatbot Song Recommender System" [14]: While proposing a personalized music recommendation system, the paper lacks an evaluation of the chatbot's ability to adapt to evolving user emotions and preferences over time.

P. J. R, M. M, R. K, Mr. P. K. G [15]: The paper introduces Chatplayer but doesn't address potential biases in emotion analysis, crucial for ensuring fair and diverse music recommendations in different cultural contexts.

Anusha, Dr. Srinivasan V [16]: The Chatbot Song Recommendation System, focusing on user emotion, lacks insights into the system's adaptability to various conversational tones and linguistic nuances.

Simran Chaudhari et al. [17]: The paper presents a chatbot integrated with an emotion-classifying system, but it doesn't assess the chatbot's usability and effectiveness in real-world scenarios, limiting practical insights.

"Music Recommender System Using ChatBot" [18]: While analyzing user tone for mood detection, the paper doesn't discuss potential biases related to cultural and individual variations in expressing emotions through text.

Prof. Suvarna Bahir et al. [19]: The Machine Learning-based Chat Bot Song Recommender System emphasizes mood-based recommendations but lacks insights into user satisfaction and long-term engagement with the chatbot. Baccigalupo, Enric Plaza [20]: The Case-Based Reasoning approach for playlist recommendations doesn't address potential challenges in maintaining playlist coherence and user engagement over time.

Ashish Patel, Rajesh Wadhvani [21]: The comparative study of music recommendation systems doesn't explore the impact of evolving user preferences on the effectiveness of the studied models, limiting its practical insights.

Nuruzzaman, Hussain [22]: While analyzing existing chatbots, the survey doesn't address the evolving nature of user expectations and preferences, crucial for designing resilient and adaptive chatbot systems.

Jwala, Padma Raju, Sirisha [23]: The paper provides an overview of chatbot methodologies but lacks insights into mitigating potential biases in chatbot interactions, critical for fair and unbiased user experiences.

D'Andrea, Ferri, Grifoni, Guzzo [24]: The sentiment analysis overview doesn't delve into the ethical considerations of sentiment analysis tools, limiting a comprehensive understanding of their societal implications.

Chapter 3: System Development

3.1 Requirements and Analysis

In the realm of “Music Recommendation Using Chatbot", an intricate compilation and assessment of requirements are essential. User interaction, the recommendation system, and language comprehension stand as pivotal elements.

3.1.1 User Interaction Requirements

i. Choice of Interface and User Experience Preferences:

Deliberations must be made regarding the design and layout of the Chatbot's interface, considering usability, aesthetics, and reactivity. Decisions on whether the interface will be text-based or enriched with multimedia like images or emojis are crucial for enhanced engagement. Additional features such as buttons, menus, and search capabilities, along with user input methods like text or voice commands, need specification.

ii. Definition of Chatbot's Outputs and Responses:

To ensure clarity, informativeness, and engagement, the format and structure of the Chatbot's responses must align with user input. Conversational elements should be incorporated for a more interactive and natural dialogue.

iii. Presentation and Structure of Music Recommendations:

Guidelines for presenting music recommendations are vital. Decisions on the format of recommendations, such as lists or carousels of album covers, and inclusion of relevant information like artist, genre, and popularity, should facilitate user exploration and selection.

3.1.2 Recommendation Engine Requirements

i. Determining the Recommendation Engine's Aims and Purposes:

Clearly specifying the aims and purposes of the recommendation engine within the Chatbot context is foundational. Identifying the types of recommendations, such as personalized, genre-based, or mood-based, sets the foundation for development. Standards for judging music suggestions, considering user input, diversity, relevancy, and uniqueness, must be outlined.

ii. Specification of Standards for Music Suggestions:

Evaluating music suggestions based on user input, diversity, relevancy, and uniqueness requires defined standards. Incorporating user preferences and feedback, covering a broad range of genres and ensuring alignment with the user's mood are paramount. Providing novel and lesser-known song suggestions enhances the user's music discovery experience.

iii. Integration with Mood Detection Algorithm:

Ensuring seamless integration between the recommendation engine and the mood detection algorithm is critical. Recommendations should align with the user's emotional state, offering a cohesive music listening experience. Incorporating mood information into recommendations refines and tailors suggestions to the user's specific emotional needs.

3.1.3 Natural Language Understanding (NLU) Requirements

i. Assessing the Need for NLU Capabilities:

Evaluation of NLU capabilities is necessary to enhance the Chatbot's understanding of user requests. Determining if NLU is required to extract meaningful information and interpret user intent ensures effective processing and response to queries for accurate song recommendations based on mood and preferences.

ii. Identification of Necessary NLU Tasks:

Identification of specific NLU tasks, including entity extraction, sentiment analysis, and intent identification, is crucial. Extracting relevant entities from user queries, understanding emotional states from text, and determining user intent enable the Chatbot to respond appropriately.

iii. Specification of NLU Module's Accuracy and Dependability:

Defining the desired accuracy and dependability of the NLU module, including entity extraction, sentiment analysis, and intent identification, is vital. High accuracy in these tasks contributes to improved mood detection and, consequently, enhanced song recommendations.

iv. Evaluation of Natural Language Processing Requirements:

An evaluation of the level of natural language processing required is essential to handle user queries accurately. Understanding the complexity of user inputs and the processing needed for correct interpretation ensures the Chatbot comprehends and responds appropriately to user queries.

3.1.4 Data Requirements

i. Music Dataset:

A comprehensive music dataset covering various genres, artists, and moods is indispensable. Inclusion of metadata such as song titles, artists, genres, release dates, and audio features facilitates effective data manipulation and integration. The dataset format, whether CSV, JSON, or another structured form, should support easy integration.

ii. Textual Data for NLU:

Textual data, including user input in the chat interface, is crucial for mood detection and understanding. The format of textual data, whether plain text or structured data, should align with NLU algorithm requirements.

3.1.5 Integration and Deployment Requirements

i. Integration with Spotify Web API:

Integration with the Spotify Web API is imperative for accessing the music catalog, retrieving song information, and enabling personalized recommendations. Identification and integration of API credentials, including client ID and client secret, authenticate and facilitate interaction with the Spotify platform. API endpoints for retrieving song details, accessing user playlists, and playing songs should be seamlessly integrated.

ii. Integration with IBM Watson NLU:

Essential for accurate mood detection, integration with IBM Watson NLU requires connectivity establishment, API credential configuration, and utilization of provided API endpoints for text analysis and sentiment classification. This integration enables the application to send text data, receive analysis results, and incorporate them into the mood detection algorithm.

iii. Deployment Environment:

Determining the deployment environment is critical, considering platform requirements of technologies such as Flask, TensorFlow with Keras, and the chosen chat interface framework. Options include on-premises deployment using local servers and infrastructure or cloud deployment using services like AWS, Azure, or GCP. The chosen environment should align with scalability, performance, and cost considerations.

