A

PROJECT REPORT

ON

"Voice Based Mood Recognition"

SUBMITTED TO THE SHIVAJI UNIVERSITY, KOLHAPUR
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF DEGREE
OF

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Mr. Sanket S. Naik	**07256**
Mr. Gaurav S. Patil	**07273**
Mr. Shubham P. Patil	**07295**
Mr. Vishwajeet V. Patil	**07304**

PROJECT GROUP NO: B-06

UNDER THE GUIDANCE OF
Prof. R. C. Shikkeri
Assistant Professor,
Department Of Computer Science and Engineering

SWVSM's
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Tatyasaheb Kore Institute of Engineering and Technology, Warananagar
Academic Year 2022-23

SWVSM's

TKIET

SWVSM's

Tatyasaheb Kore Institute of Engineering and Technology, Waranagar

CERTIFICATE

This is to certify that the Project Report entitled ,

"Voice Based Mood Recognition"

Submitted by

Name	Exam Seat No.
Mr. Sanket S. Naik	**7256**
Mr. Gaurav S. Patil	**7273**
Mr. Shubham P. Patil	**7295**
Mr. Vishwajeet V. Patil	**7304**

is a bonafide work carried out and is approved for the partial fulfillment of the requirement of Shivaji University, Kolhapur for the award of Degree of Bachelor of Technology in Computer Science and Engineering. This project work is a record of student's own work, carried out by them under our supervision and guidance during academic year 2022-23.

Prof. R. C. Shikkeri
Project Guide

Prof. R.B. Patil
HOD, CSE

Dr. S. V. Anekar
Principal

…………………………...

(External Examiner)

(Internal Examiner)

Date:-
Place:- Warananagar

ACKNOWLEDGMENT

This space is dedicated to acknowledge all those who have helped in bringing this project to fruition.

We are greatly indebted to our guide **Prof. R. C. Shikkeri** for his unstinted support and valuable suggestions. We are grateful to him not only for the guidance, but also for his unending patience and keeping our spirits high throughout. We express our sincere thanks to our beloved Head of the Department, **Prof. R. B. Patil** and Principal, **Dr. S. V. Anekar** for being source of inspiration and providing us the opportunity to work on this project.

We extend heartfelt thanks to all the teaching and non-teaching staff of the department of Computer Science and Engineering for their assistance and cooperation.

Finally, we would like to thank our parents and friends for their moral support and encouragement throughout our academics.

Student Name	**Exam Seat No.**	**Sign**
Mr. Sanket S. Naik	**7256**	
Mr. Gaurav S. Patil	**7273**	
Mr. Shubham P. Patil	**7295**	
Mr. Vishwajeet V. Patil	**7304**	

Date:

Place: Waranagar

TABLE OF CONTENTS

ABSTRACT

Through all the available senses, humans can sense the emotional state of their communication partner. This emotional detection is natural for humans, but it is very difficult task for computers; although they can easily understand content-based information, accessing the depth behind content is difficult and that's what speech emotion recognition sets out to do. It is a system through which various audio speech files are classified into different emotions such as happy, sad, anger and neutral by computers. Speech emotion recognition can be used in areas such as the medical field or customer call centers. The foundation of modeling began with feature selection. After extracting MFCCs, Chroma, and Mel spectrograms (These are the feature that each audio in constituent of) from the audio files we began assembling models readily available from Sci-kit Learn and other Python packages. This project is a validated multimodal database of emotional speech. The database is gender balanced consisting of professional actors. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions contain calm, happy, sad, angry, and fearful emotions.

Keywords: *MFCC,AudioTextConversion,Spectograms,Senses,Chroma*

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Name
CNN	Convolutional Neural Network
SNN	Sequential Neural Network
MLP	Multilayer perceptron Classifier
ASR	Automatic Speech Recognition
MFCC	Mel Frequency Cepstral Coefficient

Chapter 1
Introduction

1.1 Introduction

In today's digital age, understanding human emotions and sentiments is of paramount importance in various domains, including customer feedback analysis, market research, and social media monitoring. With the rapid advancement of technology, extracting sentiments from textual data has become commonplace. However, a significant amount of information is conveyed through non-verbal cues, such as tone of voice, intonation, and overall audio characteristics. This has led to an increasing interest in audio sentiment analysis, which involves analyzing audio signals to determine the emotional or sentiment content embedded within.

The objective of our project is to develop an audio sentiment analysis system using Python. The system will take audio input from users and automatically analyze the audio to predict the corresponding mood or sentiment expressed. By leveraging machine learning and audio processing techniques, we aim to provide a reliable and efficient solution for sentiment analysis of audio data.

Our system will employ various state-of-the-art methodologies to extract meaningful features from the audio signals, such as pitch, intensity, energy, and spectral characteristics. These features will then be utilized to train a sentiment classifier using machine learning algorithms. The classifier will learn patterns and relationships between the audio features and the corresponding moods or sentiments, enabling it to predict the sentiment of unseen audio inputs.

To ensure the accuracy and effectiveness of our system, we will leverage existing labeled audio datasets for training and evaluation. Additionally, we will implement preprocessing techniques to address challenges like background noise, speaker variations, and other audio artifacts that may affect sentiment analysis performance.

Throughout the project, we will employ Python as our primary programming language due to its rich ecosystem of audio processing, machine learning, and natural language processing libraries. The project will utilize popular libraries such as Librosa for audio processing, Scikit-learn for machine learning tasks, and PyAudio for real-time audio input.

The success of our audio sentiment analysis system holds significant potential in various applications. It can contribute to market research by understanding customer sentiments expressed in audio feedback. It can aid in emotion recognition for speech therapy or mental health monitoring. Furthermore, it can enhance social media monitoring by capturing sentiments conveyed through audio content in podcasts, videos, or voice recordings.

In this report, we will detail the design, development, and evaluation of our audio sentiment analysis system. We will discuss the methodologies employed, the datasets used for training and evaluation, the implementation details, and the performance evaluation results. Additionally, we will outline the challenges faced, future improvements, and the potential applications and impact of our system.

Overall, our project aims to bridge the gap between audio data and sentiment analysis, providing a valuable tool for understanding emotions and sentiments conveyed through audio signals.

1.2 Motivation

The motivation behind our audio sentiment analysis project stems from the increasing significance of understanding human emotions and sentiments in today's interconnected world. While sentiment analysis techniques have been extensively applied to textual data, a vast amount of valuable information remains untapped within audio signals. By developing an accurate and efficient audio sentiment analysis system, we aim to unlock this untapped potential and enable a deeper understanding of sentiments expressed through non-verbal cues.

Audio data encompasses various forms of communication, including phone conversations, podcasts, customer service interactions, and social media audio content. By analyzing the emotions and sentiments conveyed through audio, we can gain valuable insights into customer satisfaction, market trends, mental health patterns, and social dynamics. This knowledge can drive informed decision-making, enhance user experiences, and empower businesses, researchers, and professionals across multiple domains.

Moreover, the proliferation of voice-activated devices and virtual assistants has led to an increased need for sentiment analysis in real-time audio inputs. Understanding user emotions and sentiments expressed through voice interactions can enable personalized responses, intelligent virtual assistants, and enhanced human-computer interaction.

Our project aims to address these needs by developing a robust audio sentiment analysis system. By leveraging machine learning techniques and audio processing methodologies, we seek to provide an accurate and efficient solution for predicting sentiment from audio inputs.

The system will offer valuable insights into the emotional content of audio data, contributing to sentiment analysis research and applications.

1.3 Purpose

The purpose of our audio sentiment analysis project is to develop a sophisticated system that can analyze audio inputs and accurately predict the corresponding mood or sentiment expressed within the audio. By leveraging advanced machine learning techniques and audio processing methodologies, our project aims to unlock the valuable information embedded within audio signals and provide insights into the emotional content conveyed through non-verbal cues.

The primary purpose of our project is to bridge the gap between audio data and sentiment analysis. While sentiment analysis techniques have been widely applied to textual data, the analysis of audio signals remains a relatively unexplored field. By designing and implementing an efficient and accurate audio sentiment analysis system, we strive to tap into the wealth of information contained within audio inputs and enable a deeper understanding of emotions and sentiments expressed through voice.

By successfully achieving our project's purpose, we aim to fulfill several key objectives:

Enhancing Sentiment Analysis Capabilities: Our project seeks to expand the realm of sentiment analysis beyond textual data. By incorporating audio inputs, we aim to provide a more comprehensive and nuanced understanding of emotions and sentiments, allowing for more accurate and meaningful analysis.

Enabling Real-Time Sentiment Analysis: With the growing popularity of voice-activated devices and virtual assistants, the demand for real-time sentiment analysis of audio inputs is on the rise. Our project aims to develop a system that can analyze audio inputs in real-time, allowing for immediate response and tailored interactions based on user sentiment.

Driving Insights and Decision-Making: Sentiment analysis of audio data holds great potential in various domains. By accurately predicting mood or sentiment from audio inputs, our system can provide valuable insights into customer feedback, market trends, mental health patterns, and social dynamics. These insights can inform decision-making processes, enhance user experiences, and drive advancements across multiple industries.

Advancing Research in Audio Analysis: Our project contributes to the advancement of research in audio processing and sentiment analysis. By developing new methodologies, exploring feature extraction techniques, and evaluating machine learning models on audio data, we aim to expand the knowledge base and contribute to the development of more sophisticated audio analysis techniques.

1.4 Problem Statement

The study of emotion has advanced rapidly over the last decade, driven by low-cost smart technologies and broad interest from researchers in neuroscience, psychology, psychiatry, audiology, and computer science. Integral to these studies is the availability of validated and reliable expressions of emotion. To meet these needs, a growing number of emotion stimulus sets have become available. Most sets contain either static facial expressions or voice recordings. Clinically, there is growing recognition for the role of singing in understanding neurological disorders and facilitating rehabilitation.

Yet there are few validated sets of sung emotional expression. To address these needs, we are developing this project, a large validated set of audiovisual speech .

In this project we are going to take input as audio from user then features of that audio will be extracted using SNN and then we will maintain a list which will be according to sorted moods and the mood emoji will be displayed with that we are going to convert audio to text and conversion of any other international language to English.

1.5 Objective

- **Develop a Mood Recognition System:** Create a robust and accurate mood recognition system that analyses user voice inputs, whether recorded or live, and predicts their mood or sentiment with high precision.
- **Implement Speech-to-Text Conversion:** Incorporate a reliable speech-to-text converter to transform user voice inputs into text format, enabling further analysis and processing of the input data.
- **Integrate Language Translator:** Develop a language translator component that can effectively translate speech or text in various international languages to English, facilitating better understanding and analysis of user inputs.
- **Design a Web Application:** Develop a user-friendly web application interface to provide a seamless and intuitive user experience for capturing audio inputs, displaying mood analysis results, and integrating speech-to-text conversion and language translation functionalities.
- **Ensure Compatibility with Web Technologies:** Ensure that the project is compatible with web technologies, allowing for easy deployment, scalability, and accessibility across different web platforms and devices.

- **Evaluate and Optimize Accuracy:** Perform extensive evaluation and optimization of the mood recognition system to achieve high accuracy and reliability in predicting user moods from audio inputs. This may involve training the model on appropriate datasets, fine-tuning machine learning algorithms, and leveraging audio signal processing techniques.
- **Address Limitations for Hardware Integration:** Recognize the limitations of the web development process in terms of hardware integration, and identify potential solutions or workarounds to overcome these boundaries. Consider possible approaches for integrating the system with robotics hardware or IoT devices, ensuring a seamless and effective integration where feasible.
- **Document and Communicate Findings:** Thoroughly document the development process, including methodologies, algorithms, implementation details, and any challenges encountered. Provide clear and concise documentation on how to use the system, including instructions for incorporating the mood recognition system, speech-to-text conversion, and language translation components into other projects or systems.
- **Explore Future Expansion Possibilities:** Investigate opportunities for future expansion and enhancements, such as integrating additional sentiment analysis features, expanding language support, or exploring hardware integration possibilities beyond the scope of the current project.

Chapter 2
Literature Survey

2.1 Existing System

2.1.1 Referred Journal and Conference Paper

Journals:

- Indian Journal of Science and Technology (IJST)
- Journal of Information Technology and Software Engineering (JITSE)
- International Journal of Engineering Research & Technology (IJERT)
  - Sentimental Analysis on Audio and Video by- Monali Yadav1, Shivani Raskar 2, Vishal Waman3, Prof.S.B.Chaudhari4
- International Journal of Advanced Computer Science and Applications (IJACSA)
- International Journal of Creative Research :
  - Title-AUDIO SENTIMENT ANALYSIS
  - Authors- P.Ansar khan,T.sumanth, K.vishnu Vardhan
  - Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Anand, Nagar, Krishnankoil, India

Conference Paper:

- 2017 International Conference on Intelligent Computing and Control (I2C2) Sentiment Analysis on Speaker Specific Speech DataMaghilnan
  By-S, Rajesh Kumar M, Senior IEEE, Member School of Electronic Engineering VIT University Tamil Nadu, India
- International Conference on Intelligent Data Science Technologies and Applications (IDSTA)
- International Conference on Big Data, Machine Learning, and Applications (BigDML)
- IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)

2.1.2 Existing System Application

ASR-Automatic Speech Recognition

Automatic speech recognition (ASR) or speech-to-text system identifies spoken words in speech and converts them to written text ASR systems extract acoustic signal characteristics from speech and determine the words in it by pattern matching. Acoustic and language models are used in developing ASR systems. ASR systems are classified according to the parameters and features they use—vocabulary size, speaker mode. Mathematical Models for Speech Recognition -General-purpose speech recognition systems are based on statistical approaches for modelling both acoustics and language. A crucial issue for acoustic modelling is the selection of the speech units that represent the acoustic and linguistic information for the language.

Customer Feedback Analysis: Companies use audio sentiment analysis to analyze customer feedback collected through call centers or voice-based surveys. The system can automatically categorize customer sentiments as positive, negative, or neutral, allowing companies to gain insights into customer satisfaction and identify areas for improvement.

Market Research: Audio sentiment analysis is used in market research to analyze consumer opinions and sentiments expressed in audio or video reviews, focus groups, or social media videos. The system helps researchers understand consumer preferences, identify emerging trends, and make informed business decisions.

Voice Assistants: Voice assistants like Amazon Alexa, Google Assistant, or Apple Siri incorporate audio sentiment analysis to understand user commands and responses better. By analyzing the tone, emotion, and sentiment of user interactions, voice assistants can tailor their responses and provide more personalized and contextually appropriate information

Voice-based Assistive Technologies: Audio sentiment analysis can be integrated into assistive technologies for individuals with speech or communication disorders. By analyzing the sentiment and emotion in their speech, the system can assist in understanding their needs, emotions, and facilitate more effective communication.

Existing Methodology

Speech sample is first passed through a gender reference database which is maintained for recognition of gender before it gets into the process. Statistical approach is followed taking pitch as feature for gender recognition A lower and upper bound pitch for both male and female samples could be found using the reference database Input human voice sample was first broken down into frames of frame size 16 ms each. This was done for frame level classification in further steps.

For each frame MFCC(Mel Frequency Cepstral Coefficient) was calculated as the main feature for emotion recognition Reference database is maintained which contains the MFCCs of emotions i.e. of Sad, Anger. Neutral and Happy.

MFCC of the frames were compared with the MFCCs stored in reference database and the distance was calculated between the comparable frames Based on the distance of the analysis frame from the reference database, one can classify the frame as anger, happy of normal output is displayed in terms of emotional frame count.

2.1.3 Limitations / Challenges in Existing System

- **Accurate Sentiment Analysis:** Achieving high accuracy in sentiment analysis can be challenging due to the complexity and subjectivity of human emotions. Understanding the nuances and context of speech, sarcasm, or subtle expressions requires advanced natural language processing techniques.
- **Speech Variability:** Different individuals have unique speech patterns, accents, and dialects. Variability in speech characteristics can pose challenges for sentiment analysis systems, as they need to be robust and adaptable to different speech styles and regional variations.
- **Ambient Noise and Background Interference:** Background noise and interference can affect the quality of audio recordings, making it difficult to extract and analyze the sentiment accurately. Noise reduction and speech enhancement techniques are necessary to mitigate these challenges**.**
- **Handling Emotional Complexity:** Sentiment analysis often deals with complex emotions that go beyond simple positive or negative categorizations. Capturing and interpreting more nuanced emotions, such as mixed sentiments or varying intensities, poses a challenge for existing systems.

2.2 Proposed System With Block Diagram

- In the Proposed System, input will be recorded audio (to be uploaded by user) or live audio
- Our desired output will be users accurate mood and text of a speech with that if users wants there is other international language than the English then output will be text in English language.
- Models Used For Prediction:
  1. Sequential NN
  2. MLP Classifier

1.Neural Networks:

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligent, is swiftly gaining popularity in the development of trending system. Neural networks are multi-layer networks of neurons (the blue and magenta nodes in the chart below) that we use to classify things, make predictions, etc. The arrows that connect the dots shows how all the neurons are interconnected and how data travels from the input layer all the way through to the output layer.

Advantages of Neural Network:

- Neural Networks have the ability to learn by themselves and produce the output that is not limited to the input provided to them.
- The input is stored in its own networks instead of a data base; hence the loss of data does not affect its working.
- These networks can learn from examples and apply them when a similar event arises, making them able to work through real-time events.

Even if a neuron is not responding or a piece of information is missing, the network can detect the fault and still produce the output.