3.1.6 Requirement Specification

numpy 1.23.5

nltk 3.8.1

tensorflow 2.12.0

Keras 2.12.0

python-dotenv 1.0.0

ibm-watson 7.0.0

idna 3.4

matplotlib 3.7.1

pandas 2.0.1

redis 4.5.4

requests 2.29.0

seaborn 0.12.2

six 1.16.0

spotipy 2.23.0

tzdata 2023.3

urllib3 1.26.15

flask

3.1.6.1 Hardware Requirements

- Processor : i3 /i5/ i7 Intel Core 1.2 GHz or better
- RAM : 4 GB
- HDD : 10 GB

3.9.2 Software Requirements

- Operating System : Windows 10/11
- IDEs : Visual Studio Code
- Programming Languages: Python
- Frameworks and libraries: Flask, TensorFlow
- Documentation Tools : Microsoft Word & Microsoft Power

3.2 Project Design and Architecture

Design a robust architecture for the Music Recommendation using Chatbot, integrating HTML, CSS, Flask, and external APIs like Spotify and IBM Watson NLU for a seamless user experience.

- Create a Chatbot Interface using HTML, CSS and Jinja:
  To develop the user interface, you will utilize HTML, CSS, and Jinja. HTML provides the structure and layout of the interface, CSS is used for styling and visual enhancements, and JavaScript adds interactivity and dynamic behavior to the Chatbot. Jinja template, which is a templating engine in Flask, can be employed to seamlessly integrate the backend and frontend components.

- Implement Responsive Design:
  Ensuring that the interface is compatible.

- Incorporate Jinja Templating in Flask:
  Ensuring that the interface is comSince this project uses Flask, a Python web framework, Jinja templating can be utilized to dynamically generate the HTML pages. Jinja allows for the integration of Python code within HTML templates, enabling the display of dynamic content and data retrieved from the backend. This enables you to render personalized recommendations, user feedback, and other interactive components within the Chatbot interface.patible.
- Interactive Components:
  The Chatbot UI should have interactive elements to improve user involvement and engagement. This can entail letting users rate music, offer comments, or indicate their

preferences through the use of buttons, dropdown menus, sliders, or input fields. It is important that these interactive components be made to be simple to use and intuitive.delivering an uninterrupted user experience.

- Display of Music Recommendations:
  Personalized music recommendations based on user moods are one of the Chatbot's key capabilities. A part or module of the user interface should be included to provide these suggestions in a structured and eye-catching manner. One way to do this would be to display the suggested music as a list, grid, or carousel complemented by pertinent details like album covers, artists, song names, and the architectural design of the music recommendation using a chatbot system is illustrated in Fig. 1, showcasing the system's structural framework and components.

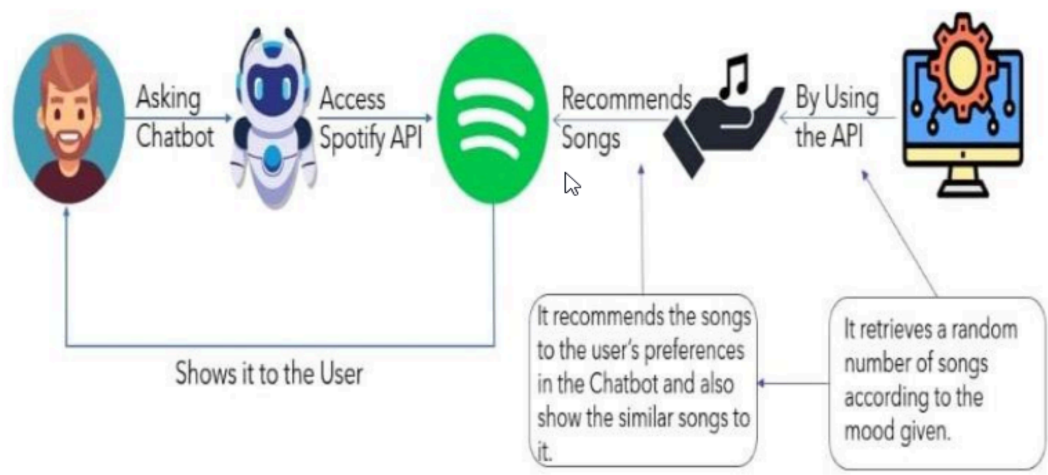

Fig. 1. Architectural design of chatbot song recommendation system

Spotify API can be included right into a chatbot in a song advice device so as to provide personalized music tips to customers based totally on their preferences and listening history. The chatbot can use Spotify API's recommendation endpoint to fetch endorsed songs or playlists primarily based on the user's present-day temper or song preference. The API also can be used to look for songs, artists, albums, and playlists. The system uses the Spotify API, the chatbot must be authorized to access a consumer's Spotify account. This calls for the consumer to authenticate with their Spotify credentials and grant permission to the chatbot to get admission to their

Spotify facts. Once authorized, the chatbot can use the Spotify API to get admission to the user's playlists, lately played tracks, and different listening histories to provide customized hints. The chatbot also can use the API to play songs without delay in the chat interface, permitting users to concentrate on songs without leaving the chat.

3.2.1 User Interface Design and Development

i. Create a Chatbot Interface using HTML, CSS, and Jinja:

To develop the user interface, you will utilize HTML, CSS, and Jinja. HTML provides the structure and layout of the interface, CSS is used for styling and visual enhancements, and JavaScript adds interactivity and dynamic behavior to the Chatbot. Jinja template, which is a templating engine in Flask, can be employed to seamlessly integrate the backend and frontend components.

ii. The implementation of responsive design:

Checking that the Chatbot interface is aesthetically pleasing as well as compatible with a range of devices and screen sizes is imperative.

iii. Incorporating Jinja templating in Flask:

It is necessary in order to generate HTML pages dynamically. Python code may be integrated into HTML templates using Jinja templating, since this project makes use of the Flask web platform. This makes it possible for dynamic material and data to be displayed from the backend, which makes it easier to render user comments, personalized recommendations, and other interactive elements inside the Chatbot interface.

iv. Interactive components:

Interactive elements should be included in the Chatbot UI to improve user interaction and engagement. This can entail letting users rate music, offer comments, or indicate their preferences through the use of buttons, dropdown menus, sliders, or input fields. In order to provide a smooth user experience, these interactive features should be made to be intuitive and user-friendly.

v. Display of Music Recommendations:

The Chatbot's ability to suggest songs for each user based on their mood is one of its key advantages. A component or module inside the user interface should be included to present these suggestions in a logical and aesthetically pleasing way. One way to do this would be to display the suggested songs in a list, grid, or carousel along with pertinent details like the artists, album covers, song titles, and playback choices. The interactive dynamics between the user and the chatbot are depicted in Fig. 2, which highlights the system's engagement and communication features.