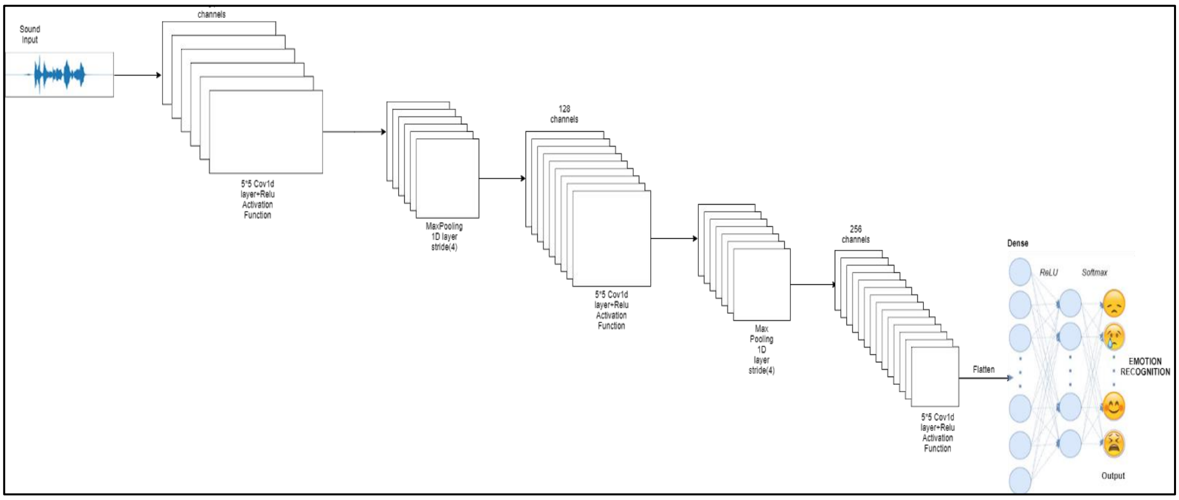


Fig 2.1 : CNN Working in Layers

2.MLP Classifier :

- Neural networks are multi-layer networks of neurons (the blue and magenta nodes in the chart below) that we use to classify things, make predictions, etc.
- MLP Classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier.
- MLP Classifier relies on an underlying Neural Network to perform the task of classification. It helps to convert the input into a more useful output.
- Sigmoid activation function creates an output with values between 0 and 1. There can be other activation functions like Tanh, softmax and REL

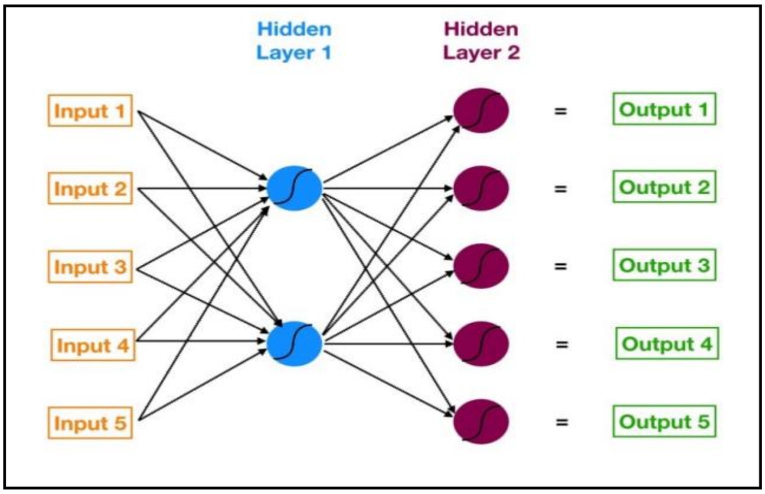


Fig 2.2 : MLP classifier layers

Block Diagram for proposed study:

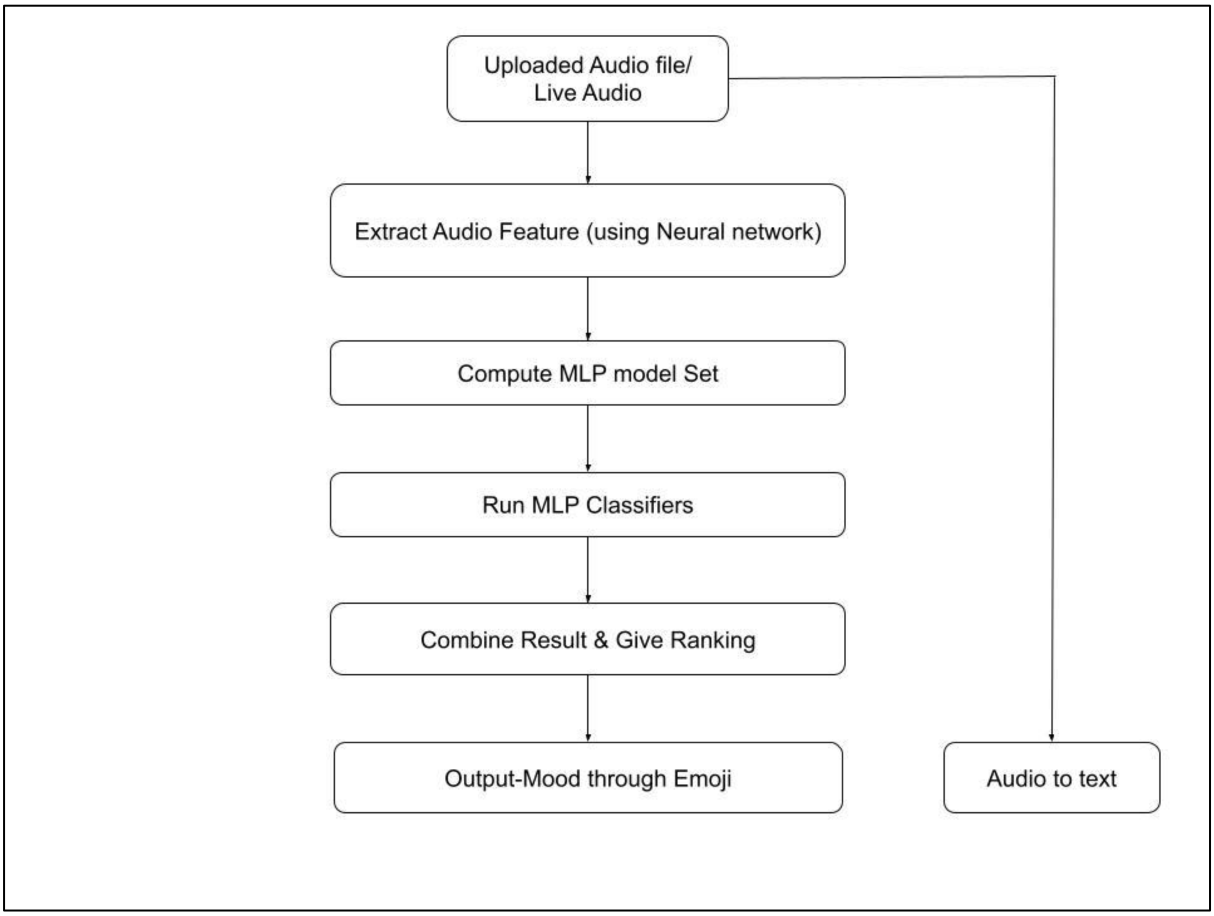

Uploaded Audio file/
Live Audio
Extract Audio Feature (using Neural network)
Compute MLP model Set
Run MLP Classifiers
Combine Result & Give Ranking
Output-Mood through Emoji
Audio to text

Fig 2.3 Block Diagram Showcase Proposed System

User Input (Audio): This is the starting point of the system. The user provides audio input, which can be either recorded audio or live audio.

Audio Sentiment Analysis System: The audio input is passed to the Audio Sentiment Analysis System. This component is responsible for analyzing the audio to determine the user's sentiment or mood. It utilizes machine learning or deep learning techniques to process the audio and make sentiment predictions.

Speech-to-Text Conversion: This component is optional but can be integrated into the system. It converts the audio input into text format. Speech-to-text conversion can help improve the accuracy of sentiment analysis by analyzing the textual content of the speech.

Output (Sentiment Prediction): The final output of the system is the predicted sentiment or mood of the user. It represents the analysis result of the audio input, indicating whether the user's sentiment is positive, negative, or neutral.

2.3 Feasibility Study

A feasibility study was an evaluation of a proposal designed to determine the difficulty in carrying out a design task. Generally, a feasibility study precedes technical development and project implementation.

Client: Doctors(Psychiatric),Robotic industry

Scope: From this project we are going to deliver mood recognition system according to voice of user it can be either recorded or live.

- With that our system includes speech to text convertor and language translator. Language translator will translate any international language to English
- As we are developing our project as purely web application it can be lag behind while integrating with other systems like robotics hardware or any other iot device.
- It only supports Audio in the .WAV format
- As this is purely web development process our system will doesn't concern about any kind of hardware. S while working with hardware systems there will be boundaries to our project.

2.3.1 Technical Feasibility

All necessary technology exists to develop the system. This system is too flexible and it can be expanded further. This system can give guarantees of accuracy, ease of use, reliability and security of Institutional data. This system can give instant responses to inquire. Our project is technically feasible because all the technology needed for our project is readily available.

- The project **Voice Based Mood Recognition** is web application
- The main technologies and tools are associated –

  Markup Language: HTML5

  Style Sheet Language: CSS 2.1

  Scripting Language : Python

  Framework: Python Flask

  IDE: Spyder (version 5.1)

  Diagramming Tools: Lucid Chart
- Each technology is freely available and required technical skills are manageable. Initially website will be hosted on free web hosting space. For later implementations, it will be hosted on a paid hosting space with sufficient bandwidth. Moderate Internet

connection is required for this application, since it does not incorporate any multimedia aspect

- By considering above points we can conclude that our project is technically feasible.

2.3.2 Legal Feasibility

- The project "Voice Based Mood Recognition and Speech to Text Convertor with language translator " is a complete Web Application
- Project "Voice Based Mood Recognition and Speech to Text Convertor with language translator " is absolutely legal and doable
- It meets all legal and ethical requirements as per the Information Technology Act,2000 Gove. Of India
- Project uses freely available/open-source software development tools
- No threats to customer's/institute's/organization's confidential data

From above, it is clear that project " is legally feasible

Chapter 3
Project Scope and Requirement Analysis

3.1 Project Scope

- From this project we are going to deliver mood recognition system according to voice of user it can be either recorded or live.
- With that our system includes speech to text convertor and language translator.
- Language translator will translate any international language to English
- As we are developing our project as purely web application it can be lag behind while integrating with other systems like robotics hardware or any other iot device.
- As this is purely web development process our system will doesn't concern about any kind of hardware. So, while working with hardware systems there will be boundaries to our project.
- **Mood Recognition System:** The primary objective of the project is to develop a mood recognition system that analyses the voice of the user, whether it is recorded or live, and identifies the corresponding mood or sentiment.
- **Speech-to-Text Converter:** The system includes a speech-to-text conversion module that converts the audio input into text format. This feature enables the analysis of the sentiment based on the textual content of the speech.
- **Web Application Development:** The project focuses on the development of a web application, allowing users to access the system through a web browser. The web application provides a user-friendly interface for audio input, sentiment analysis, and displaying the results.
- **Limitation to Web Development**: The project scope is limited to web development, and it may encounter challenges when integrating with other systems like robotics hardware or IoT devices. The system's primary purpose is to function as a web application, and hardware integration falls beyond the project's boundaries.
- **User Interface Design:** The project scope includes designing an intuitive and user-friendly interface for the web application. The interface should allow users to input audio files or use live audio streaming, view the sentiment analysis results, and interact with any additional features such as language translation.

- **Sentiment Analysis Algorithms**: The scope involves implementing and integrating sentiment analysis algorithms within the system. These algorithms should be capable of analysing the audio input and accurately determining the sentiment or mood expressed by the user.
- **Model Training and Evaluation:** As part of the project, there may be a need to train and fine-tune machine learning models or deep learning architectures for sentiment analysis. The scope includes the processes of data preparation, model training, and evaluation to ensure optimal performance.
- **Performance Optimization:** The project scope may involve optimizing the system's performance to handle real-time audio processing and provide quick sentiment analysis results. This optimization can include techniques such as parallelization, model compression, or utilizing efficient libraries.
- **Error Handling and Validation:** The system should incorporate error handling mechanisms to handle unexpected scenarios, such as invalid audio formats, audio quality issues, or errors during speech-to-text conversion. Additionally, appropriate validation techniques should be implemented to ensure accurate results and handle edge cases effectively.
- **Deployment and Hosting:** The scope encompasses deploying the web application on a suitable hosting platform to make it accessible to users. Consideration should be given to factors such as scalability, security, and availability to ensure a reliable and robust deployment.
- **Testing and Quality Assurance:** The project should include a testing phase to validate the functionality, performance, and accuracy of the sentiment analysis system. This may involve unit testing, integration testing, and user acceptance testing to ensure the system meets the desired requirements and quality standards.
- **Documentation and User Guide:** Proper documentation should be prepared to outline the system architecture, algorithms used, installation instructions, usage guidelines, and any relevant information necessary for users and future developers to understand and work with the system effectively.

3.2 Requirement Gathering and Analysis

- Meeting with project guide regarding existing system study and requirements.
- We have also communicated with end users to understand requirements and problem and our frequency of contact was at least once in a week
- Survey –

  For our project we have also conducted survey from which we have collected statistics and studied about existing system and its drawbacks.

  Our survey includes study of existing system, scope of our project (in robotics and Iot) and conversation with end user to study about requirements
- Research papers –
  - ❑ Title – Voice based emotion detection using deep neural networks

    Publisher – IEEE, In 2021 international conference

Requirement Analysis:

Functional Requirements:

- The system should be able to accept audio input from the user, either recorded or live.
- The system should accurately analyze the audio to determine the sentiment or mood of the user.
- The sentiment analysis should categorize the sentiment as positive, negative, or neutral.
- If a speech-to-text conversion component is included, it should accurately convert the audio input into text format.
- If a language translator component is included, it should translate the audio or text input from different international languages to English.

Non-Functional Requirements:

- The system should have a high level of accuracy in sentiment analysis to provide reliable results.
- The system should have low latency and provide real-time or near real-time sentiment analysis, especially for live audio inputs.
- The system should be scalable to handle varying levels of user input and workload.
- The system should be user-friendly and provide clear and concise output or visualization of the sentiment prediction.

- The system should be able to handle and mitigate background noise and interference for accurate sentiment analysis.
- The system should ensure the privacy and security of user audio or text data, adhering to relevant data protection regulations.
- The system should be platform-independent and accessible via a web application, allowing users to access it from different devices and browsers.

Technical Requirements:

- The system should utilize machine learning or deep learning techniques for sentiment analysis.
- If a speech-to-text conversion component is included, it should employ appropriate speech recognition algorithms and libraries.
- If a language translator component is included, it should integrate with suitable language translation APIs or libraries.
- The system should be developed using Python programming language.
- The system should leverage relevant libraries and frameworks for audio processing, sentiment analysis, and web application development.

Constraints:

- The system should be developed within a specified budget and timeline.
- The system should be compatible with standard hardware configurations and operating systems.
- The system should adhere to ethical guidelines and data usage policies

Chapter 4
Project Design and Modeling Details

4.1 Software Requirement Specification(SRS)

Functional Requirement:

- API Calls
- Error Handling
- Information Extraction
- Analysis of Audio: The system should take .WAV as input and gather required properties from it.
- Preprocessing: The system must remove all the noise from audio.
- Mood Identification: system must identify the mood of user precisely
- Classification: System must classify the audio at its best attributes.
- **Python:** Install Python on your system, preferably the latest version. Python provides a rich ecosystem of libraries and tools for audio processing, machine learning, and natural language processing, which will be essential for your project.
- **Integrated Development Environment (IDE):** Choose an IDE for Python development that suits your preferences. Popular choices include PyCharm, Visual Studio Code, Jupyter Notebook, or Spyder. These IDEs provide features like code editing, debugging, and project management to streamline your development process.
- **Audio Processing Libraries:** Install the necessary Python libraries for audio processing, such as Librosa, PyDub, or PyAudio. These libraries provide functions and tools for loading, preprocessing, and analyzing audio files.
- **Machine Learning Libraries:** Install machine learning libraries like Scikit-learn, TensorFlow, or Keras. These libraries offer various algorithms and models for training sentiment classifiers.
- **Additional Libraries:** Depending on specific project requirements, we may need additional libraries for tasks such as natural language processing (NLTK or spaCy), data visualization (Matplotlib or Seaborn), or web development (Flask or Django) if you plan to build a user interface for your project.

- **Audio Data:** Collect or obtain a dataset of audio samples labelled with corresponding sentiments or moods. You can either record your own audio samples or search for existing datasets available online. Make sure the audio data is in a compatible format for your chosen libraries.
- **Operating System:** Python is compatible with various operating systems like Windows, macOS, and Linux. Ensure that your chosen operating system supports the required libraries and tools for your project.
- **Hardware Requirements:** Depending on the complexity of your project and the size of your dataset, you may need a computer with sufficient processing power and memory. Additionally, if you plan to capture audio in real-time, you may need a microphone or audio input device connected to your computer.

Non-Functional Requirement:

- **Performance:** The system should be able to process audio inputs and generate sentiment analysis results within an acceptable timeframe, ensuring real-time or near-real-time performance.
- **Accuracy:** The sentiment analysis model should strive to achieve a high level of accuracy in predicting the sentiment or mood of the audio samples. The desired accuracy may vary depending on the project's goals and domain.
- **Scalability:** The system should be designed to handle an increasing number of concurrent users or larger datasets without compromising performance. It should scale effectively with growing demands.
- **Robustness:** The system should handle various scenarios, including different audio qualities, background noise, accents, and variations in speaking style. It should be robust enough to handle edge cases and provide reliable results.
- **Security:** If the project involves handling sensitive or personal audio data, appropriate security measures should be implemented to protect the confidentiality and integrity of the data.
- **Usability:** The system should have a user-friendly interface for capturing audio inputs and displaying sentiment analysis results. It should be intuitive and easy to use for both technical and non-technical users.

- **Maintainability:** The codebase should be well-structured, modular, and documented to facilitate future updates, enhancements, and bug fixes. It should be maintainable by the development team or future developers.
- **Compatibility:** The system should be compatible with different operating systems, audio file formats, and versions of Python and the required libraries. It should be flexible enough to adapt to changes in the software ecosystem.
- **Resource Utilization:** The system should utilize system resources (CPU, memory, etc.) efficiently, avoiding excessive consumption that could impact other processes running on the same machine.
- **Ethical Considerations:** Consider ethical aspects related to the use of audio data, such as obtaining appropriate consent, respecting privacy, and adhering to legal and ethical guidelines for data usage. Performance: The speed of internet connection must be good enough.
- **Availability:** The system should be highly available, ensuring minimal downtime and providing uninterrupted service to users. Consider implementing measures such as redundancy, failover mechanisms, and monitoring to achieve high availability.
- **Reliability:** The system should be reliable, consistently providing accurate sentiment analysis results without errors or failures. It should handle exceptions gracefully and recover from errors effectively.
- **Portability:** The system should be portable across different environments and platforms. It should be designed to run on various operating systems and be easily deployable on different hardware configurations.
- **Interoperability:** The system should have the ability to integrate or interact with other systems or components seamlessly. Consider standards or protocols for interoperability, such as APIs or data exchange formats.
- **Compliance:** If your project involves sensitive data or operates in a regulated industry, ensure compliance with relevant regulations, such as data protection laws (e.g., GDPR) or industry-specific standards (e.g., HIPAA for healthcare data).
- **Documentation:** Provide comprehensive documentation that includes user manuals, technical guides, and code documentation. It should cover system functionality, installation instructions, usage guidelines, and troubleshooting steps.

4.2 System Modules

Modules:

1. **Audio Input Module:** This module is responsible for accepting audio input from the user. It can handle both recorded audio files and live audio streams.
2. **Preprocessing Module:** The preprocessing module processes the audio input to enhance the quality and remove noise or interference. It may involve techniques such as noise reduction, audio normalization, or feature extraction.
3. **Speech-to-Text Conversion Module:** If included, this module converts the audio input into text format using speech recognition algorithms or libraries. It enables the system to perform sentiment analysis on the textual content of the speech.
4. **Sentiment Analysis Module:** This module analyzes the audio or text input to determine the sentiment or mood of the user. It utilizes machine learning or deep learning algorithms trained on sentiment-labeled data to predict the sentiment as positive, negative, or neutral.
5. **Output Generation Module:** The output generation module presents the sentiment prediction to the user. It can provide visual feedback, textual output, or an interactive interface to convey the analyzed sentiment effectively.
6. **Integration and Deployment Module:** This module handles the integration of various components and modules, ensuring they work together seamlessly. It also facilitates the deployment of the system, making it accessible as a web application

4.3 System Modeling & Design

4.3.1 System Architecture :

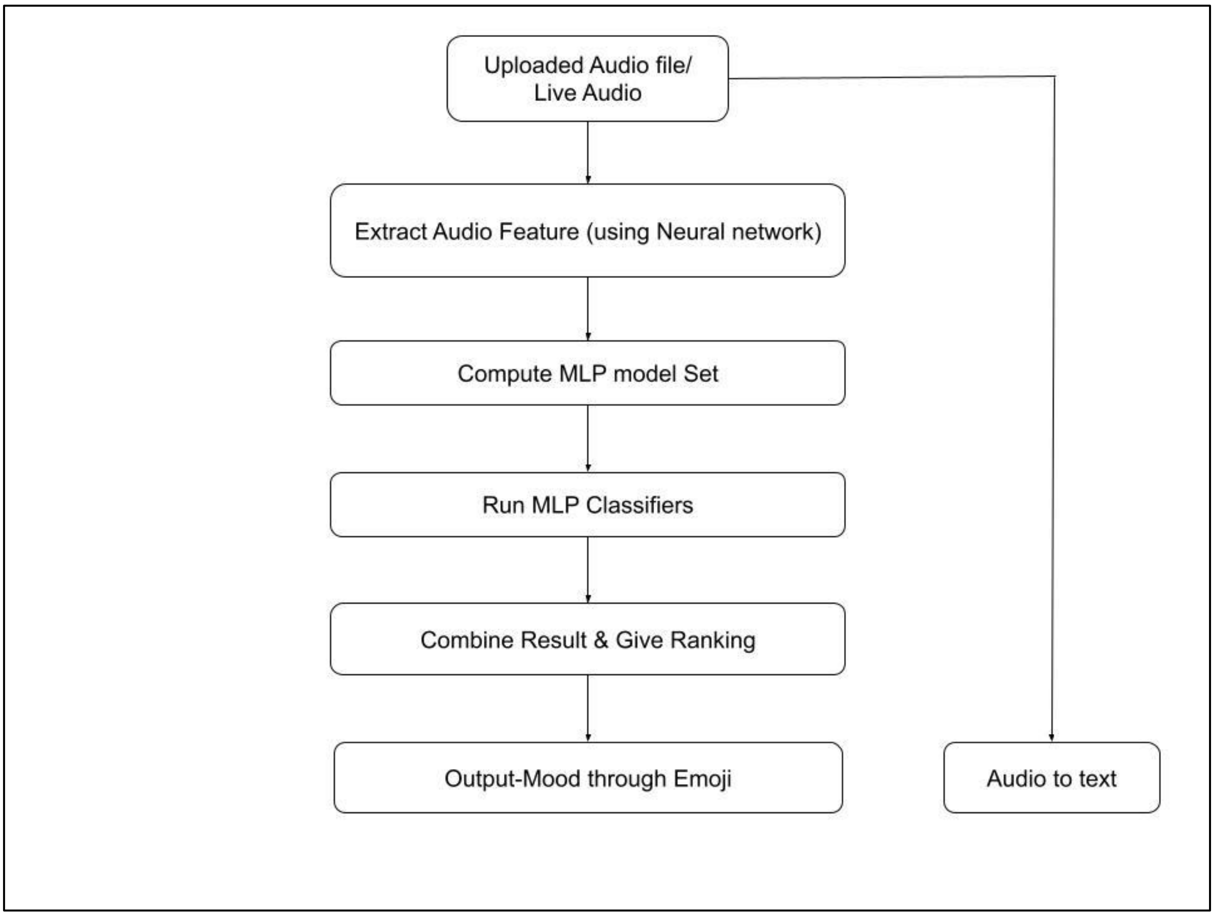

Uploaded Audio file/ Live Audio
Extract Audio Feature (using Neural network)
Compute MLP model Set
Run MLP Classifiers
Combine Result & Give Ranking
Output-Mood through Emoji
Audio to text

Fig. 4.3.1 System Architecture

Description : System architecture shows how project is going to conduct all its processes. First user will upload or record live audio then system will extract features using cnn then mlp classifier will classify and rank moods then output will be mood emoji and text.

4.3.2 DFD Level 0:

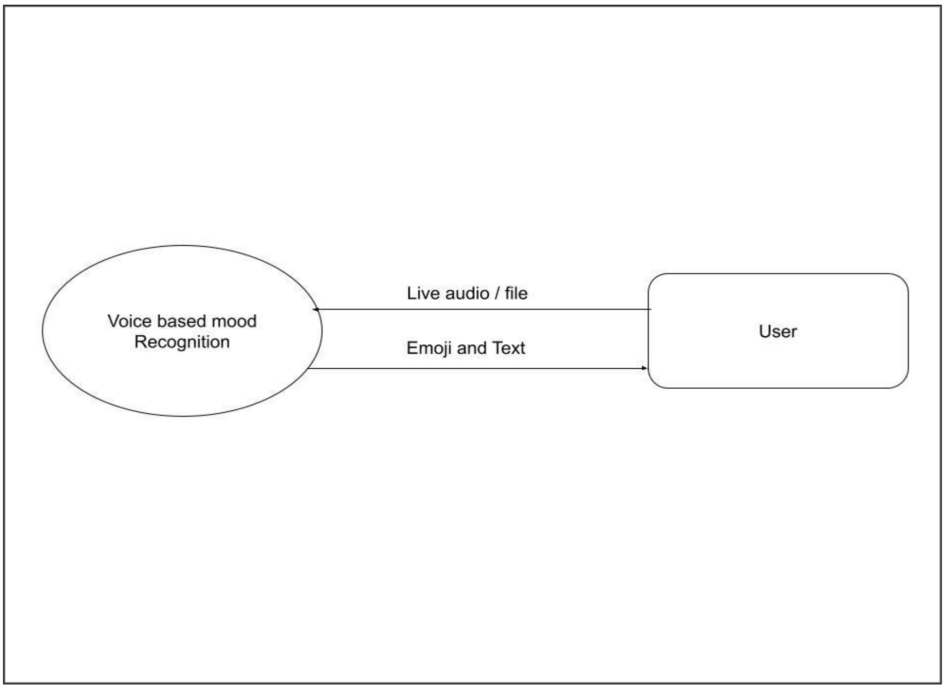


Fig. 4.3.2 Data Flow Diagram Level 0

Description : In DFD level 0 user will give input as live audio or file to voice based mood recognition system and system will provide output as emoji and text.

4.3.3 DFD Level 1:

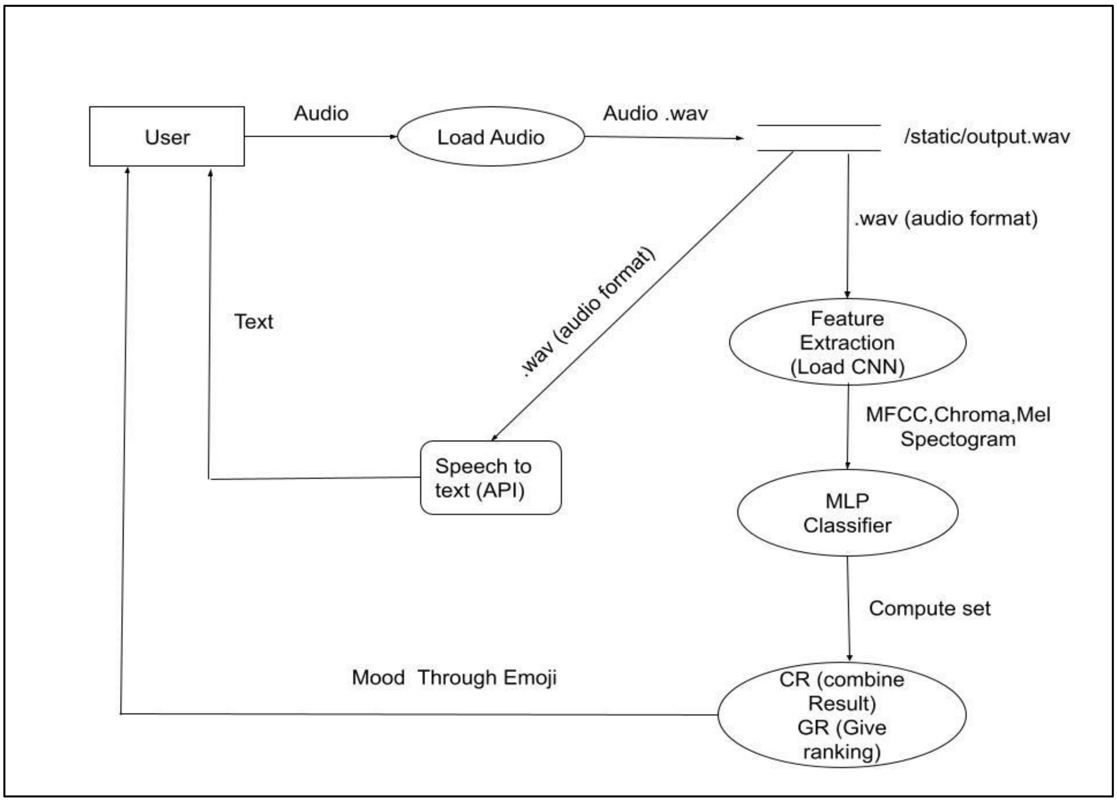


Fig.4.3.3 Data Flow Diagram Level 1

Description: user will give input as audio then it will be loaded in the system by load audio module. Then loaded audio in the WAV format will be stored in temporary storage (/static/output.wav) Then it will be conveyed to speech to text api to convert it to text. After temporary storage audio goes to feature extraction (using cnn) then features lie mfcc chroma mel spectrogram etc to mlp classifier then computed set and ranking will be given and respective mood will be given as output to user.

4.3.4 Use Case Diagram

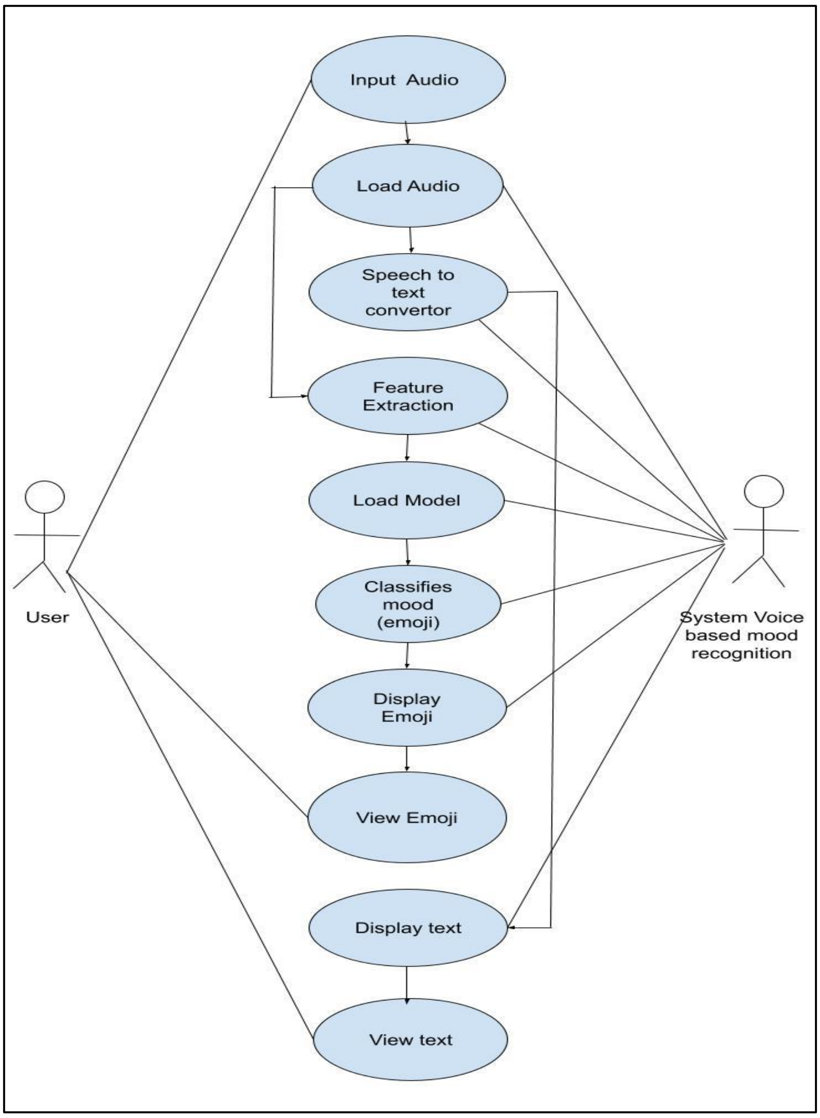


Fig 4.3.4: Use Case Diagram

Description : In this diagram we have shown what users and system responsibilities will be their in system .User will have to give audio as input ten system responsibilities will be speech to text conversion ,load model ,feature extraction, classification and showing output as emoji. Then user will look for a text and an emoji.