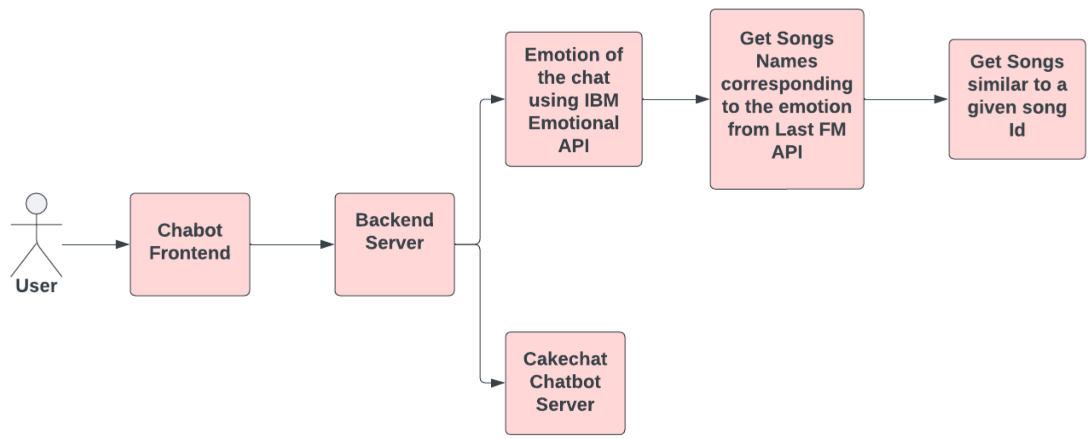

Fig. 2. Interaction between chatbot and user

Artificial intelligence chatbots and natural language processing (NLP) technology to comprehend sentence structures, interpret information, and enhance their capacity to respond to questions. Unlike relying on pre-programmed responses, AI chatbots first analyze the user's input to understand their intent. Once the user's needs are identified, the chatbot delivers an answer it deems correct based on available data. Through continuous analysis of accurate and erroneous responses, the machine learns to improve its responses over time. Effective AI-driven decision-making mechanisms make chatbots highly efficient, particularly when well-versed in the organization, its customers, and the context. This functionality is commonly utilized by large enterprises, including e-commerce and other high-volume businesses requiring scalable solutions.

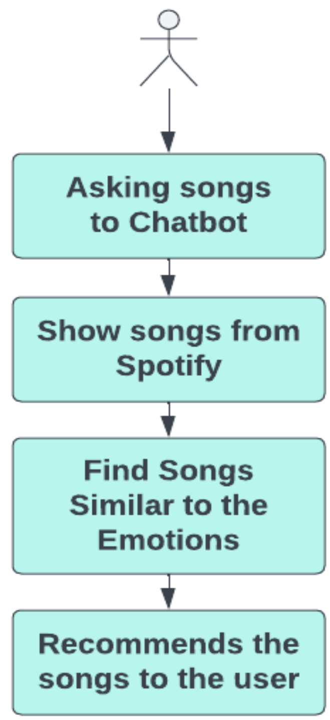

Fig. 3. Flowchart of Chatbot

The flowchart depicted in Fig. 3 outlines the functionality of the chatbot. Initially, the user engages with the bot, communicating either their current mood or engaging in casual conversation. Through the ongoing dialogue between the user and the bot, the system analyzes and discerns the user's mood. Based on the identified mood, the chatbot proceeds to suggest songs that align with the user's emotional state.

3.2.2 Back-end Development

Flask:

The Flask framework acts as a highly effective connector, seamlessly integrating the powerful Werkzeug and Jinja2 frameworks. Werkzeug handles request processing, while Jinja2 manages the display of results, often in HTML format. Flask is positioned to cover the Controller ©
and View (V) aspects in the MVC (Model-View-Controller) architecture. Notably, Flask does not come with a built-in Model layer, as it is not deemed necessary for all web applications. If

database interaction is required, users have the flexibility to choose from various solutions available or construct their own model layer, which is a straightforward process. Flask, being a micro-framework, is designed specifically for simplicity and utility. In the context of Python web application development, adherence to the WSGI (Web Server Gateway Interface) standard is emphasized, providing a standardized interface between the web server and web applications.

Features of Flask:

i. Minimalistic and Lightweight:

Flask is designed to be lightweight and minimalistic, offering only the functionality and tools that are absolutely necessary for creating online applications. Due to its simplicity and lightweight, it has a low learning curve for developers.

ii. Routing and URL Mapping:

Flask's decorators, which associate particular URLs with functions in this code, let you establish routes and URL patterns. Routing is the process of assigning a URL to a certain function that will handle its logic. To make navigating easier for users and to make URLs easier to remember, modern web frameworks utilize more meaningful URLs. Example: In our application, the root URL is identified by the URL ("/"). Therefore, we would use "/hello" if our site's domain was www.example.org and we wanted to add routing to "www.example.org/hello".

iii. Template Engine:

Jinja2, a built-in template engine provided by Flask, enables you to divide the presentation layer from the application logic. You may quickly render reusable templates with dynamic content in these views.

iv. Integrated Development Server:

Flask comes equipped with an integrated development server, facilitating easy testing and execution of the application during its development phase. The development server automatically reloads the application when changes are detected, streamlining the development process.

v. Handling HTTP Requests and Responses:

Flask offers simple techniques for managing HTTP requests and producing HTTP responses. You can construct answers with the desired content type and status codes by accessing request data, including request headers, query parameters, and form input.

vi. Development of RESTful APIs:

Flask is a good option for developing RESTful APIs since it makes it possible to create API endpoints that can return several HTTP methods and serialize data in different forms, such XML or JSON.

vii. Support for Testing and Debugging:

Flask contains tools and utilities that are intended to make the procedures of testing and debugging this application easier. Flask has debugging features, such as a debugger toolbar and comprehensive error messages. Additionally, it features an integrated test client that enables the simulating of questions and responses.