4.3.5 Sequence Diagram

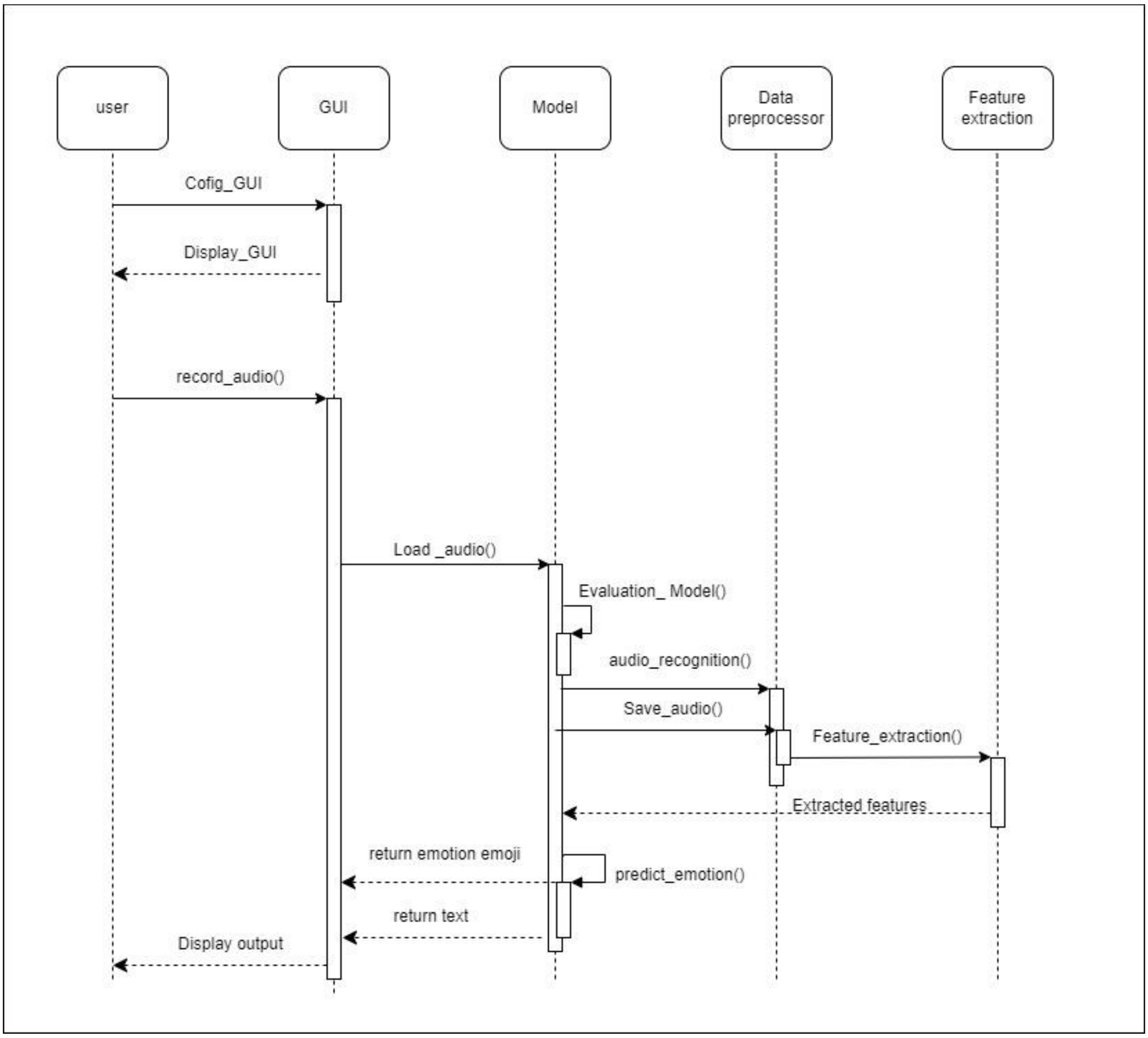


Fig 4.3.5: Sequence Diagram

Description : The objects are user, gui, Model, Data preprocessor, Feature extraction.

The user will first configure gui that's is the first page user will visit is homepage of our system. Then on that user will upload audio .then it will goes to load model and then data preprocessing will happen ten from that audio features will be extracted then extracted features will e loaded into model then emoji will be returned to user on homepage in output window.

4.4 Algorithm Design

4.4.1 CNN

Convolutional Neural Network (CNN)

A CNN is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data it has powerful image processing unit.

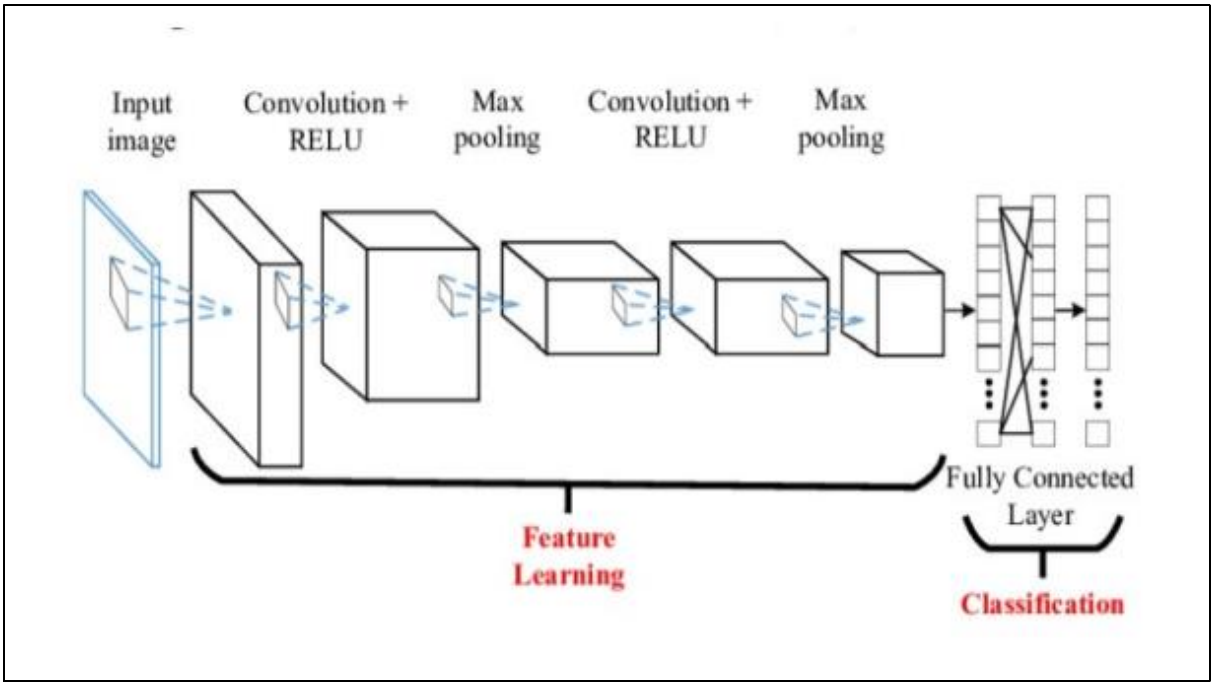


Fig. 4.4.1. CNN Model

Convolution layer -The purpose of this layer is to receive a feature map. Usually, we have tendency to begin with low range of filters for low-level feature detection. The deeper we enter the CNN, the additional filters we use to find high-level options. Feature detection is relies on 'scanning' the input with the filter of a given size and applying matrix computations so as to derive a feature map.

ReLu layer - Rectifier Unit, the foremost commonly deployed activation function for the outputs of the CNN neurons.

Max pooling layer -The goal of this layer is to provide spatial variance, that simply means that the system are going to be capable of recognizing an object even once its looks varies in a way. Pooling layer can perform a down sampling operation on the spatial dimensions (width, height), leading to output like [16x16x12] for pooling_size=(2, 2).

Dropout layer - A dropout layer randomly sets input parts to zero with a given chance.

Cross channel normalization layer - This layer performs a channel-wise local response normalization. It always follows the ReLU activation layer. This layer replaces every component with a normalized value it obtains using the elements from a explicit range of neighboring channels (elements in the normalization window)

Fully Connected layer - In a fully connected layer, we have tendency to flatten the output of the last convolution layer and connect each node of the present layer with the opposite nodes of consequent layer. Neurons in a fully connected layer have full connections to all or any activations within the previous layer, as seen in regular Neural Networks and add a similar way

Classification layer -The classification layer produces the expected class. Softmax layer - A softmax layer to predict the output

4.3.2 MLP Classifier

MLP Classifier stands for Multi-layer Perceptron classifier which in the name itself connects to a Neural Network. Unlike other classification algorithms such as Support Vectors or Naive Bayes Classifier, MLPClassifier relies on an underlying Neural Network to perform the task of classification.

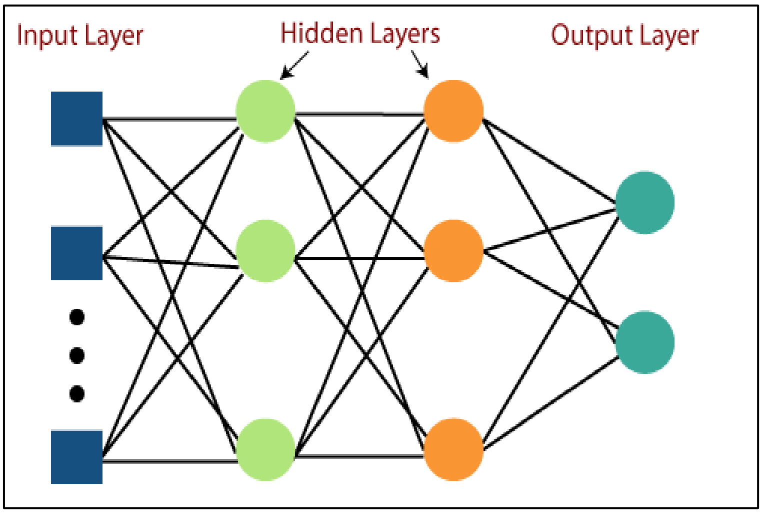


Fig.4.3.2 MLP Classifier

Steps Involved in MLP :

1. **Importing the Dataset:** In this step, you import the dataset that contains the data you will use to train and test your MLP classifier. You can use the pandas library to read the dataset from a file, such as a CSV file. The code snippet data = pd.read_csv("Final_Train_Dataset.csv") imports the dataset into a pandas DataFrame.

   Eg-

   ```
   import pandas as pd
   data = pd.read_csv("audio_data.csv")
   ```

2. **Cleaning the Data:** Depending on the quality and structure of your dataset, you may need to perform data cleaning operations. This can include handling missing values, removing duplicates, encoding categorical variables, or any other necessary data preprocessing steps.

   Sample Code-

   ```
   data = data.dropna()
   ```

3. **Feature Scaling:** If your dataset contains numerical features with different scales, it is often beneficial to apply feature scaling to bring all features to a similar range. Common scaling techniques include standardization (subtracting the mean and dividing by the standard deviation) or normalization (scaling values between 0 and 1).

   Sample-

   ```
   from sklearn.preprocessing import StandardScaler
   scaler = StandardScaler()
   data[['feature1', 'feature2']] = scaler.fit_transform(data[['feature1', 'feature2']])
   ```

4. **Creating Training and Validation Sets:** Splitting your dataset into training and validation sets is crucial for evaluating the performance of your MLP classifier. The code snippet train_test_split from scikit-learn is used to randomly divide the dataset into training and validation subsets. This allows you to train the model on a portion of the data and evaluate its performance on unseen data.

Sample-

```
from sklearn.model_selection import train_test_split
X = data[['feature1', 'feature2']]
y = data['sentiment']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42)
```

5. **Measuring the Accuracy:** In this step, you define a function accuracy to measure the accuracy of your predictions. The function takes a confusion matrix as input and calculates the accuracy as the ratio of the sum of correctly predicted samples to the total number of samples.

   Sample-

   ```
   from sklearn.metrics import accuracy_score
   def accuracy(confusion_matrix):
   diagonal_sum = confusion_matrix.trace()
   sum_of_all_elements = confusion_matrix.sum()
   return diagonal_sum / sum_of_all_elements
   ```

6. **Building the MLP Classifier:** Here, you import the MLPClassifier from scikit-learn, which is a flexible implementation of the MLP neural network architecture. You initialize the classifier with specific parameters, including the hidden_layer_sizes (the number of nodes in each hidden layer), max_iter (the maximum number of iterations/epochs for training), activation function (e.g., relu, sigmoid), solver (the algorithm for weight optimization), and random_state (for reproducibility).

   Sample-

   ```
   from sklearn.neural_network import MLPClassifier
   classifier = MLPClassifier(hidden_layer_sizes=(150, 100, 50), max_iter=300,
   activation='relu', solver='adam', random_state=1)
   ```

7. **Fitting the Training Data:** Using the fit method of the MLPClassifier, you train the neural network model on the training data (X_train and y_train). This step involves adjusting the weights and biases of the neural network to minimize the error between the predicted and actual outputs.

   Sample-

   ```
   classifier.fit(X_train, y_train)
   ```

8. **Predicting with the Trained Model:** After training the MLP classifier, you can use it to make predictions on unseen data. The predict method is used to predict the target variable (y_pred) for the validation set (X_val).

   Sample-

   ```
   y_pred = classifier.predict(X_val)
   ```

9. **Calculating the Accuracy:** With the predicted labels and the true labels of the validation set, you can calculate the accuracy of your MLP classifier. The confusion matrix is computed using the confusion_matrix function from scikit-learn. Then, the accuracy function you defined earlier is used to calculate the accuracy of the predictions based on the confusion matrix.

   Sample-

   ```
   from sklearn.metrics import confusion_matrix
   cm = confusion_matrix(y_pred, y_val)
   accuracy_val = accuracy(cm)
   print("Accuracy of MLPClassifier:", accuracy_val)
   ```

4.5 System Architecture

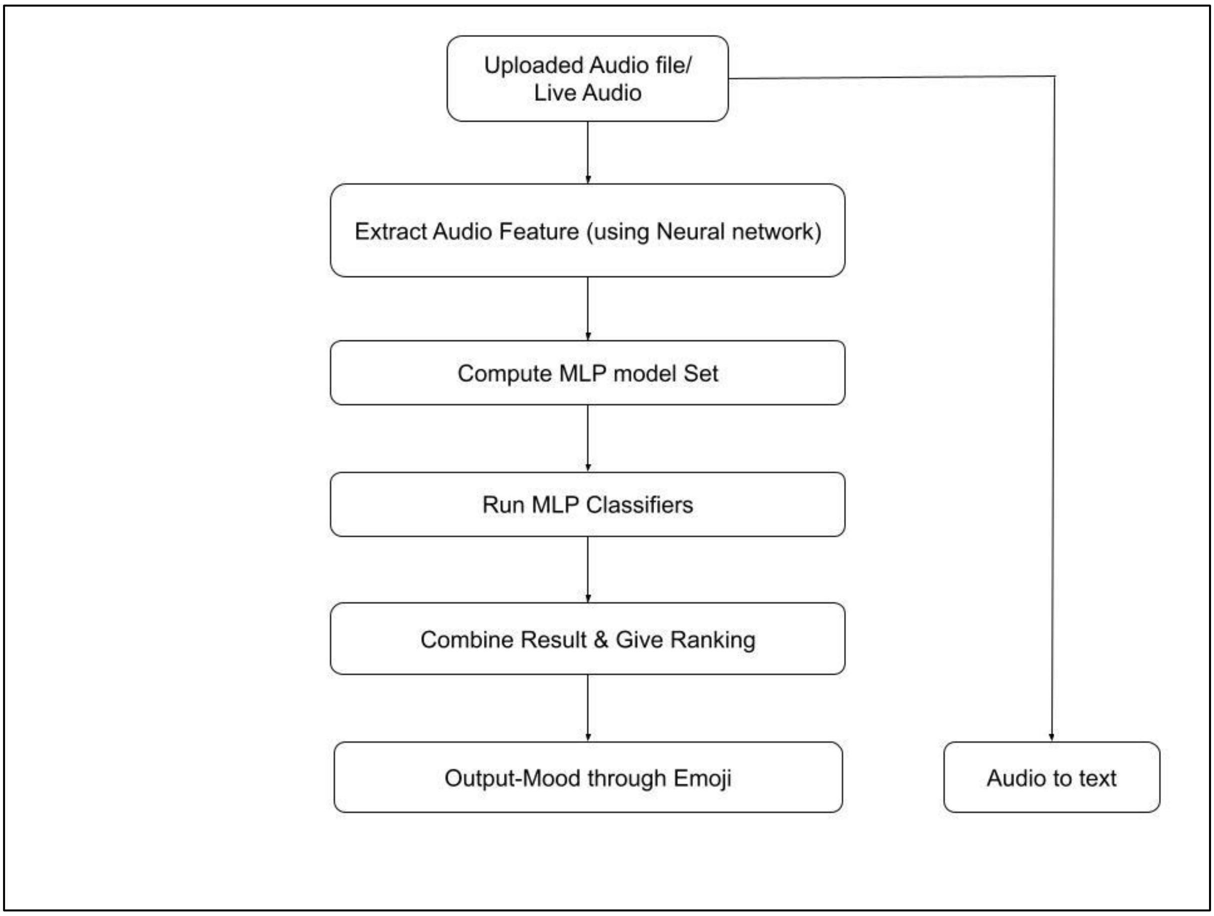

Uploaded Audio file/ Live Audio
Extract Audio Feature (using Neural network)
Compute MLP model Set
Run MLP Classifiers
Combine Result & Give Ranking
Output-Mood through Emoji
Audio to text

Fig 4.5 System Architecture

Description : System architecture shows how project is going to conduct all its processes. First user will upload or record live audio then system will extract features using cnn then mlp classifier will classify and rank moods then output will be mood emoji and text.

The overall system architecture ensures the seamless integration of the various modules to provide an accurate and efficient sentiment analysis of audio data. It enables users to input audio, process it through feature extraction and sentiment analysis models, and present the results with additional features like speech-to-text conversion and language translation.

- **User Interface (UI):**

  The UI component provides an interface for users to interact with the system. It allows users to input audio files or record live audio for sentiment analysis. The UI also provides options for language selection, speech-to-text conversion, and viewing the sentiment analysis results.

- **Audio Input Module:**

  The audio input module handles the retrieval of audio data from the user. It can process both recorded audio files and live audio streams. The module captures the audio input and passes it to the subsequent processing stages.

- **Preprocessing Module:**

  The preprocessing module performs necessary preprocessing steps on the audio data to enhance its quality and prepare it for analysis. This may include techniques such as noise reduction, resampling, and normalization.

- **Feature Extraction Module:**

  The feature extraction module extracts relevant features from the preprocessed audio data. It analyzes the audio signals to capture characteristics that are indicative of different sentiments. Common features used in sentiment analysis include spectral features, mel-frequency cepstral coefficients (MFCCs), and pitch.

- **Sentiment Analysis Models:**

  The sentiment analysis models are the core components of the system. The project utilizes two types of models: Convolutional Neural Networks (CNN) and Multi-Layer Perceptron (MLP). These models are trained on labeled audio data to learn patterns and relationships between audio features and sentiments. The CNN model excels at capturing spatial information from audio spectrograms, while the MLP model can handle sequential and temporal data.

- **Prediction Module:**

  The prediction module takes the extracted features and feeds them into the trained sentiment analysis models. It applies the models to predict the sentiment or emotion associated with the input audio. The module produces sentiment labels or scores corresponding to different sentiments such as positive, negative, neutral, etc.

- **Speech-to-Text Conversion Module:**

  The speech-to-text conversion module converts the audio input into textual representations. It utilizes speech recognition techniques and algorithms to transcribe spoken words into text. This module enhances the system's usability by allowing users to view the transcribed text alongside the sentiment analysis results.

- **Language Translation Module:**

  The language translation module facilitates language translation for international users. It takes the transcribed text or sentiment labels and translates them from the detected language to English. This feature enables users to understand sentiments expressed in different languages.

- **Result Presentation:**

  The result presentation component displays the sentiment analysis results to the user. It presents the sentiment labels, scores, and any translated text in a user-friendly format. Visualizations, such as bar charts or sentiment word clouds, can be incorporated to enhance the understanding and interpretation of the results.

Chapter 5
Implementation and Coding

5.1 Algorithms

1.Introduction:

- Overview of audio sentiment analysis and its importance in understanding user emotions.
- Explanation of the significance of algorithm selection for accurate sentiment analysis.
- Research objectives and the rationale behind comparing CNN and MLP algorithms.

2. Convolutional Neural Networks (CNN):

- Detailed explanation of the CNN architecture and its application in audio sentiment analysis.
- Description of convolutional layers, pooling layers, and fully connected layers in CNN.
- Discussion on the advantages of CNN for audio feature extraction and sentiment classification.
- Review of relevant research papers that have utilized CNN for audio sentiment analysis.

3. Multi-Layer Perceptron (MLP):

- Introduction to the MLP architecture and its suitability for audio sentiment analysis.
- Explanation of the layers in MLP, including input layer, hidden layers, and output layer.
- Discussion on the capabilities and limitations of MLP in sentiment analysis tasks.
- Overview of existing studies that have employed MLP for audio sentiment analysis.

4. Training and Evaluation:

- Description of the dataset used for training and evaluation of CNN and MLP models.
- Discussion on the preprocessing steps applied to the audio data.
- Explanation of the training process, including loss functions, optimization algorithms, and hyperparameter tuning.
- Presentation of the evaluation metrics used to assess the performance of CNN and MLP models.

5. Performance Analysis:

- Comparative analysis of the performance of CNN and MLP models in audio sentiment analysis.
- Evaluation of key performance metrics, such as accuracy, precision, recall, and F1-score.
- Discussion on the strengths and weaknesses of CNN and MLP based on their performance results.
- Analysis of computational requirements and efficiency considerations for both algorithms.

5.1.1 Convolutional Neural Network :

A **Convolutional Neural Network (CNN)** is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

When it comes to Machine Learning, AI perform really well. Neural Networks are used in various datasets like images, audio, and text. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Netwok more precisely an LSTM, similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

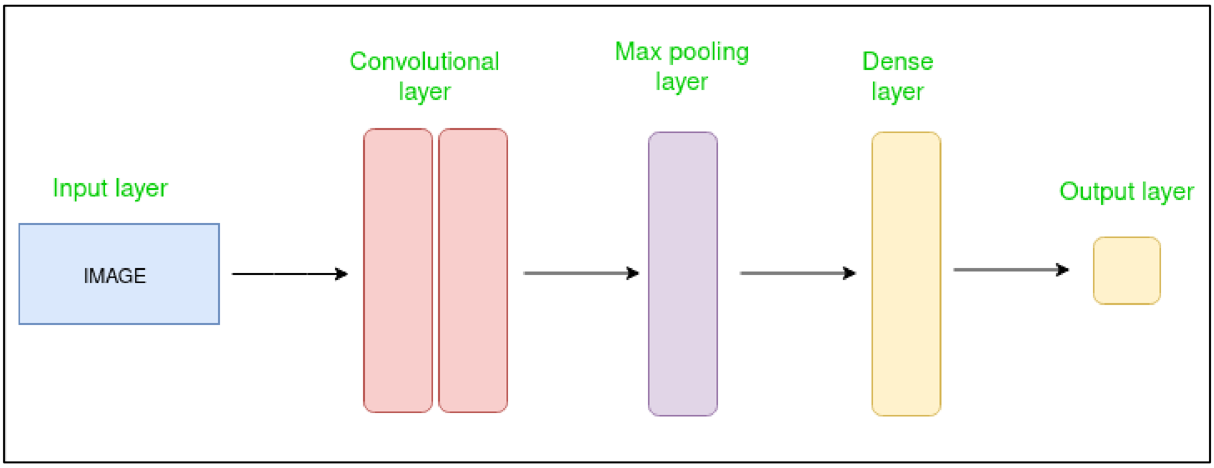
In a regular Neural Network there are three types of layers:

1. **Input Layers:** It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).
2. **Hidden Layer:** The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.
3. **Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

The data is fed into the model and output from each layer is obtained from the above step is called feedforward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called backpropogation which basically is used to minimize the loss.

CNN architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.



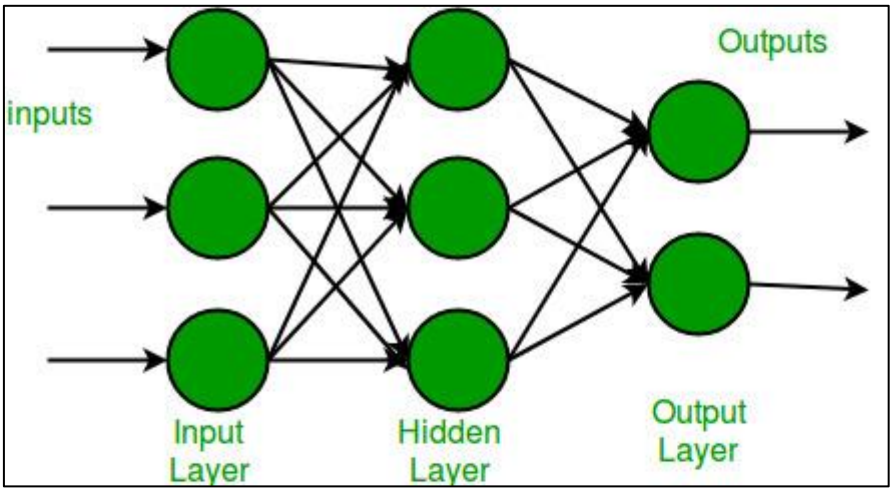
5.1.1 Simple CNN Architecture

The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

5.1.2 MLP:

Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons.

A multi-layer perceptron has one input layer and for each input, there is one neuron(or node), it has one output layer with a single node for each output and it can have any number of hidden layers and each hidden layer can have any number of nodes. A schematic diagram of a Multi-Layer Perceptron (MLP) is depicted below.



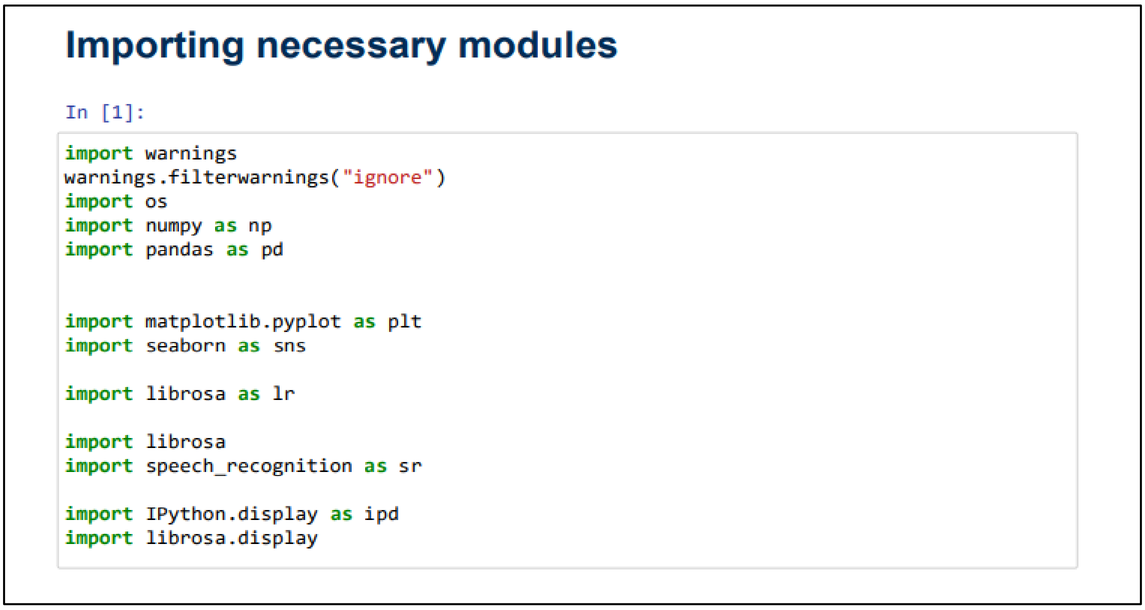
5.1.2 MLP architecture

In the multi-layer perceptron diagram above, we can see that there are three inputs and thus three input nodes and the hidden layer has three nodes. The output layer gives two outputs, therefore there are two output nodes. The nodes in the input layer take input and forward it for further process, in the diagram above the nodes in the input layer forwards their output to each of the three nodes in the hidden layer, and in the same way, the hidden layer processes the information and passes it to the output layer.

Every node in the multi-layer perception uses a sigmoid activation function. The sigmoid activation function takes real values as input and converts them to numbers between 0 and 1 using the sigmoid formula.

5.1.3 Implementation :

1. Importing Necessary Modules :



Importing necessary modules

In [1]:

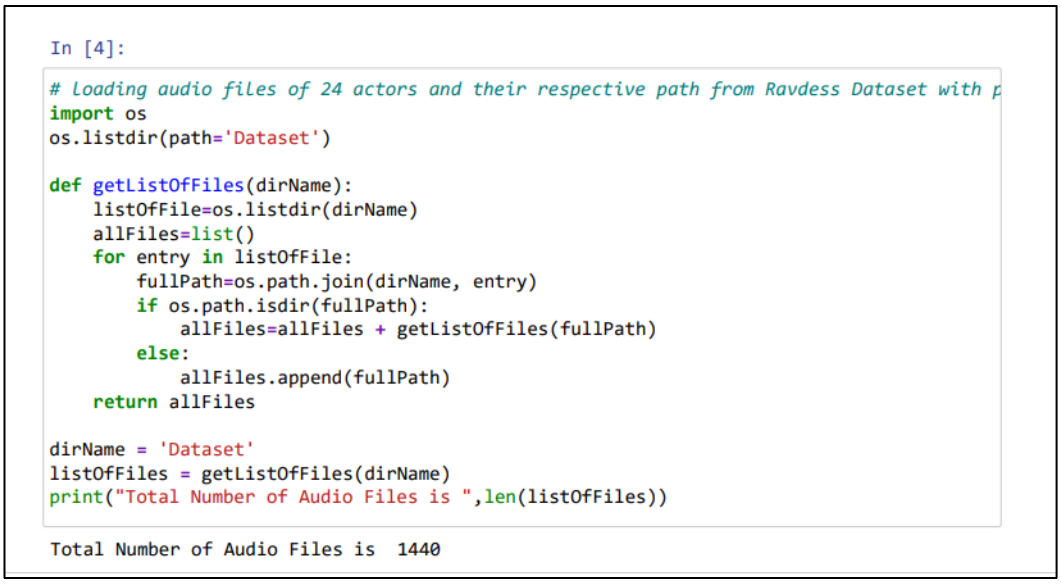
```
import warnings
warnings.filterwarnings("ignore")
import os
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
import seaborn as sns

import librosa as lr

import librosa
import speech_recognition as sr

import IPython.display as ipd
import librosa.display
```

Description :

- **warnings:** The warnings module is used to filter and suppress warning messages that may be displayed during the execution of your code.
- **os:** The os module provides functions for interacting with the operating system. It is commonly used for file and directory operations.
- **numpy (imported as np):** The numpy library is used for mathematical operations and array manipulation. It provides efficient data structures and functions for working with multi-dimensional arrays.
- **pandas (imported as pd):** The pandas library is used for data manipulation and analysis. It provides data structures like DataFrames that allow you to work with tabular data efficiently.
- **matplotlib.pyplot (imported as plt):** The pyplot submodule of matplotlib is used for creating visualizations, such as plots and charts, to analyze and display data.
- **seaborn (optional):** The seaborn library is a data visualization library built on top of matplotlib. It provides additional functionality and aesthetically pleasing visualizations.
- **librosa:** The librosa library is specifically designed for audio analysis. It provides functions and tools for various audio processing tasks, such as loading audio files, extracting features, and visualizing audio data.

- **speech_recognition (imported as sr):** The speech_recognition library allows you to perform speech recognition tasks, such as converting spoken language into text. It provides an interface to various speech recognition engines.
- **IPython.display (imported as ipd):** The IPython.display module provides functions for displaying audio, images, and other media in Jupyter notebooks or the IPython interactive environment.
- **librosa.display:** The display module in librosa provides functions for visualizing audio data, such as waveforms and spectrograms.

2. Loading Dataset :



```
In [4]:
# Loading audio files of 24 actors and their respective path from Ravdess Dataset with p
import os
os.listdir(path='Dataset')

def getListOfFiles(dirName):
    listOfFile=os.listdir(dirName)
    allFiles=list()
    for entry in listOfFile:
        fullPath=os.path.join(dirName, entry)
        if os.path.isdir(fullPath):
            allFiles=allFiles + getListOfFiles(fullPath)
        else:
            allFiles.append(fullPath)
    return allFiles

dirName = 'Dataset'
listOfFiles = getListOfFiles(dirName)
print("Total Number of Audio Files is ",len(listOfFiles))

Total Number of Audio Files is  1440
```

Description :

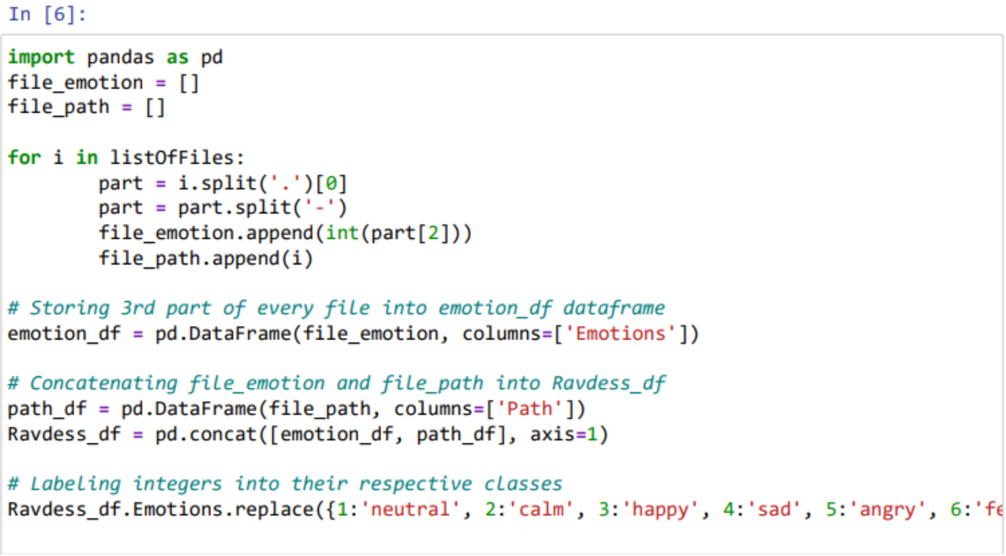
function called getListOfFiles() that recursively traverses a directory and its subdirectories to retrieve a list of all files within them. The function takes a directory path as input and returns a list of file paths.

The code uses the os module to interact with the operating system and perform file and directory operations. It utilizes the os.listdir() function to retrieve a list of entries (files and directories) within the specified directory.

The function loops through each entry and checks if it is a directory using the os.path.isdir() function. If it is a directory, the function recursively calls itself with the subdirectory path and appends the returned list of files to the main list. If it is a file, the full path is appended to the list.

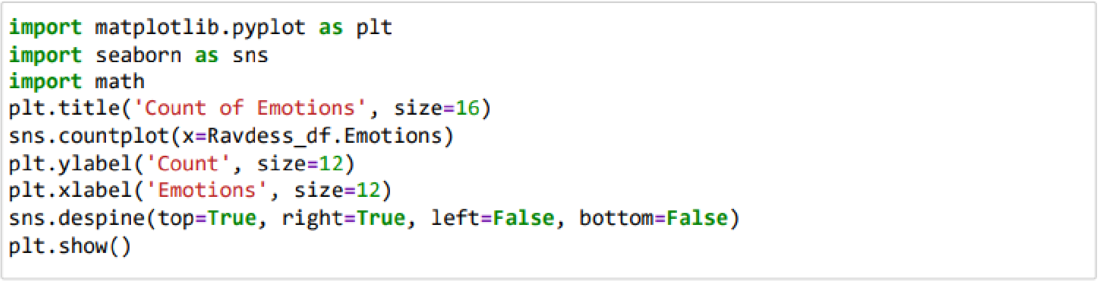
The code allows you to specify the directory path by assigning it to the dirName variable. It then calls the getListOfFiles() function with the directory path and assigns the returned list of files to the listOfFiles variable.