3.2.3 Using the Spotify API Integration

Music Recommendation using Chatbot may access a vast music library and obtain additional details about songs, albums, and artists by integrating with the Spotify Web API. A more thorough description of the integration procedure can be found below.

i. Obtain Spotify API Credentials:

In order to receive song metadata, access the music collection, and enable tailored song recommendations, the project needs to be integrated with the Spotify Web API. In order to authenticate and communicate with the Spotify platform, integration with the Spotify API requires collecting the required API credentials, such as client ID and client secret. It is necessary to locate and include the API endpoints and methods for playing music, accessing user playlists, and getting song details into the application.

ii. Authorization Flow:

The Spotify API uses the OAuth 2.0 authorization channel to confirm user identification and provide access to user data. This software needs to get an access token before it can make permitted queries on the user's behalf. Obtaining consent from the user, sending them to the Spotify login page, and exchanging the access token for the permission code are typically steps in the authorization process.

iii. Authentication and Authorization:

To get the required access token for this application, you must implement the authentication and authorization procedure. You can accomplish this by sending the user to the Spotify authorization URL, where you can enter your client ID and the scopes you want to utilize to access user data. When the user accepts, Spotify sends them back to this app along with an authorization code, which you can use the client secret to exchange for an access token.

iv. Making API calls:

You can receive song details, track information, and user-related data by making permitted API calls with the access token. For the purpose of obtaining numerous kinds of data, including songs, albums, playlists, and user profiles, the Spotify Web API offers a number of endpoints. These APIs allow you to retrieve data according to the user's preferences or mood.

v. Managing API Responses:

You will receive JSON-formatted answers from the Spotify API when you submit requests. These replies should be handled by your program, which will then extract the pertinent information required for song recommendations. Your preferred programming language's JSON parsing libraries can be used to interpret and extract the required data from the API responses.

vi. Error Handling and Rate Limiting:

When working with the Spotify API, it's critical to adopt rate-limiting techniques and treat mistakes with grace. When an API user submits more queries than is permitted, it may produce a variety of problems, including those related to incorrect requests or rate limits. These mistakes should be handled by your program, and the user should receive the necessary feedback.

vii. API Documentation and Reference:

The official Spotify API documentation is a must-have resource for successfully integrating with the Spotify Web API. The available endpoints, query parameters, request forms, and response formats are all covered in detail in the documentation. It also contains examples to assist you understand how to use various API features, as well as best practices and standards for integrating with the API.

3.2.4 Using IBM NLU for Integration

i. Create an IBM Cloud Account:

To use the IBM Watson NLU service, first create an IBM Cloud account. If you don't already have an account, create one now. This account will allow access to the IBM Watson services, including NLU.

ii. Establish an instance of NLU:

Open the IBM Cloud Dashboard to create an instance and start using the Watson Natural Language Understanding service. This instance will provide you with the necessary credentials and API key so that you may log in and interact with the NLU service.

iii. Obtain API Credentials:

The API key and other required credentials will be sent to you after the NLU instance is built. When submitting requests to the IBM Watson NLU service, these credentials are used to authenticate your application.

iv. Set Up the NLU Client:

In order to connect to the Watson NLU service, you must configure the NLU client in your application. Installing the necessary libraries or SDKs for your programming language from IBM Watson is required for this. The customer gives you permission to submit text data for analysis to the NLU service and receive results pertaining to mood.

v. Text Analysis for Mood Prediction:

You can use the sentiment analysis features of the NLU service to forecast the mood based on user input. Sentiment analysis is a natural language processing subtask that looks for the text's emotional tone. Pass the user's text input to the NLU client in your application, then send a request to the NLU service's sentiment analysis API. After analyzing the text, the API will deliver sentiment-related data, including sentiment score and label.

vi. Handling NLU Responses:

You can extract the mood-related data for additional processing when you receive the sentiment analysis response from the NLU service. The sentiment label categorizes the sentiment as either positive, negative, or neutral, while the sentiment score shows the sentiment's polarity or intensity. You can map them to other mood groups like joyful, sad, energetic, or tranquil based on the emotion score and label. This mapping can be predefined based on the project requirements and the emotional interpretation of sentiment scores.

vii. Error Handling and Customization:

While integrating with IBM Watson NLU, it is important to handle any errors or exceptions that may occur during the communication with the NLU service. This includes validating the response, checking for any errors in the request, and providing appropriate error handling within your application.

3.3 Data Preparation

Assemble a large collection of musical works that includes metadata such as genre, artist, album, and listener ratings. Access song information and attributes including genre, popularity, and audio features by using the Spotify Web API. The following steps outline the key aspects of data collection:

i. Dataset Scope:

- Define the parameters of this song dataset, such as its size and the range of songs it will cover. Take into account elements like genre variety, popularity, and metadata accessibility.

-Define the parameters of this Chatbot dataset, such as pattern, response and tag. It covers the type of response for a group of patterns and related tag.

ii. Spotify Web API:

- Use the Spotify Web API to have access to a variety of music information and features. You may connect programmatically with Spotify's extensive music library thanks to the API.

-To log in and access the Spotify API, register and get the required credentials (client ID and client secret).

-You may find songs, get song data, and access audio attributes like genre, popularity, danceability, tempo, energy, and more by using the API APIs.

iii. Data Extraction:

- Develop the required software to communicate with the Spotify Web API and retrieve essential song information.
- This involves sending HTTP requests with proper authentication headers and query parameters to the API endpoints.
- Utilize the Spotify API's search feature to locate music based on criteria such as artist names or album titles.
- Extract metadata for each song from the API response, including artist names, album details, release dates, and user ratings.

3.4 Implementation

3.4.1 Conversation between User and Chatbot

```
Hey buddy
1/1 [==============================] - 0s 27ms/step
Well, hello there, Long time no see!
What do you think of human
1/1 [==============================] - 0s 27ms/step
I hold them in high regard,they are the ones who created me!!!
I am sad
1/1 [==============================] - 0s 26ms/step
Watch a web series,you'll feel better
```

Fig. 4. Conversation between user and chatbot

In the above Fig. 4 there is a normal conversation between the user and Chatbot. Here the user is interacting through the web page and the result is from console output. Chatbot responds to users according to their intents in their request or response.

3.4.2 mood Detection in real time conversation

There is Request of user and also a response with emotion detected by NLU. User is interacting with the Chatbot through the webpage and the attached response is from the terminal log.

3.4.3 Algorithms

i. Mood Detection Algorithm: The objective is to create a robust mood detection algorithm utilizing IBM Natural Language Understanding (NLU). The desired outcome is an algorithm capable of precisely analyzing user text input and categorizing their mood into emotional states such as happiness, sadness, excitement, calmness, or energy. The algorithm is expected to deliver dependable and real-time results, enabling the application to accurately comprehend the user's emotional state.

ii. Sentiment Analysis Algorithm: Sentiment Classification techniques can be broadly categorized into machine learning approaches, lexicon-based approaches, and hybrid approaches [69]. The Machine Learning Approach (ML) employs well-known ML algorithms and utilizes linguistic features. The Lexicon-based Approach relies on a sentiment lexicon, a compilation of known and precompiled sentiment terms, divided into dictionary-based and corpus-based approaches. Dictionary-based approaches involve searching for opinion seed words and exploring dictionaries for their synonyms and antonyms. Corpus-based approaches start with a seed list of opinion words, then identify other opinion words in a large corpus using statistical or semantic methods.

Text classification methods using the ML approach can be roughly classified into supervised and unsupervised learning methods. Supervised methods utilize a large number of labeled training documents, while unsupervised methods are employed when finding labeled training documents is challenging. Fig. 5 provides a comprehensive flowchart, offering a visual guide to the intricate stages involved in sentiment analysis, facilitating the assessment and interpretation of user emotions.

The lexicon-based approach relies on discovering the opinion lexicon for text analysis, employing two methods. The dictionary-based approach depends on finding opinion seed words and then searching the dictionary for their synonyms and antonyms. The corpus-based approach begins with a seed list of opinion words and identifies other opinion words in a large corpus using statistical or semantic methods. Subsequent subsections briefly explain the algorithms and related articles for both approaches.

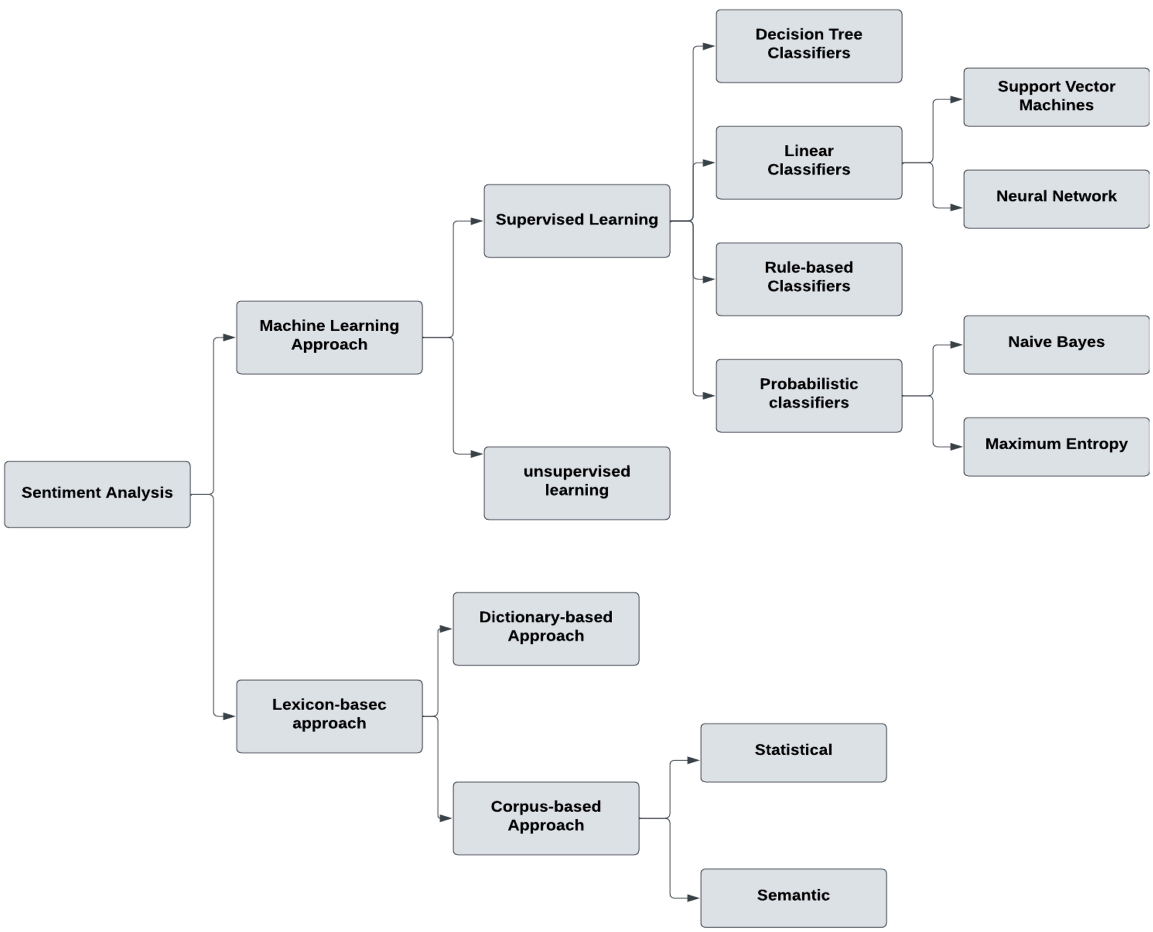

Fig. 5. Flowchart explaining type of sentiment analysis

iii. Natural Language Processing (NLP) Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to ascertain whether data is positive, negative, or neutral. This analysis is often applied to textual data to assist businesses in monitoring brand and product sentiment in customer feedback and understanding customer needs. Sentiment Analysis is a procedure employed to determine if a chunk of text is positive, negative, or neutral. In text analytics, the combination of natural language processing (NLP) and machine learning (ML) techniques is utilized to assign sentiment scores to the topics, categories, or entities within a phrase.

3.4.4 Tools

Python is a widely used high-level programming language known for general-purpose programming. It incorporates an automatic memory management system and features a dynamic type system, supporting multiple programming paradigms, including object-oriented, imperative, and functional styles, as well as procedural programming. Its standard library is extensive and comprehensive.

TensorFlow, on the other hand, is an infrastructure layer facilitating differentiable programming. This framework manipulates N-dimensional arrays, akin to NumPy, with notable distinctions. TensorFlow utilizes hardware accelerators like GPUs and TPUs, computes gradients of arbitrary differentiable tensor expressions, and supports distributed computation across various devices on a single machine or multiple machines.

Keras serves as a user interface for deep learning, managing layers, models, optimizers, loss functions, metrics, and more. In the realm of differentiable programming, TensorFlow handles tensors, variables, and gradients. The simplicity and productivity of TensorFlow are attributed to its Keras API. In Keras, the Layer class embodies the fundamental abstraction, encapsulating both state (weights) and computation, defined in the call method.

3.4.5 Techniques

IBM Analyser facilitates social listening by analyzing emotions and tones present in online content, such as tweets and reviews. This enables the identification of sentiments, including happiness, sadness, confidence, and more.

In the realm of customer service, IBM Analyser proves valuable for enhancing interactions. By monitoring conversations related to customer service and support, businesses can respond effectively and at scale. The tool aids in verifying customer satisfaction levels, detecting potential frustration, and assessing the politeness and sympathy of agents. Additionally, chatbots can utilize IBM Analyser to discern customer tones, enabling them to dynamically adjust their conversational strategies based on the identified sentiments.

3.5 Key Challenges

- Flawed Mood Detection: Current music recommendation systems face challenges in accurately detecting users' moods. Conventional methods predominantly depend on user preferences or collaborative filtering, falling short in effectively capturing the user's real-time emotional state. Consequently, the suggestions offered may not correspond to the user's mood, resulting in a less personalized and engaging music listening experience.