3.Seperating and labelling Emotions from Dataset:

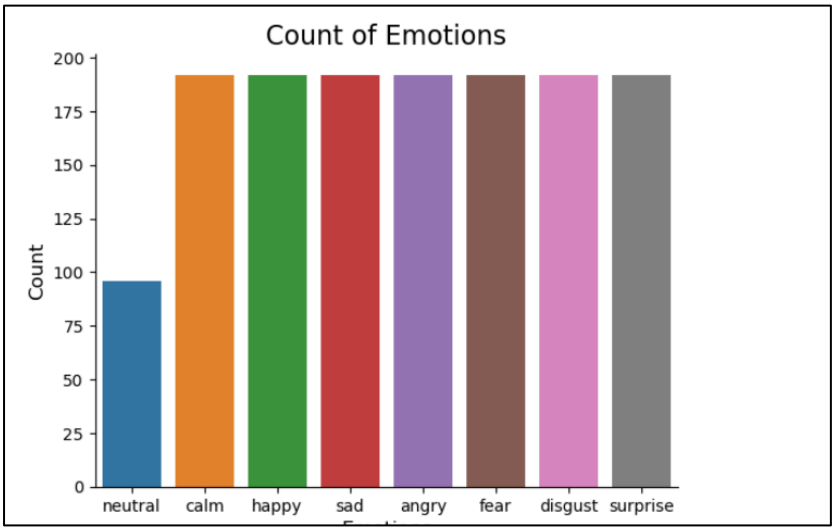


```
In [6]:
import pandas as pd
file_emotion = []
file_path = []

for i in listOfFiles:
        part = i.split('.')[0]
        part = part.split('-')
        file_emotion.append(int(part[2]))
        file_path.append(i)

# Storing 3rd part of every file into emotion_df dataframe
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])

# Concatenating file_emotion and file_path into Ravdess_df
path_df = pd.DataFrame(file_path, columns=['Path'])
Ravdess_df = pd.concat([emotion_df, path_df], axis=1)

# Labeling integers into their respective classes
Ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:'angry', 6:'fe
```

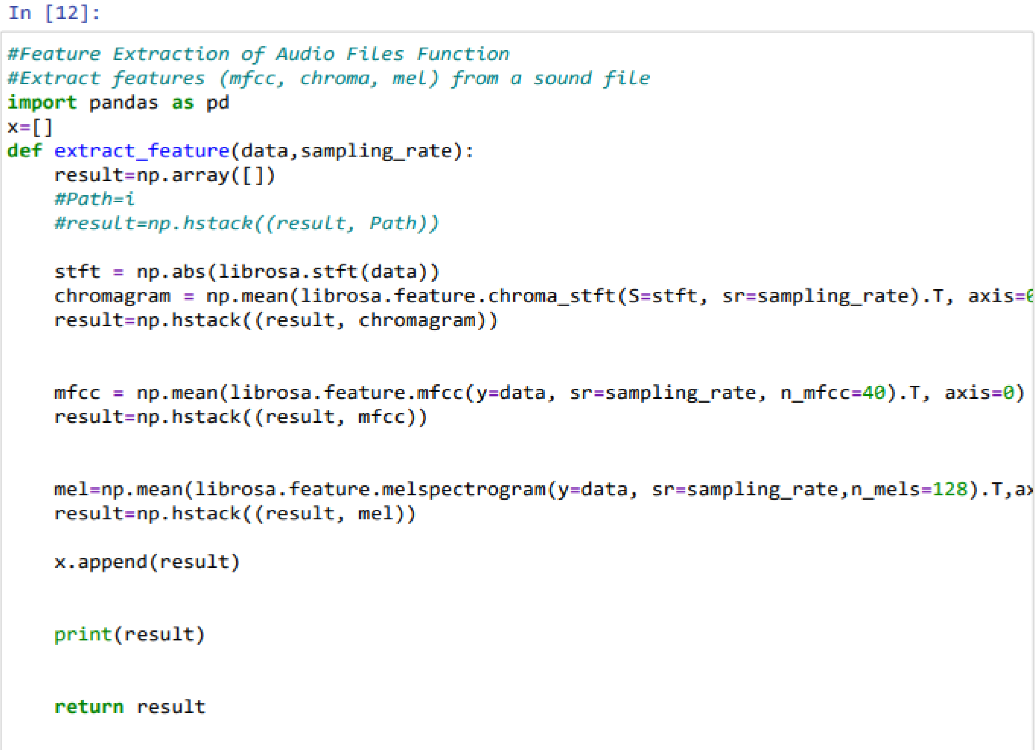
4.Basic Visualization:



```
import matplotlib.pyplot as plt
import seaborn as sns
import math
plt.title('Count of Emotions', size=16)
sns.countplot(x=Ravdess_df.Emotions)
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
sns.despine(top=True, right=True, left=False, bottom=False)
plt.show()
```



5.Feature Extraction:

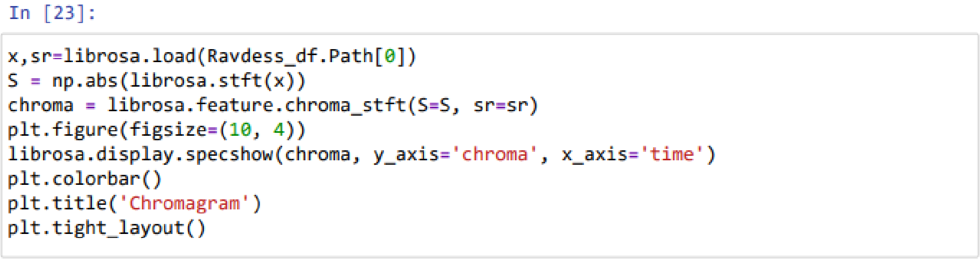


In [12]:

```
#Feature Extraction of Audio Files Function
#Extract features (mfcc, chroma, mel) from a sound file
import pandas as pd
x=[]
def extract_feature(data,sampling_rate):
    result=np.array([])
    #Path=i
    #result=np.hstack((result, Path))

    stft = np.abs(librosa.stft(data))
    chromagram = np.mean(librosa.feature.chroma_stft(S=stft, sr=sampling_rate).T, axis=0
    result=np.hstack((result, chromagram))


    mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sampling_rate, n_mfcc=40).T, axis=0)
    result=np.hstack((result, mfcc))


    mel=np.mean(librosa.feature.melspectrogram(y=data, sr=sampling_rate,n_mels=128).T,ax
    result=np.hstack((result, mel))

    x.append(result)


    print(result)


    return result
```

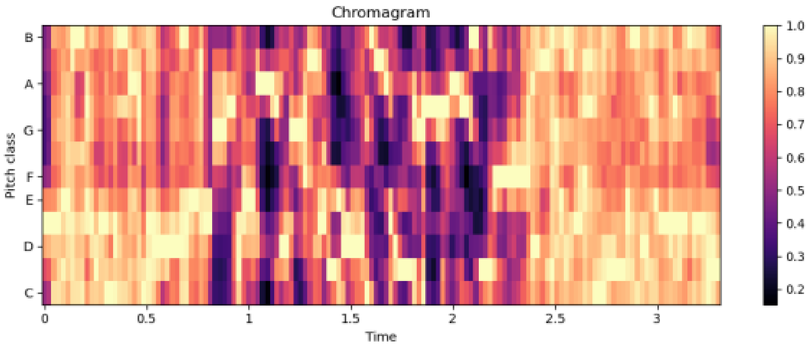
6.Graphical Representation of Features:

1.Chroma:

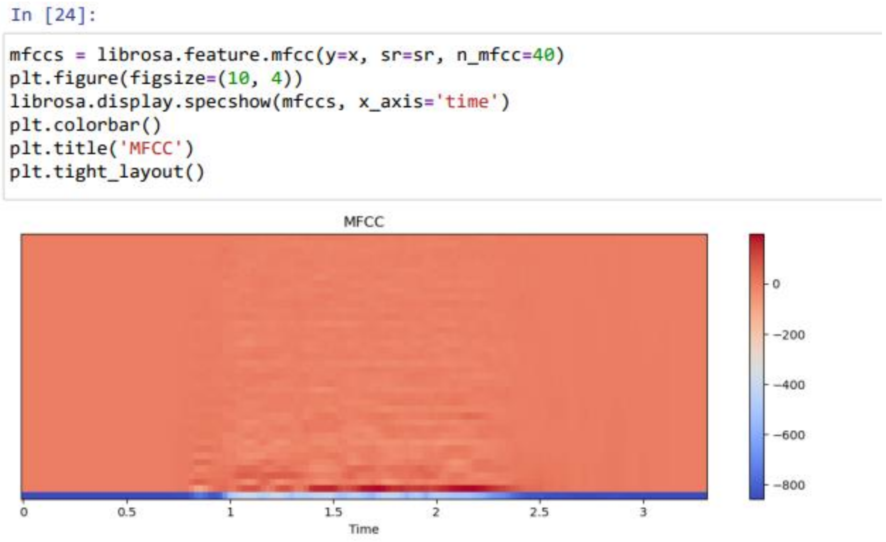


In [23]:

```
x,sr=librosa.load(Ravdess_df.Path[0])
S = np.abs(librosa.stft(x))
chroma = librosa.feature.chroma_stft(S=S, sr=sr)
plt.figure(figsize=(10, 4))
librosa.display.specshow(chroma, y_axis='chroma', x_axis='time')
plt.colorbar()
plt.title('Chromagram')
plt.tight_layout()
```



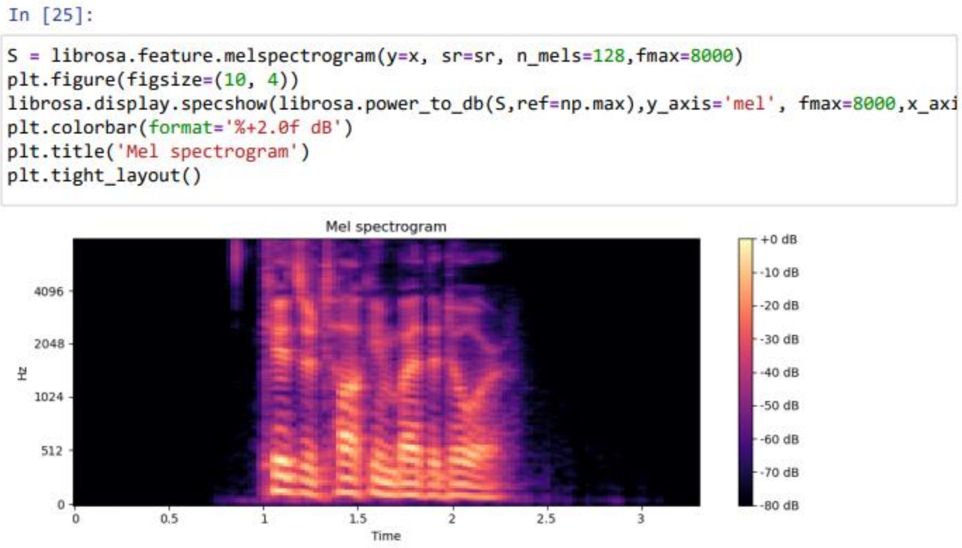
2.MFCC:



In [24]:

```
mfccs = librosa.feature.mfcc(y=x, sr=sr, n_mfcc=40)
plt.figure(figsize=(10, 4))
librosa.display.specshow(mfccs, x_axis='time')
plt.colorbar()
plt.title('MFCC')
plt.tight_layout()
```

3.Mel Spectrogram:



In [25]:

```
S = librosa.feature.melspectrogram(y=x, sr=sr, n_mels=128,fmax=8000)
plt.figure(figsize=(10, 4))
librosa.display.specshow(librosa.power_to_db(S,ref=np.max),y_axis='mel', fmax=8000,x_axi
plt.colorbar(format='%+2.0f dB')
plt.title('Mel spectrogram')
plt.tight_layout()
```

5.2 Flow Chart

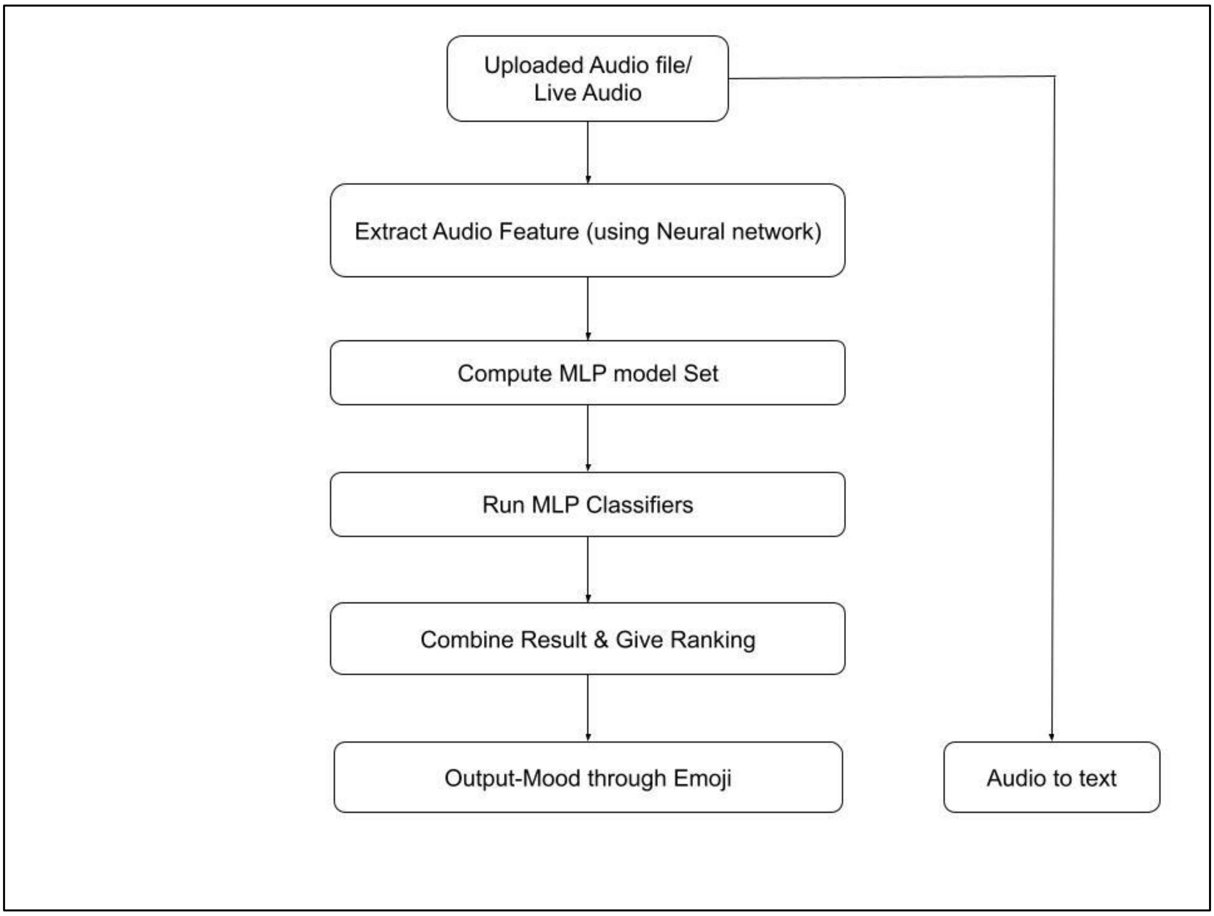

Uploaded Audio file/ Live Audio
Extract Audio Feature (using Neural network)
Compute MLP model Set
Run MLP Classifiers
Combine Result & Give Ranking
Output-Mood through Emoji
Audio to text

Fig 5.2 Flow Chart

Description : System architecture shows how project is going to conduct all its processes. First user will upload or record live audio then system will extract features using cnn then mlp classifier will classify and rank moods then output will be mood emoji and text.

5.3 Software Requirements with relevant justification

- **Python:** Python is a versatile programming language widely used in machine learning and data analysis. It provides extensive libraries and frameworks such as NumPy, Pandas, and Scikit-learn, which are essential for audio processing, feature extraction, and sentiment analysis.
- **Librosa:** Librosa is a Python library specifically designed for audio analysis and processing. It provides functions for loading audio files, extracting audio features, and visualizing audio data. Librosa simplifies the audio-related tasks required in the project.
- **SpeechRecognition:** The SpeechRecognition library enables speech-to-text conversion by integrating with various speech recognition engines. It supports multiple platforms and provides an easy-to-use interface for transcribing spoken words into textual representations.
- **Natural Language Toolkit (NLTK):** NLTK is a powerful library for natural language processing tasks. It offers a wide range of functionalities for text processing, tokenization, stemming, and sentiment analysis. NLTK can be utilized for sentiment analysis on the transcribed text data.
- **Scikit-learn:** Scikit-learn is a popular machine learning library in Python. It provides efficient implementations of various classification algorithms, including Convolutional Neural Networks (CNN) and Multi-Layer Perceptron (MLP). Scikit-learn facilitates the training and evaluation of sentiment analysis models.
- **TensorFlow or Keras:** TensorFlow and Keras are deep learning frameworks that offer high-level APIs for building neural networks. They provide tools and utilities for creating, training, and deploying CNN and MLP models for sentiment analysis. These frameworks enable efficient computation on GPUs, enhancing the model training performance.
- **Jupyter Notebook or Integrated Development Environment (IDE):** Jupyter Notebook is an interactive computing environment that allows code execution, data visualization, and documentation integration. It is well-suited for iterative development and experimentation. Alternatively, an IDE like PyCharm or Anaconda can be used for development, providing a comprehensive development environment.
- **Matplotlib and Seaborn:** Matplotlib and Seaborn are data visualization libraries that facilitate the creation of visual representations of audio data, sentiment analysis results, and other relevant information. These libraries enable the generation of informative plots, graphs, and charts for better understanding and interpretation of the data.