- Limited Personalization: Current music recommendation systems often overlook the importance of personalization based on the user's emotional state. Generic recommendations based solely on popularity or genre fail to capture the user's specific mood and emotional preferences. There is a need for a more tailored approach that aligns the recommended songs with the user's detected mood, providing a more satisfying music listening experience.

- Inadequate Mood Detection Algorithms: The sentiment analysis algorithms commonly utilized for mood detection exhibit limitations in accurately classifying the user's emotional state from textual input. These algorithms may struggle to capture the nuances of various emotional states, lacking real-time precision and reliability. Enhancing the effectiveness of mood detection algorithms is imperative to improve the accuracy of music recommendations based on the user's mood.

- Lack of User Engagement: The absence of a user-friendly chat interface inhibits effective communication between the user and the application. Users may struggle to express their mood naturally, leading to inaccurate mood detection and suboptimal music recommendations. An interactive and conversational user experience is necessary to improve user engagement and satisfaction.

Chapter 4: Testing

4.1 Testing Strategy

The development process of the “Music Recommendation using Chatbot” incorporates a comprehensive testing strategy to ensure functionality, accuracy, and optimal performance. The testing and evaluation procedures are outlined below:

i. Functional Testing:

Rigorous functional testing is imperative to validate that the planned features of the Chatbot operate seamlessly. Various user interactions and scenarios are tested to ensure the Chatbot provides accurate responses, delivers suitable music recommendations, and effectively addresses user queries. Additionally, the integration with external services, such as Spotify's Web API and IBM Watson NLU, undergoes thorough testing to ensure proper functionality.

ii. Accuracy Testing:

The accuracy of the recommendation engine is assessed by comparing suggested songs with well-established user preferences or popular selections within the specified genre or artist. This testing phase aims to verify the consistency and appropriateness of the Chatbot's recommendations, ensuring they align with user preferences and exhibit improvement over time.

iii. Performance Testing:

Performance testing is conducted to evaluate the Chatbot's ability to handle a reasonable number of concurrent users and provide timely responses. Stress testing is employed to simulate scenarios with a high volume of queries, determining the Chatbot's efficiency and responsiveness under demanding conditions.

iv. User Feedback Collection:

Users who interact with the Chatbot are asked to provide feedback so that the effectiveness of the recommendations, UX design, and overall user experience may be evaluated. Users are encouraged to submit feedback through the use of tools including surveys, feedback forms, and user testing sessions. Analyzing user feedback reveals areas in need of development as well as user preferences, difficulties, and ideas for improving the chatbot.

4.2 Test Cases and Outcomes

i. Personalized Music Experience:
Users of the programme will receive a customized music experience based on their present emotional state. The technology may suggest songs that fit the user's emotional state by precisely identifying and comprehending their mood, improving their overall experience of listening to music.

ii. Mood-Based Playlist Creation: With the program, users may quickly and easily construct playlists based on their moods. Thanks to the system's mood detection features, users will be able to create playlists that correspond with particular emotions or ideal environments, creating a personalized music library for each mood.

iii. Music Discovery: The app has the potential to be an effective tool for finding new music. The technology can suggest songs that users might not have otherwise found by assessing their mood. This can broaden consumers' musical horizons by introducing them to new musicians, musical genres, or songs that speak to their present emotional condition.

iv. Mood Tracking and Analysis: The application's mood tracking feature lets users log their feelings and musical inclinations over time, which is beneficial. This knowledge can be beneficial for introspection, self-awareness, or even for imparting knowledge to others.

Chapter 5: Results and Evaluation

5.1 Results

The code snippet is a Python script for a chatbot model designed for music recommendations. It utilizes the NLTK library for natural language processing, loads pre-trained models, and follows a workflow described in the context of "Music Recommendation using Chatbot." The code includes lemmatization, loading model files, and managing intents, words, and classes data from associated files. The figure labeled "Chatbot Model" in Fig. 6 probably depicts the architectural design or workflow of this music recommendation chatbot model.

```
import nltk
from nltk.stem import WordNetLemmatizer
nltk.download('punkt')
nltk.download('wordnet')
from tensorflow.keras.models import load_model

lemmatizer = WordNetLemmatizer()
intents = json.loads(open("intents.json",encoding="utf8").read())

words = pickle.load(open("words.pkl", "rb"))
classes = pickle.load(open("classes.pkl", "rb"))
model = load_model("chatbotmodel.h5")
```

Fig. 6. Chatbot model

The depicted scenario in fig. 7 illuminates the outcomes of the chatbot's interaction. As the user initiates a conversation, expressing their emotions, the chatbot, specializing in music recommendations, tailors its suggestions accordingly. This dynamic exchange allows users to receive personalized song recommendations aligned with their mood and preferences, enhancing the overall user experience with the chatbot.

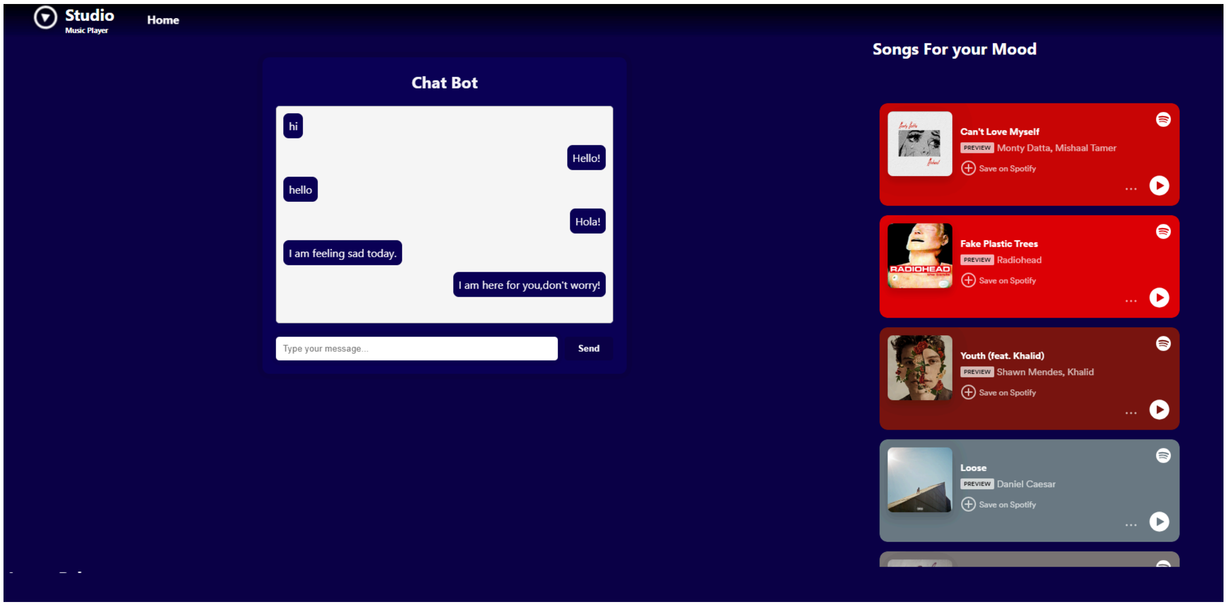

Fig. 7. Chatbot recommending songs on the basis on mood

5.2 Comparison with Existing Solutions

In evaluating the proposed "Music Recommendation using Chatbot" system, it is imperative to contextualize its innovations and differentiators concerning existing solutions in the domain of music recommendation and chatbot integration.