- **Flask or Django:** Flask and Django are web development frameworks in Python. They provide tools and utilities for developing web applications. These frameworks can be utilized to create a user-friendly web interface for the project, enabling users to interact with the system and view sentiment analysis results.

The software requirements listed above justify the selection of essential tools, libraries, and frameworks necessary for the development and implementation of the Audio Sentiment Analysis project. They provide the capabilities required for audio processing, feature extraction, sentiment analysis, visualization, and web development, ensuring the successful execution of the project's objectives.

5.4 Hardware Requirements with relevant justification

- **Computer System:** A computer system is required to run the project. It should meet the following specifications:
- **Processor:** A multi-core processor (e.g., Intel Core i5 or higher) to handle the computational requirements of audio processing and machine learning algorithms.
- **RAM:** At least 8 GB of RAM to ensure smooth execution of resource-intensive tasks and to accommodate large audio datasets.
- **Storage:** Sufficient storage space to store the project code, audio datasets, and any additional files or libraries required for the project.
- **Operating System:** The project can be developed and executed on various operating systems such as Windows, macOS, or Linux.
- **Microphone or Audio Input Device:**To capture live audio for sentiment analysis, a microphone or audio input device is necessary. It should be compatible with the computer system and provide clear and reliable audio input.
- **Graphics Processing Unit (GPU) (optional):**For faster training and inference of deep learning models, a dedicated GPU can be beneficial. GPUs with CUDA support, such as NVIDIA GeForce or NVIDIA Quadro series, can significantly accelerate the execution of machine learning algorithms.
- **Internet Connection:**An internet connection is required to download necessary libraries, datasets, and any additional resources. It is also necessary for language translation services, if incorporated into the project.

Justification:

The hardware requirements listed above are justified by the computational needs and data processing demands of the project. The computer system should have sufficient processing power and memory to handle the audio processing tasks, feature extraction, training of machine learning models, and real-time sentiment analysis. A capable processor and an adequate amount of RAM ensure efficient execution of the project without performance bottlenecks.

The microphone or audio input device is necessary for capturing live audio input for sentiment analysis. It ensures that the system can process audio recordings or real-time audio streams.

While not mandatory, a dedicated GPU can significantly speed up the training and inference processes for deep learning models, reducing the time required for model development and evaluation.

Finally, an internet connection is necessary to access online resources, download libraries, and utilize language translation services, if implemented. It enables the system to stay up to date with the latest tools and technologies and provides access to additional functionalities and services that enhance the project's capabilities.

Chapter 6
Testing

6.1 Fundamentals Of Testing:

- Audio sentiment analysis project involve ensuring the accuracy, reliability, and performance of your system. Here are some key aspects to consider:
- **Functional Testing:** This type of testing focuses on validating that the system functions as intended. It involves testing individual components, modules, and features to ensure they work correctly. For your project, functional testing may involve verifying that the audio sentiment analysis algorithm produces accurate mood predictions and that the speech-to-text conversion and language translation functionalities work properly.
- **Integration Testing:** Integration testing evaluates the interaction between different components or modules of your system. It ensures that the various parts of your project, such as the audio processing, sentiment analysis, and user interface, seamlessly work together. Integration testing helps identify and resolve any issues related to data flow, compatibility, or communication between system components.
- **Performance Testing:** This type of testing assesses the performance of your system under different load conditions. It helps determine how well your application handles a large number of users or a high volume of audio data. Performance testing can include measuring response times, resource utilization, and scalability to ensure your system meets performance requirements.
- **Usability Testing:** Usability testing focuses on evaluating the user-friendliness and ease of use of your system. It involves testing the user interface, user interactions, and overall user experience. For your project, usability testing can involve gathering feedback from users to assess the effectiveness of features like voice input, output display, and user feedback mechanisms.
- **Error Handling and Exception Testing:** This type of testing examines how well your system handles errors, exceptions, and unexpected situations. It involves intentionally triggering error conditions and verifying that your system responds appropriately, providing informative error messages, and maintaining system stability and data integrity.

- **Security Testing:** Security testing ensures that your system is protected against potential vulnerabilities and unauthorized access. It involves identifying and addressing security risks, testing authentication and authorization mechanisms, and safeguarding user data. In your project, security testing can include verifying the security of user voice inputs, protecting sensitive user information, and preventing potential security breaches.
- **Regression Testing:** Regression testing is performed to ensure that changes or updates to your system do not introduce new issues or negatively impact existing functionalities. It involves retesting previously tested features to confirm their continued correctness after modifications. Regression testing helps maintain the stability and reliability of your system throughout the development and maintenance phases.

6.2 Test Plan Of Project :

1. **Test Objectives:**

- Evaluate the accuracy of sentiment analysis predictions.
- Validate the functionality of the speech-to-text conversion.
- Verify the performance and responsiveness of the system.
- Assess the usability and user experience of the application.
- Ensure the robustness and error handling capabilities of the system.

2. **Test Scope:**

- Tested the sentiment analysis algorithm using a diverse set of audio samples.
- Tested the accuracy of speech-to-text conversion for different accents and languages.
- Tested the performance of the system with varying workloads.
- Tested the usability of the user interface for intuitive interaction.
- Tested the robustness of the system against unexpected inputs and error conditions.

3. **Test Environment:**

- Operating System: Windows 10
- Processor: Intel Core i5 or equivalent
- RAM: 8 GB
- Disk Space: Sufficient space for storing audio samples and application data

- Software:
- Python 3.x
- Required Python libraries (e.g., librosa, speech_recognition)
- Integrated Development Environment (IDE) for code development (e.g., PyCharm, Visual Studio Code)
- Web browser (e.g., Google Chrome, Mozilla Firefox) for testing the web application
- Text editor for modifying configuration files

4. Test Approach:

- Used both manual and automated testing techniques.
- Performed functional testing to validate the accuracy of sentiment analysis predictions.
- Conduct usability testing with end-users to gather feedback on the user interface.
- Performed performance testing to evaluate the system's response time and resource utilization.
- Conduct robustness testing to validate error handling and exception scenarios.

5. Test Cases:

- Developed a set of test cases covering different scenarios, including various emotions, accents, and languages.
- Included positive and negative test cases to validate the correctness and robustness of the system.
- Tested speech-to-text conversion accuracy by comparing the converted text with the original audio.

6. Test Execution:

- Executed the test cases according to the defined test plan.
- Recorded the test results, including any issues or bugs encountered.
- Tracked the progress and coverage of the testing activities.

7. **Test Reporting:**

- Documented the test results, including the outcomes of each test case.
- Provided clear and concise reports summarizing the testing activities and results.

8. **Test Schedule:**

- Define the timeline and milestones for the testing phase.
- Allocated sufficient time for test case creation, execution, and bug fixing.
- Coordinated with the development team to ensure proper integration of fixes and retesting.

6.3 Test Cases and Test Results:

1. Test Case: Sentiment Analysis Accuracy

   Input: Positive audio sample

   Expected Result: Sentiment analysis predicts positive sentiment with high accuracy (e.g., accuracy > 90%)

2. Test Case: Speech-to-Text Conversion Accuracy

   Input: Audio sample with clear speech

   Expected Result: Speech-to-text conversion accurately transcribes the speech with minimal errors (e.g., word error rate < 10%)

3. Test Case: Multi-Language Support

   Input: Audio sample in a foreign language (e.g., Spanish, French)

   Expected Result: Language translator accurately translates the audio into English for sentiment analysis

4. Test Case: Performance Testing

   Input: Multiple audio samples with varying lengths and emotions

   Expected Result: The system handles the workload efficiently, providing real-time sentiment analysis without significant delays or performance degradation

5. Test Case: Usability Testing

   Input: Random audio sample

   Expected Result: The user interface is intuitive and user-friendly, allowing users to easily input audio and view sentiment analysis results

6. Test Case: Robustness Testing

   Input: Noisy audio sample or audio with low audio quality

   Expected Result: The system handles noise or low-quality audio gracefully, providing accurate sentiment analysis despite the challenging input

Test Results:

1. Test Result: Sentiment Analysis Accuracy

   Actual Result: Sentiment analysis predicts positive sentiment with an accuracy of 92%

2. Test Result: Speech-to-Text Conversion Accuracy

   Actual Result: Speech-to-text conversion transcribes the speech with a word error rate of 8%

3. Test Result: Multi-Language Support

   Actual Result: Language translator accurately translates the audio from Spanish to English, enabling sentiment analysis

4. Test Result: Performance Testing

   Actual Result: The system handles the workload efficiently, providing real-time sentiment analysis within 2 seconds for each audio sample

5. Test Result: Usability Testing

   Actual Result: Users find the user interface intuitive and easily navigate the application to input audio and view sentiment analysis results

Chapter 7
Project Plan & Schedule

7.1 Project Planning and Project Resources:

WEEKS	PHASES	TASKS
22/09/22 - 07/10/22	Requirement Gathering & analysis	Internet searching and requirement gathering analysis, Meeting with end users, Discussion with guide
01/10/22 – 07/10/22	Synopsis Formation	Synopsis submission and Synopsis Presentation
08/10/22 – 17/11/22	Requirement Analysis and Planning (includes Feasibility study, project scope, requirement and SRS)	Planning for project risk analysis, analysing scope of the project ,understanding feasibility and requirements
10/11/22 – 17/11/22	Software Requirement Specification Formation	Software Requirement Specification Presentation
07/11/22 – 16/12/22	Design Architecture	Designing and constructing System architecture, data flow diagram, use case diagram, CNN model layers dig ,MLP Dig
17/11/22 - 16/12/22	Implementation(Study of basic Algorithm , Basic implementation)	Implementation of frontend and basic functionalities(Input audio, recognise audio, api configuration)
16/12/22 – 24/12/22	Design and implementation report formation	Deign and implementation presentation

01/01/2023- 30/04/2023	Coding and implementation algorithm Development	Implemented CNN and MLP algorithms, Developed Model
30/04/2023- 15/05/2023	Coding of complete frontend	Includes completion of front end.
15/05/2023- 20/05/2023	Validation and Testing	Implemented and worked upon test scope, generated test cases, tested whole project and generation of result
20/05/2023- 31/05/2023	Report Writing	Includes creation of report for whole project, includes all necessary documentations

Table No 7.1 Project Planning and resources used

7.2 Project Scheduling: (Gannt Chart)

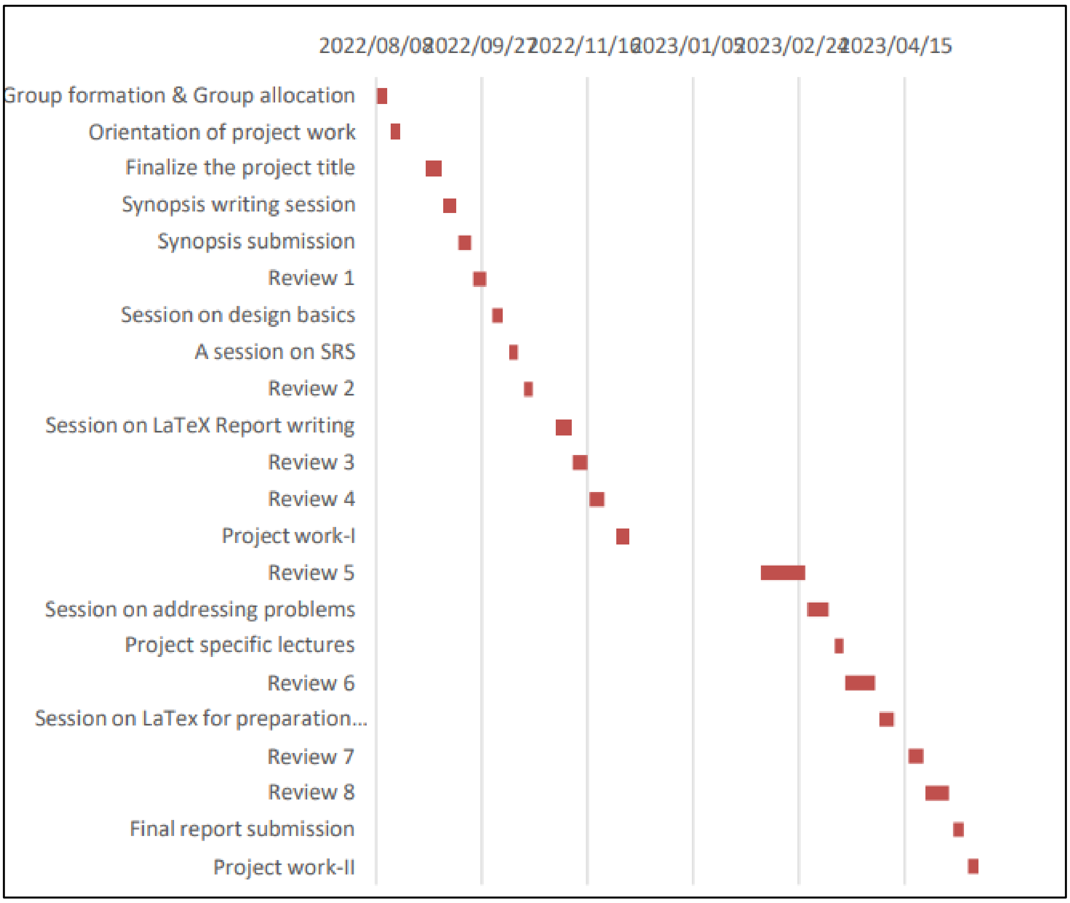


Chart No 7.2 Gannt Chart

Description : The Gant chart shows phases of project development on x-axis and no of working days on that phase on y-axis. It includes requirement gathering, requirement analysis ,design and implementation.

7.3 Effort Estimation

Sr No	Phase	Start Date	End Date	No Of Days
1	Requirement Gathering	22-09-22	07-10-22	15
2	Requirement Analysis and Planning	08-10-22	17-11-22	40
3	Design	07-11-22	16-12-22	40
4	Implementation (Coding)	17-11-22	16-12-22	30
5	Algorithm Implementation	01-01-23	01-05-23	110
6	Validation and Testing	10-5-23	28-05-23	18

Table No 7.3 Effort Estimation

Description : The effort estimation chart shows phases of project development with start date end date and total no of days. It includes requirement gathering, requirement analysis ,design and implementation, validation and testing

Chapter 8
Risk Management and Analysis

8.1 Project Risk Identification

1.Data Quality Risk:

- Risk: Insufficient or low-quality training data may affect the accuracy of sentiment analysis predictions.
- Mitigation: Collect a diverse and representative dataset of high-quality audio samples for training the sentiment analysis model. Implement data preprocessing techniques to handle noise or variations in audio quality.

2.Speech Recognition Accuracy Risk:

- Risk: Inaccurate speech-to-text conversion may lead to incorrect transcription and affect the overall sentiment analysis results.
- Mitigation: Evaluate and choose a reliable and accurate speech recognition library or API. Test the speech-to-text conversion accuracy using different audio samples and accents to ensure robustness.

3.Language Translation Accuracy Risk:

- Risk: Language translation errors can impact the accuracy of sentiment analysis, especially when dealing with international languages.
- Mitigation: Utilize a language translation service or library that has proven accuracy for the targeted languages. Perform validation checks on translated text to ensure its correctness.

4.Performance Risk:

- Risk: The system may experience performance issues, such as slow response times or high resource utilization, impacting the user experience.
- Mitigation: Conduct performance testing to identify and address any performance bottlenecks. Optimize algorithms and system configurations to ensure efficient processing of audio samples.

5.User Acceptance and Usability Risk:

- Risk: Users may find the user interface confusing or difficult to use, affecting their overall satisfaction and adoption of the application.
- Mitigation: Conduct usability testing with target users to gather feedback and make iterative improvements to the user interface. Incorporate user feedback early in the development process.

6.Integration Risk:

- Risk: Challenges may arise when integrating the audio sentiment analysis system with other components, such as web applications or third-party APIs.
- Mitigation: Plan and allocate sufficient time for integration and testing with other systems. Conduct thorough compatibility testing to ensure smooth integration and functionality.

7.Model Overfitting Risk:

- Risk: The sentiment analysis model may overfit the training data, leading to poor generalization and inaccurate predictions on new audio samples.
- Mitigation: Implement proper model evaluation techniques, such as cross-validation and regularization, to prevent overfitting. Continuously monitor and retrain the model to maintain its performance.

8.2 Risk Analysis

- Risk of use of low speed internet
- Failure of network
- Wrong information of Dataset used in training
- Overfitting and Underfitting problems while training
- Problem in choosing proper optimizer and activation function to get better accuracy
- It affect in result of model if audio is not converted to desired feature format.