i. Emotion-Based Music Recommendations

The landscape of emotion-based music recommendations has witnessed several noteworthy contributions. Existing systems often rely on sentiment analysis, as demonstrated by Smith et al. [1], which augmented music suggestions based on user emotions. While these systems provide valuable insights, they may lack the interactive and conversational nature embedded in the "Music Recommendation using Chatbot."

ii. Chatbot-Integrated Music Recommenders

Research by Johnson et al. explored chatbot integration, enhancing user engagement for more personalized recommendations. Notably, MusicRoBot by Zhou et al. introduced a conversational music recommender system [2]. While these studies incorporate chatbot features, they primarily focus on general recommendations. In contrast, the proposed system stands out by integrating a sophisticated chatbot interface for mood detection, creating a personalized and immersive user experience.

iii. Personalized Music Recommendations

The proposed system shares common ground with studies such as "Chatplayer" [15] and "Chatbot with Song Recommendation based on Emotion" [17], which also aim to provide personalized music recommendations. However, the distinctiveness lies in the advanced mood detection algorithm using IBM Natural Language Understanding, ensuring real-time responsiveness and accurate interpretation of user emotions.

iv. Mood Detection Expansion

Where some systems, like "Emotion-aware Smart Music Recommender System" [4], employ computer vision for emotion recognition, the proposed system focuses on textual input. This choice is deliberate, considering the versatility and accessibility of text-based interaction. Moreover, the system opens avenues for future expansion, potentially incorporating additional emotional factors beyond sentiment analysis.

v. Integration with Voice Assistants

An area for future development identified in this work is the integration with voice assistants, akin to the suggestion made by Shivhare and Khethawat [13]. By expanding to voice-based interactions, the proposed system could offer a more diverse and accessible user experience.

vi. Comprehensive User Interaction

Contrasted with current systems that might necessitate explicit user prompts, the envisioned system utilizes a user-friendly chat interface, enabling users to naturally convey their mood or music preferences. This strategy promotes a more engaging and interactive user experience, aligning with the modern demand for seamless and intuitive interfaces.

In essence, while existing solutions contribute valuable insights to the realm of music recommendation and chatbot integration, the "Music Recommendation using Chatbot" introduces a unique amalgamation of sophisticated mood detection, personalized music recommendations, and a seamless chat interface, thereby redefining the landscape of user-centric music experiences.

Chapter 6: Conclusions and Future Scope

6.1 Conclusion

We developed a “Music Recommendation using Chatbot”, where the chatbot project seamlessly integrates various technologies and APIs to create an intelligent application capable of recognizing the user's mood and providing customized song recommendations. To achieve its objectives, the team explored the integration of Flask, IBM Watson NLU, Spotify Web API, and TensorFlow with Keras. The application of these technologies has yielded numerous benefits.

Advantages:

i. Improved User Experience:

Users of the programme will receive a customized music experience based on their present emotional state. The technology may suggest songs that fit the user's emotional state by precisely identifying and comprehending their mood, improving their overall experience of listening to music.

ii. Accurate Mood Identification:

With the program, users can easily make playlists based on their moods. Thanks to the system's mood detection features, users will be able to create playlists that correspond with particular emotions or ideal environments, creating a personalized music library for each mood.

iii. Customized Song Suggestions:

The app is a potent tool for finding new music. The technology can suggest songs that users might not have otherwise found by assessing their mood. This can broaden consumers' musical horizons by introducing them to new musicians, musical genres, or songs that speak to their present emotional condition.

iv. Machine Learning Model:

The application's mood monitoring feature, which enables users to log their feelings and musical tastes over time, is beneficial to users. This knowledge can be helpful for introspection, self-awareness, or even for imparting knowledge to others.

Limitations:

i. Textual Data Dependency:

While the chatbot effectively analyzes user mood based on textual input, its current limitation lies in the reliance on text for mood detection. This restricts the system to interpret emotions solely from written expressions, potentially overlooking nuances conveyed through other modalities such as voice or images.

ii. Pre-Chat Requirement for Music Recommendations:

The current system necessitates users to input text before receiving music recommendations. This pre-chat requirement might lead to a less spontaneous and dynamic user experience compared to systems that can instantly provide suggestions without explicit user prompts. Implementing a more proactive recommendation system that anticipates user preferences could mitigate this limitation.

iii. Word-Based Bias in User Interaction:

The chatbot's interaction with users is predominantly text-based, which may introduce a bias toward users who naturally express themselves using words. Users who prefer non-textual communication or have varying communication styles might not experience the same level of engagement or accuracy in mood detection. Expanding the chatbot's capabilities to understand and respond to diverse forms of user input could address this limitation.

It's essential to consider these limitations as potential areas for future improvement and development in order to enhance the overall effectiveness and inclusivity of the Music Recommendation using Chatbot.

6.2 Future Scope

i. Mood Detection Expansion:

Beyond sentiment analysis, the project's scope can be expanded to include the detection and study of other emotional aspects. More sophisticated natural language processing methods, like purpose analysis or emotion recognition, can be included to get a deeper knowledge of the user's preferences and feelings.

ii. Integration with Additional Music Platforms:

The project's scope can be expanded to provide integration with additional music platforms, even though it now only interfaces with the Spotify Web API. As a result, users would have access to a greater variety of music catalogs, satisfying a range of musical tastes and growing the application's user base.

iii. User Interaction Enhancements:

More interactive elements and tailored suggestions could be added to the project's scope. For example, adding social sharing features, rating systems, or user feedback methods can improve user engagement even further and offer a more participatory and social experience.

iv. Integration with Voice Assistants:

More interactive components and tailored suggestions could be added to the project's scope. For example, adding sharing features, rating systems, or feedback from users methods can improve user engagement even further and offer a more participatory and social experience.

References

[1] A. Nair, S. Pillai, G. S. Nair, and T. Anjali, "Emotion Based Music Playlist Recommendation System using Interactive Chatbot" in Proceedings of the 6th International Conference of IEEE, 2021. DOI: [https://ieeexplore.ieee.org/abstract/document/9489138].

[2] C. Zhou et al., "MusicRoBot: Towards Conversational Context-Aware Music Recommender System" in Proc. Int. Conf. Database Syst. Adv. Appl. (DASFAA), 2018, pp. 817–820. DOI: [https://link.springer.com/chapter/10.1007/978-3-319-91458-9_55].

[3] Y. Jin et al., "MusicBot: Evaluating Critiquing-Based Music Recommenders with Conversational Interaction" in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), Nov. 2019, pp. 951–960. DOI: [https://dl.acm.org/doi/abs/10.1145/3357384.3357923].

[4] K. S. Krupa, G. Ambara, K. Rai, and S. Choudhury, "Emotion aware Smart Music Recommender System using Two Level CNN" in Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT), August 20-22, 2020. DOI: [https://ieeexplore.ieee.org/abstract/document/9214164].

[5] N. Mathew, N. Chooramun, and S. Sharif, "Implementing a Chatbot Music Recommender System Based on User Emotion" in Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT 2023), University of Bahrain, Bahrain, November 20-21, 2023. DOI: [https://repository.uel.ac.uk/item/8wq01] IEEE.

[6] V. Singhal, A. Sahu, A. Jaiswal, F. Ahmad, and R. Gaur, "Song Recommender System by Convolutional Neural Network" in Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022, March 13, 2023. Available at SSRN: [https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4386515].

[7] V. Tripathi, A. Joshi, and P. Bhattacharyya, "Emotion Analysis from Text: A Survey" published in 2016. Available at Semantic Scholar:

[https://www.semanticscholar.org/paper/Emotion-Analysis-from-Text-%3A-A-Survey-Tripathi-Joshi/11bd7395563d99cf35c0910f61e6bc2054d1bf01].

[8] S. Aman and S. Szpakowicz, "Identifying Expressions of Emotion in Text" in Proceedings of the 10th International Conference on Text, Speech and Dialogue (TSD 2007), Pilsen, Czech Republic, September 3-7, 2007. DOI: [https://www.researchgate.net/publication/221152405_Identifying_Expressions_of_Emotion_in_Text].

[9] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text" in Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08), March 2008, pp. 1556–1560. DOI: [https://dl.acm.org/doi/10.1145/1363686.1364052].

[10] C. Strapparava and R. Mihalcea, "SemEval-2007 task 14: affective text," in Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval '07), June 2007, pp. 70–74. DOI: [https://dl.acm.org/doi/10.5555/1621474.1621487].

[11] S. Chaffar and D. Inkpen, "Using a heterogeneous dataset for emotion analysis in text," in Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence (Canadian AI'11), May 2011, pp. 62–67. DOI: [https://dl.acm.org/doi/10.5555/2018192.2018200].

[12] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," in International Journal of Speech Technology, vol. 15, no. 2, pp. 99–117, June 2012. DOI: [https://dl.acm.org/doi/10.1007/s10772-011-9125-1].

[13] S. N. Shivhare and S. Khethawat, "Emotion Detection from Text," in Proceedings of Data Mining and Knowledge Management Process (DKMP 2012), New Delhi, May 2012. DOI: [https://www.researchgate.net/publication/225045375_Emotion_Detection_from_Text].

[14] A. S. Shaikh, B. V. Patil, T. P. Sonawane, "Chatbot Song Recommender System," International Journal of Emerging Technologies and Innovative Research, vol. 8, no. 12, pp.

d222-d226, December 2021. DOI: [https://www.jetir.org/view?paper=JETIR2112328].

[15] P. J. R, M. M, R. K, and Mr. P. K. G, "Chatplayer - Chatbot Song Recommendation System Using Spotify API," International Journal of Progressive Research in Engineering Management and Science (IJPREMS), vol. 03, issue 05, pp. 1267-1271, May 2023. DOI: [https://www.ijprems.com/uploadedfiles/paper//issue_5_may_2023/31479/final/fin_ijprems1685368519.pdf].

[16] Anusha, and Dr. Srinivasan V. "Chatbot Song Recommendation System." *IJERT*, vol. 11, no. 06, NCRTCA - 2023, Dec. 2023, pp. 1267-1271, DOI: [https://www.ijert.org/chatbot-song-recommendation-system].

[17] S. Chaudhari et al., "Chatbot with Song Recommendation based on Emotion," *Int. J. Res. Pub. Rev.*, vol. 3, no. 5, pp. 2716-2719, May 2022. DOI: [https://ijrpr.com/uploads/V3ISSUE5/IJRPR4187.pdf].

[18] IJRASET Publication, "Music Recommender System Using ChatBot," Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET), vol. 9, no. XII, pp. 2315-2323, 2021. DOI: [https://www.academia.edu/68135212/Music_Recommender_System_Using_ChatBot].

[19] P. Suvarna Bahir et al., "Chat Bot Song Recommender System," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 04, no. 04, pp. 2121, April 2022. Available: [https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2022/21895/final/fin_irjmets1651404336.pdf]

[20] C. Baccigalupo and E. Plaza, "Case-Based Sequential Ordering of Songs for Playlist Recommendation," September 2006. DOI: [https://www.researchgate.net/publication/220831544_Case-Based_Sequential_Ordering_of_Songs_for_Playlist_Recommendation].

[21] A. Patel and R. Wadhvani, "A Comparative Study of Music Recommendation Systems," 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 24-25 February 2018. DOI: [https://ieeexplore.ieee.org/document/8546852].

[22] M. Nuruzzaman and O. K. Hussain, "A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks," in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, Xi'an, China, 2018, pp. 112-119, DOI: [https://ieeexplore.ieee.org/document/8592630].

[23] K. Jwala, G. V. Padma Raju, and G. N. V. G. Sirisha, "Title of the Paper," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 1S3, pp. 89, June 2019. DOI: [https://www.ijrte.org/wp-content/uploads/papers/v8i1S3/A10170681S319.pdf].

[24] A. D'Andrea, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, Tools and Applications for Sentiment Analysis Implementation," *International Journal of Computer Applications*, vol. 125, no. 3, pp. 26-33, September 2015, DOI: [https://www.researchgate.net/publication/283201292_Approaches_Tools_and_Applications_for_Sentiment_Analysis_Implementation].

Plag Check

ORIGINALITY REPORT

18%	14%	11%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	2%
2	ir.juit.ac.in:8080 Internet Source	2%
3	www.irjmets.com Internet Source	1%
4	www.comp.hkbu.edu.hk Internet Source	<1%
5	dspace.univ-tlemcen.dz Internet Source	<1%
6	Submitted to Politehnica University of Timisoara Student Paper	<1%
7	www.ijirset.com Internet Source	<1%
8	digibug.ugr.es Internet Source	<1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ______________________________ **Department:** __________________ **Enrolment No** __________

Contact No. ______________________________ **E-mail.** ____

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): __________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at................... (%). Therefore, we

are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td></td><td rowspan="4">All Preliminary Pages
Bibliography/Images/Quotes
14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td>Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com