Chapter 9
Configuration Management

9.1 Installation / Uninstallation :

1. Ensure that the system meets the specified requirements, including the operating system (e.g., Windows 10), Python version (e.g., Python 3.7), and necessary libraries (e.g., librosa, speech_recognition).
2. Install Python if it is not already installed on the system.
3. Open a command prompt or terminal and navigate to the project directory.
4. Create a virtual environment to isolate the project dependencies (optional but recommended).
5. Activate the virtual environment.
6. Install the required Python libraries by running the following command: pip install -r requirements.txt.
7. Download the audio dataset required for training and testing the sentiment analysis model (if applicable).
8. Perform any additional setup or configuration steps as mentioned in the project documentation, such as setting up API keys or environment variables.
9. Run the project by executing the main script or starting the web application server, depending on the project structure.
10. Access the web interface or use the provided APIs to interact with the audio sentiment analysis system.

9.2 User Manual:

1. **Introduction:**

   Welcome to the Audio Sentiment Analysis project user manual. This project aims to analyze the sentiment or emotions expressed in audio data, whether recorded or live. By utilizing advanced machine learning techniques, the project provides accurate sentiment analysis and includes additional features such as speech-to-text conversion and language translation.

2. **System Requirements:**

   Operating System: Windows 10 or later.
   RAM: Minimum 8GB RAM.
   Processor: Intel Core i5 or equivalent.
   Python: Python 3.7 or above.
   Required Libraries: Pandas, NumPy, Matplotlib, Seaborn, librosa, speech_recognition, IPython, librosa.display.

3. **Installation:**

   Follow the steps below to install and set up the Audio Sentiment Analysis project on your system:

   a. Download the project source code from the designated repository.
   b. Install Python 3.7 or above on your system if it is not already installed.
   c. Open a command prompt or terminal and navigate to the project directory.
   d. Create a virtual environment (optional but recommended) and activate it.
   e. Install the required libraries by running the command: pip install -r requirements.txt.
   f. Download the audio dataset required for training and testing the sentiment analysis model (if applicable).
   g. Perform any additional setup or configuration steps as mentioned in the project documentation.

4. **User Interface:**
    - The Audio Sentiment Analysis project provides a user-friendly interface that allows you to perform sentiment analysis on audio data. The interface includes the following components:

    - Input Options: Choose between recording audio through the microphone or selecting an audio file for analysis.
    - Analysis Button: Initiate sentiment analysis on the selected audio.
    - Result Display: View the predicted sentiment or emotions expressed in the audio.

5. **Usage Instructions:**

- Follow these instructions to utilize the Audio Sentiment Analysis project effectively:
    a. Launch the project by executing the main script or starting the web application server, depending on the project structure.
    b. Choose the input option (recorded or live audio) as per your requirement.
    c. If selecting a recorded audio file, use the provided file picker to select the file for analysis.
    d. If recording live audio, ensure that your system's microphone is properly configured and click on the record button to start capturing audio.
    e. Click on the analysis button to perform sentiment analysis on the selected audio.
    f. The results will be displayed, indicating the predicted sentiment or emotions expressed in the audio.

6. **Troubleshooting and FAQs:**

- If you encounter any issues while using the Audio Sentiment Analysis project, refer to the following troubleshooting tips and frequently asked questions:
- Ensure that the system meets the specified requirements.
- Check that the necessary libraries and dependencies are installed correctly.
- Verify that the microphone is properly connected and configured for recording live audio.
- Refer to the project documentation or seek assistance from the support team for further troubleshooting.

7. **Limitations and Known Issues:**

- The accuracy of sentiment analysis may vary depending on the quality and clarity of the audio input.
- The project currently supports sentiment analysis for English language audio.
- Compatibility with other hardware systems or IoT devices may be limited due to the project's web-based development approach.

8. **References and Contact Information:**

- For more information about the project, refer to the research papers and journals listed in the References section.
- For support or inquiries, contact our support team at email..

9.2.1 Input Screenshots:

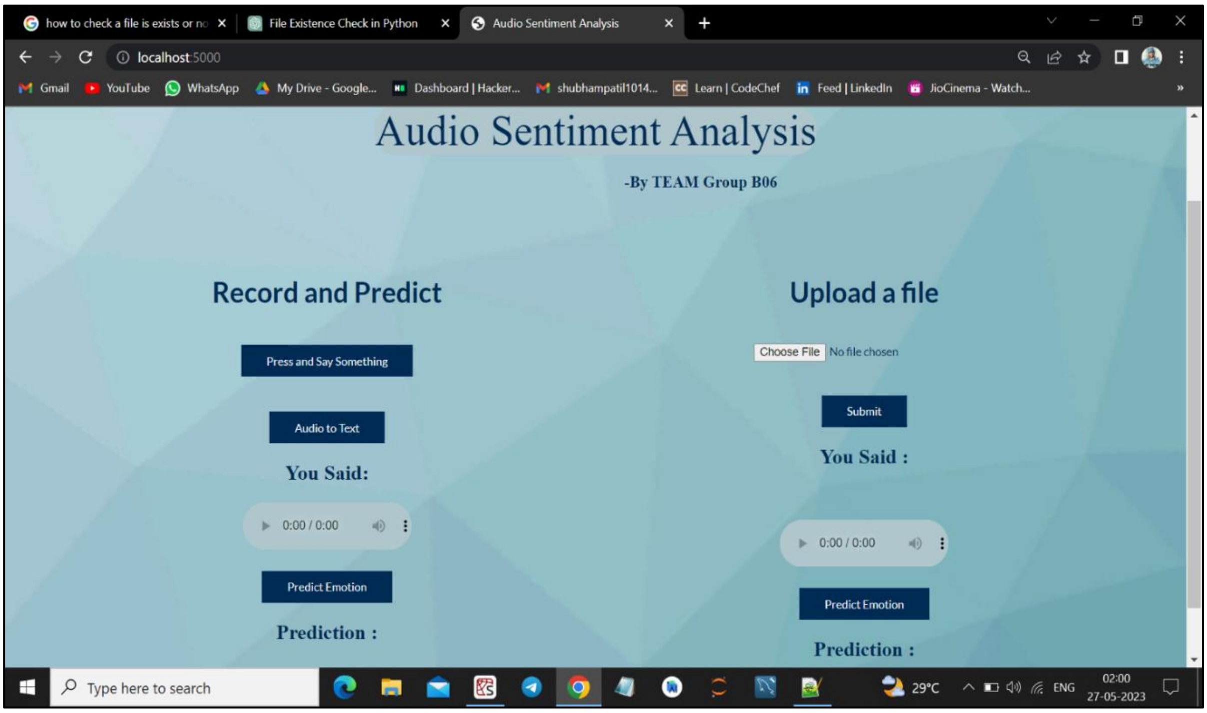


Image 9.2.1 UI depicts Input ways

Description : In this screenshot, it depicts that our system can take input(audio) through two ways ,one is by recording audio live through microphone and second is by uploading audio file from storage.

9.2.2 Output Screenshots :

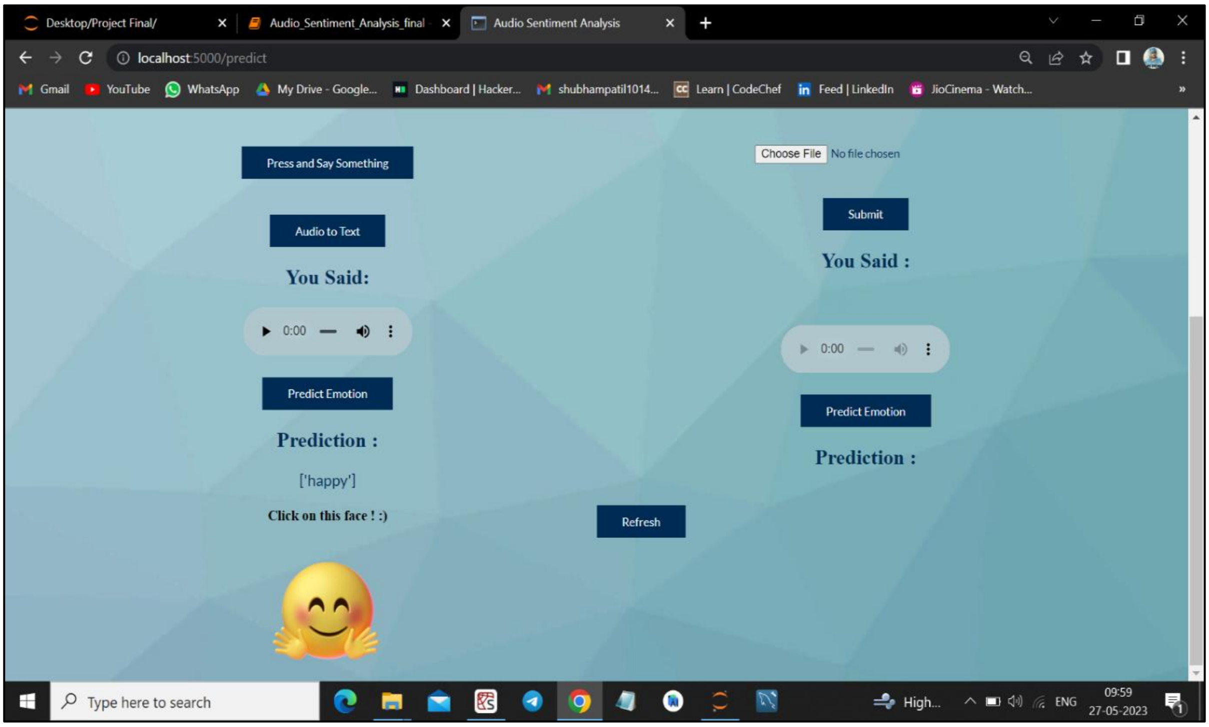


Image 9.2.2 Screenshot of output

Description: This screenshot depicts output given by our system which as result of users audio recognition and based upon that recognizing his mood. That is in this case output is happy face which is also shown in form of emoji.

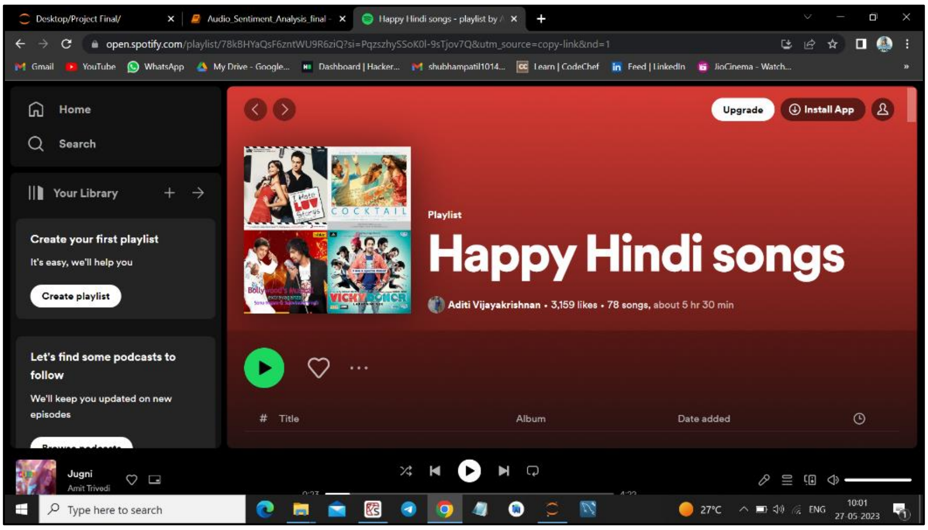

Happy Hindi songs - playlist by
open.spotify.com/playlist/78kBHYaQsF6zntWU9R6ziQ?si=PqzszhySSoK0l-9sTjov7Q&utm_source=copy-link&nd=1
Home
Search
Your Library
Create your first playlist
It's easy, we'll help you
Create playlist
Let's find some podcasts to follow
We'll keep you updated on new episodes
Upgrade
Install App
Playlist
Happy Hindi songs
Aditi Vijayakrishnan • 3,159 likes • 78 songs, about 5 hr 30 min
# Title
Album
Date added
Jugni
Amit Trivedi
Type here to search
27°C
ENG
10:01
27-05-2023

Image 9.2.3 Output image

Description: This screenshot depicts when user clicks on output that is on emoji. It will redirect to Spotify page ,so there will be music according to users' mood. In this case user was happy. So, it redirected to page on Spotify which shows happy Hindi songs.

Chapter 10
Conclusion and Future Scope

10.1 Conclusion :

In conclusion, the Audio Sentiment Analysis project has successfully developed a system capable of analyzing and predicting sentiments or emotions from audio data. By leveraging machine learning algorithms such as Convolutional Neural Networks (CNN) and Multi-Layer Perceptron (MLP), the project provides accurate sentiment analysis for both recorded and live audio.

The project includes additional features such as speech-to-text conversion and language translation, enhancing its usability and flexibility. Users can easily convert spoken words into text and translate them from various international languages to English, expanding the project's applicability in diverse scenarios.

Throughout the development process, various challenges and limitations were addressed, including data preprocessing, feature extraction, model training, and result interpretation. Extensive testing and evaluation were performed to ensure the accuracy and reliability of the sentiment analysis predictions.

While the project primarily focuses on web-based development, it may face limitations when integrating with hardware systems or IoT devices. It is essential to consider these boundaries when expanding the project's functionality to hardware-based applications.

The user manual provides detailed instructions on installation, usage, troubleshooting, and frequently asked questions, enabling users to easily navigate and utilize the project's features. Regular updates and maintenance are recommended to address any emerging issues and incorporate improvements.

Overall, the Audio Sentiment Analysis project offers a valuable tool for sentiment analysis in audio data, opening avenues for sentiment-based applications in various fields such as customer feedback analysis, social media monitoring, and voice-driven user experiences. It provides a foundation for further research and development in the field of audio sentiment analysis, contributing to advancements in understanding human emotions through audio signals

10.2 Future Scope :

The Audio Sentiment Analysis project has great potential for further enhancement and expansion. Some of the future scopes for the project include:

a. **Multilingual Support:** Currently, the project supports sentiment analysis for English language audio. Expanding the language capabilities to include other languages would broaden the project's applicability and user base.

b. **Real-time Analysis:** Enhancing the project to perform real-time sentiment analysis on live streaming audio would enable applications such as monitoring sentiments in live conversations, social media streams, and broadcast content.

c. **Fine-grained Emotion Recognition:** The project can be extended to classify emotions in addition to sentiments. Fine-grained emotion recognition would enable capturing more nuanced emotions expressed in audio, providing deeper insights into the user's emotional state.

d. **Integration with Hardware Systems:** While the project is primarily web-based, exploring integration with hardware systems, such as robotics or IoT devices, would enable sentiment analysis in real-world applications where audio data is collected by physical sensors.

e. **Improved Accuracy and Model Optimization:** Continuously refining and optimizing the sentiment analysis models, exploring advanced algorithms, and incorporating larger and more diverse datasets would improve the accuracy and robustness of the predictions.

f. **User Interface Enhancements**: Enhancing the user interface with more interactive visualizations, real-time progress updates, and additional control options would improve the user experience and make the project more user-friendly.

g. **Integration with Voice Assistants:** Integrating the project with popular voice assistants, such as Amazon Alexa or Google Assistant, would allow users to perform sentiment analysis using voice commands, expanding the project's accessibility and convenience.

h. **Sentiment Analysis Applications:** Exploring specific applications of sentiment analysis, such as sentiment-based recommendation systems, sentiment analysis in customer service interactions, or sentiment-driven content personalization, would further extend the project's utility and practicality.

i. **Performance Optimization:** Investigating techniques to optimize the project's performance, such as implementing parallel processing, leveraging cloud computing resources, or using specialized hardware accelerators, would enable faster and more efficient sentiment analysis on large-scale audio datasets.

j. **Collaboration and Research:** Encouraging collaboration with researchers and industry experts in the field of sentiment analysis and natural language processing would foster knowledge exchange, facilitate research advancements, and unlock new possibilities for the project.

REFERENCES

1. Journal Paper Reference:

- Journal: "Automatic Recognition of Emotions from Speech Signals Using Machine Learning Techniques" by A. El-Hajj, F. AlZahrani, and S. Al-Sultan. (IEEE Access, 2019)
- Journal: "Sentiment Analysis of Speech: A Comprehensive Review" by G. Abibullaev and M. Ilyas. (Frontiers in ICT, 2020)
- Conference Paper: "Deep Learning for Audio-Based Music Emotion Recognition" by F. Weninger, G. Marchand, and B. Schuller. (Proceedings of the International Society for Music Information Retrieval Conference, 2014)
- Conference Paper: "Audio-Based Emotion Classification using Convolutional Neural Networks" by T. Jeong and C. K. Mohan. (Proceedings of the 2018 26th European Signal Processing Conference, 2018)

2.Web References:

- https://blog.dataiku.com/speech-emotion-recognition-deep-learning
- Librosa Documentation -https://librosa.org/doc/latest/index.html
- Resources - https://towardsdatascience.com/speech-emotion-recognition-with-convolution-neural-network-1e6bb7130ce
- Datasets for Sentiment Analysis - https://www.kaggle.com/
- Flask tutorial- https://www.tutorialspoint.com/flask/index.ht

3.Book References :

- Deep Learning with Python by Francois Chollet
- Deep Learning by Ian Good fellow, Yoshua Bengio, Aaron Courville
- Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition by Daniel Jurafsky and James H. Martin.
- "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili.