**A PROJECT REPORT**

**On**

**“Mentalytics : AI for Mental Health Monitoring”**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfilment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**INFORMATION TECHNOLOGY**

**BY**

**AAKASH KUMAR**

**KRISHANU ROY**

**ROSHAN ARYAN**

**SOUVIK BASAK**

**SUVANKAR PANIGRAHI**

**VIJAY VISHAL** 2105511

2105550

2105570

2105583

2105585

2105594

**UNDER THE GUIDANCE OF**

**Prof. Santwana Sagnika**



**SCHOOL OF COMPUTER ENGINEERING**

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



CERTIFICATE

**“Mentalytics : AI for Mental Health Monitoring”**

Submitted by

**AAKASH KUMAR**

**KRISHANU ROY**

**ROSHAN ARYAN**

**SOUVIK BASAK**

**SUVANKAR PANIGRAHI**

**VIJAY VISHAL**

**2105511**

**2105550**

**2105570**

**2105583**

**2105585**

**2105594**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2024-2025, under our guidance.

Date: 30 / 11 / 2024

**Prof. Santwana Sagnika**

Project Guide

**Acknowledgements**

We feel immense pleasure and feel privileged in expressing our deepest and most sincere gratitude to our supervisor Professor Santwana Sagnika, for her excellent guidance throughout our project work. Her kindness, dedication, hard work, and attention to detail have inspired us greatly. Our heartfelt thanks to you mam for the unlimited support and patience shown to us. We would particularly like to thank her for all her help in patiently and carefully correcting all our manuscripts during our course of Major project works in the final year of our undergraduate course.

AAKASH KUMAR

KRISHANU ROY

ROSHAN ARYAN

SOUVIK BASAK

SUVANKAR PANIGRAHI

VIJAY VISHAL

**ABSTRACT**

This research proposes an innovative AI-powered system designed to continuously monitor and assess an individual's emotional state by analyzing multimodal data, including facial expressions, voice patterns, and textual content, to provide accurate emotion detection. Utilizing advanced machine learning techniques, the system aims to classify a wide array of emotions, such as happiness, sadness, anger, fear, and surprise, through the integration of various data sources. The system collects high-resolution facial images, audio recordings, and textual content from multiple platforms. Preprocessing steps will involve data cleaning to eliminate noise and inconsistencies, followed by normalization to ensure a uniform scale across different data types. Key features will be extracted from the preprocessed data, including facial landmarks, texture, and action units for facial features, Mel-Frequency Cepstral Coefficients (MFCCs), pitch, and intensity for acoustic features, and sentiment analysis, emotion lexicons, and word embeddings for textual content. Convolutional Neural Networks (CNNs) will be employed to process facial images and identify relevant features for emotion classification, while Recurrent Neural Networks (RNNs) will be used to capture temporal dependencies in audio and text data to model emotional evolution over time. Natural Language Processing (NLP) techniques will be applied to analyze textual data for sentiment, emotion, and intent. A multimodal fusion approach will integrate these diverse data modalities to enhance the system’s accuracy and robustness, enabling more reliable emotion detection. The models will be trained on large and diverse datasets to ensure high performance and generalization. Once operational, the system will provide real-time processing of emotional data, offering immediate feedback based on the detected emotional state. This feedback will be personalized to the individual’s emotional profile, including recommendations for relaxation techniques, mindfulness exercises, or professional counseling. Additionally, the system will include an early warning feature to alert users to potential mental health concerns, such as depression or anxiety, facilitating timely intervention. By merging cutting-edge AI techniques with a comprehensive understanding of human emotions, this research aims to develop a highly effective tool to improve mental health and overall well-being, providing both personalized support and early intervention to foster a better quality of life.



**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Chapter 1 : Introduction | |  |
|  | 1.1 | Motivation |  |
|  | 1.2 | Background Studies /Literature Survey |  |
|  | 1.3 | Objectives |  |
| 2 | Chapter 2 : Challenges and Remedy Methodology | |  |
|  | 2.1 | Challenges |  |
|  | 2.2 | Remedies of Challenges |  |
| 3 | Chapter 3 : Methodology | |  |
|  | 3.1 | Problem Statement |  |
|  | 3.2 | Applied Tools and Techniques |  |
| 4 | Chapter 4 : Implementation | |  |
|  | 4.1 | Facial Model Recognition |  |
|  | 4.2 | Textual Analysis |  |
|  | 4.3 | Speech & Voice Analysis |  |
|  | 4.4 | Multi Modal Analysis |  |
| 5 | Chapter 5 : Backend & Frontend | |  |
|  | 5.1 | Procedure - Backend |  |
|  | 5.2 | Frontend Overview |  |
|  | 5.3 | Result Obtained |  |
| 6 | Conclusion | |  |
| 7 | Individual Contribution | |  |
| 9 | Plagiarism Report | |  |



Chapter 1

Introduction

1.1 Motivation

Mental health has emerged as a critical global concern, with conditions like stress, depression, and anxiety significantly impacting individuals' well-being and productivity. Early detection and intervention are key to effective management of these conditions. This project explores the potential of artificial intelligence (AI) to revolutionize mental health monitoring by developing an AI system capable of continuously assessing an individual's emotional state through multimodal data analysis.

The need for such a system arises from the limitations of traditional mental health assessment methods, which often rely on infrequent clinical visits and self-reported symptoms. By leveraging advanced AI techniques, this project aims to provide a proactive and personalized approach to mental health monitoring. The system will analyze various modalities, including facial expressions, voice patterns, and text inputs, to identify subtle emotional cues that may indicate potential mental health issues.

The structure of this report will follow a logical sequence, commencing with an in-depth examination of the existing challenges and limitations in current mental health assessment methods. Subsequently, it will delve into the theoretical underpinnings of applying machine learning to address these challenges, exploring relevant algorithms and methodologies. The report will then transition into a discussion of practical implementation strategies, encompassing data collection, model development, and deployment considerations. Finally, it will conclude with reflections on the potential impact of integrating AI into mental health monitoring and outline avenues for future research and development. Through this comprehensive approach, the report aims to provide a holistic understanding of the role of AI in improving mental health outcomes and its implications for healthcare.



1.2 Background Studies / Literature Review

Emotion recognition, a domain focused on identifying and interpreting human emotions, has witnessed remarkable advancements in recent years. By leveraging artificial intelligence (AI), these systems analyze various modalities such as facial expressions, voice patterns, and textual content to accurately detect and classify emotions. This capability has not only enhanced the understanding of human behavior but also opened new avenues for applications in areas like mental health monitoring and human-computer interaction.

A significant body of research has been dedicated to the development of AI systems for emotion recognition. Facial emotion recognition, for instance, utilizes computer vision techniques and deep learning models to analyze facial expressions and detect emotions such as happiness, sadness, anger, and fear. Similarly, speech emotion recognition focuses on analyzing voice patterns, including pitch, tone, and intensity, to identify emotional states, making it particularly useful in detecting stress, anxiety, and depression. In addition, text-based emotion analysis employs Natural Language Processing (NLP) techniques to assess emotions expressed in written communication, such as text messages or social media posts, categorizing them as positive, negative, or neutral.

Recent advancements in AI have facilitated the development of multimodal emotion recognition systems, which integrate information from multiple modalities to achieve greater accuracy and robustness. By simultaneously analyzing facial expressions, vocal tones, and textual content, these systems offer a more comprehensive understanding of an individual’s emotional state. This multimodal approach not only enhances the reliability of emotion detection but also addresses the limitations of single-modality systems, particularly in complex or ambiguous scenarios.

The integration of emotion recognition with AI has paved the way for innovative applications in mental health monitoring. AI-powered systems can track individuals’ emotional states continuously, providing valuable insights into their mental well-being. By analyzing patterns in emotional data over time, these systems can identify trends and even predict future emotional states. This predictive capability enables the delivery of personalized interventions, such as recommending relaxation techniques or connecting users with mental health professionals. Consequently, these systems hold promise for early detection and management of mental health issues, significantly improving the quality of care.



Despite its potential, the application of AI in emotion recognition and mental health monitoring raises important ethical concerns. Privacy and security are paramount, as these systems often handle sensitive personal data. Ensuring robust data protection measures is essential to safeguard user privacy. Additionally, bias in AI algorithms can lead to unfair outcomes, especially if the training data lacks diversity and representation. Addressing this issue requires the use of comprehensive datasets and ongoing evaluation of algorithmic fairness. Furthermore, the potential for misinterpretation of emotions poses a challenge, particularly in complex or nuanced situations. To mitigate this risk, it is imperative to develop highly accurate and context-aware emotion recognition models.



1.3 Objectives

The objective of this project is to develop and implement an AI-based system for emotion recognition to facilitate real-time mental health monitoring and intervention. Objective of the project:

1. **Multimodal Emotion Detection**: Develop an AI system capable of analyzing facial expressions, voice patterns, and text inputs to accurately detect emotions such as happiness, sadness, anger, stress, and anxiety.
2. **Model Development and Integration**: Implement and integrate machine learning and deep learning models, including CNN for facial emotion recognition, RNN/LSTM for voice emotion analysis, and a Naive Bayes classifier for textual sentiment analysis.
3. **Multimodal Emotion Recognition**: Combine data from different modalities using multimodal neural networks and Gemini API to provide a comprehensive analysis of emotional states.
4. **Real-time Feedback System**: Enable the AI to provide real-time feedback, alerts, or coping suggestions based on detected emotional changes, assisting individuals in managing their mental health.
5. **Ethical Data Handling**: Ensure robust data privacy and security measures, including encryption and anonymization, to handle sensitive user data responsibly.
6. **Applications in Diverse Settings**: Adapt the system for applications in clinical support, personal emotional monitoring, and corporate well-being, providing actionable insights to users and stakeholders.
7. **Contributions to Mental Health Care**: Contribute to mental health care by offering innovative, data-driven solutions for early detection and monitoring of stress, depression, and anxiety, ultimately fostering mental well-being.

Overall, the project aims to revolutionize mental health monitoring through the integration of advanced AI technologies, offering personalized and scalable emotional insights for various use cases.





Chapter 2



2.1 Challenges

1. **Data Quality and Quantity:**  
   Acquiring diverse and high-quality multimodal data (facial expressions, voice patterns, text inputs) can be challenging, especially for sensitive emotions like depression or anxiety.Ensuring data privacy and ethical considerations while collecting and storing personal information is crucial.
2. **Variability in Emotional Expression:**  
   Emotional expressions can vary significantly across individuals, cultures, and contexts.Developing models that can accurately interpret these variations is a complex task.
3. **Contextual Understanding:**  
   Understanding the context in which emotions are expressed is essential for accurate interpretation.Factors like social situations, cultural norms, and individual personality traits can influence emotional expression.
4. **Real-time Processing:**  
   Real-time processing of multimodal data is necessary for timely feedback and intervention.Developing efficient and accurate models that can process data in real-time is a technical challenge.
5. **Ethical Considerations:**  
   Using AI for mental health monitoring raises ethical concerns, including privacy, bias, and potential misuse of the technology.Ensuring transparency, accountability, and fairness in the development and deployment of such systems is crucial.



2.2 Remedies to the Challenges :

1. **Data Quality and Quantity:**  
   Applying techniques like rotation, flipping, and adding noise to existing data, we can artificially increase the dataset size and diversity. Leveraging generative models like Generative Adversarial Networks (GANs) can help create synthetic data that closely resembles real-world data, especially for underrepresented scenarios. Accurate and consistent labeling of data is essential for training reliable models. Employing experienced annotators and quality control measures can help ensure data quality.
2. **Variability in Emotional Expression:**

Combining data from multiple modalities (facial expressions, voice patterns, text inputs) can provide a more comprehensive understanding of emotional states. Training models on diverse datasets that include data from different cultures can help improve their ability to recognize and interpret emotions across cultural boundaries. Adapting models to individual users' emotional patterns and preferences can enhance accuracy and effectiveness.

1. **Contextual Understanding:**  
   Incorporating relevant contextual features like time of day, location, and social interactions can help the model better understand the underlying emotional state. Utilizing NLP techniques can help analyze the semantic and emotional content of text-based inputs, providing valuable context for emotion recognition.
2. **Real-time Processing:**  
   Employing efficient algorithms and optimized hardware can reduce processing time. Leveraging cloud computing resources can provide the necessary computational power for real-time processing.
3. **Ethical Considerations:**  
   Developing models that are transparent and interpretable can help build trust and accountability. Implementing techniques to mitigate bias in data and models can ensure fair and equitable treatment of all users. Protecting user privacy by implementing strong security measures and obtaining informed consent is essential.



Chapter 3

Methodology

3.1 Problem Statement

With the increasing prevalence of mental health challenges globally, timely detection and intervention remain critical. However, traditional methods of assessing emotional and mental well-being rely heavily on self-reporting, which can be subjective and limited by stigma or lack of awareness. This project addresses the need for an objective, data-driven approach to detect emotional states and potential mental health concerns by integrating multimodal analysis of facial expressions, voice patterns, and textual sentiments.

The aim is to develop a comprehensive AI-driven system capable of analyzing multiple inputs to provide insights into an individual's emotional well-being, assisting mental health professionals and individuals in early identification of stress, anxiety, or depression.

3.2 Applied Techniques and Tools

**Data Collection**The data collection process involves multimodal data acquisition to capture diverse inputs for emotion recognition. Facial expressions will be captured using high-resolution webcams, ensuring detailed visual data for analysis. Voice recordings will be obtained through microphones to gather audio features such as tone and pitch. Textual data will be sourced from platforms like social media, emails, and chat messages to assess written communication. Additionally, physiological data, including heart rate and skin conductance, will be integrated from wearable devices like smartwatches to provide supplementary contextual information about the user's emotional state.

**Data Preprocessing**Preprocessing is essential to ensure high-quality input data for model training. Facial image preprocessing involves steps like face detection, alignment, noise reduction, and normalization. Features such as facial landmarks, geometric attributes, and texture details will be extracted. For audio data, noise reduction techniques will be applied, followed by the extraction of features like Mel-Frequency Cepstral Coefficients (MFCCs), pitch, and intensity.

Textual data will undergo cleaning by removing stop words, punctuation, and special characters, with tokenization and stemming applied to prepare it for analysis. Sentiment analysis and emotion lexicon-based feature extraction will enhance the quality of the textual inputs.

**Feature Extraction**Feature extraction focuses on deriving meaningful attributes from the collected data. Facial features will include geometric details, such as distances between landmarks, appearance-based attributes like texture and color, and action units representing specific muscle movements associated with emotions. Voice features will comprise acoustic details like pitch, intensity, and spectral centroid, along with prosodic characteristics such as rhythm and intonation. Textual features will involve sentiment analysis to identify positive, negative, or neutral sentiments and the mapping of words to specific emotions using an emotion lexicon.

**Model Selection and Training**Various machine learning and deep learning models will be employed for emotion recognition across different modalities. For facial emotion recognition, Convolutional Neural Networks (CNNs) will extract spatial features, while Recurrent Neural Networks (RNNs) will capture temporal dependencies in expressions. Speech emotion recognition will use Long Short-Term Memory (LSTM) networks for modeling long-term dependencies in audio and Hidden Markov Models (HMMs) for temporal emotion dynamics. Textual emotion analysis will utilize Support Vector Machines (SVMs), Naive Bayes classifiers, and RNNs to analyze sequential text data effectively.

**Multimodal Fusion**To improve accuracy and robustness, multimodal fusion techniques will combine data from different modalities. Early fusion will integrate features at the preprocessing stage, while late fusion will aggregate predictions from individual unimodal models. A hybrid fusion approach combining both methods will also be explored to optimize performance.

**Model Evaluation**The models will be evaluated using comprehensive metrics, including accuracy, precision, recall, and F1-score. A confusion matrix will provide detailed insights into classification performance. Cross-validation will assess the models’ generalization abilities, while user studies will evaluate the usability and effectiveness of the system.

Chapter 4



Implementation

4.1 Facial Model Recognition

**Convolutional Neural Networks (CNNs): Detailed Explanation**

CNNs are specialized deep learning models designed for processing grid-like data, such as images. Inspired by human visual processing, they are widely used in tasks like image classification, object detection, and segmentation.

### Key Intuition Behind CNNs

1. **Local Connectivity**:
   * Focuses on small regions of the input, reducing parameters while capturing spatial features like edges and textures.
2. **Hierarchical Feature Learning**:
   * Learn simple features (edges) in early layers and complex patterns (objects) in deeper layers.
3. **Parameter Sharing**:
   * Uses shared weights (filters) across the input, ensuring location-invariant feature detection.

**Architecture of CNNs**

1. **Convolution Layer**:
   * Extracts features using filters. Each filter detects specific patterns (e.g., edges).
2. **Activation Function**:
   * Introduces non-linearity using ReLU (ReLU(x) = max(0, x)), enabling the capture of complex patterns.
3. **Pooling Layer**:
   * Reduces feature map size (e.g., Max Pooling), retaining essential information while reducing overfitting.
4. **Fully Connected Layer**:
   * Maps flattened features to output classes for final predictions.



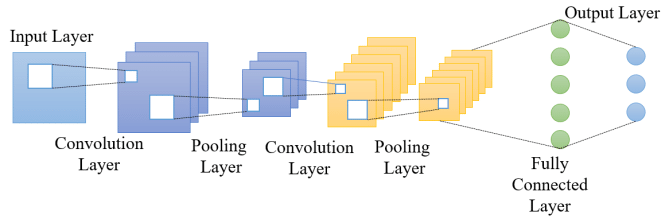
### Working of CNN: Step-by-Step

1. **Input Image**: Processes images layer by layer.
2. **Convolution**: Applies filters to detect patterns.
3. **Activation**: Introduces non-linearity.
4. **Pooling**: Summarizes data by reducing spatial dimensions.
5. **Stacking Layers**: Repeats convolution, activation, and pooling for hierarchical learning.
6. **Flattening & Output**: Final features are flattened and mapped to output labels.



### Diagram of CNN in Detail

Let me now generate a detailed diagram for this process.



### Key Parameters in CNNs

1. **Kernel Size**: Determines the size of the filter (e.g., 3×33 \times 33×3).
2. **Stride**: Determines how far the filter moves across the input.
3. **Padding**: Adds zeros around the input to preserve spatial dimensions.
4. **Number of Filters**: Defines the number of feature maps extracted.

### Backpropagation in CNNs

1. **Forward Pass**:
   * Compute the output layer values layer-by-layer using convolution, activation, and pooling.
2. **Loss Computation**:
   * Use a loss function (e.g., cross-entropy) to measure the difference between predicted and true outputs.
3. **Backward Pass**:
   * Calculate gradients of the loss with respect to weights and biases using the chain rule.
   * Gradients for convolution layers are computed for both filters and biases.
4. **Parameter Update**:
   * Use optimization algorithms (e.g., SGD, Adam) to update weights and biases.

### Advantages of CNNs

* Reduced number of parameters compared to fully connected networks.
* Effective at capturing spatial and hierarchical features.
* Translational invariance due to pooling.

### Applications of CNNs

* Image classification, object detection, medical imaging, and self-driving cars.

**Implementation of CNN in facial model recognition:**  
  
**1. Data Preparation**

* + **Image Loading**: Images are organized into labeled folders for emotions (e.g., happy, sad).
  + **Preprocessing**:
    - Convert images to grayscale and resize to 48×48 for simplicity.
    - Normalize pixel values to a range of [0, 1] to aid model learning.
  + **Label Encoding**: Emotion labels are one-hot encoded, e.g., "happy" → [0, 0, 0, 1, 0, 0, 0].

### 2. CNN Architecture

* + **Input Layer**: Takes 48×48×1 grayscale images.
  + **Convolutional Layers**:
    - Extract features using filters, starting with 128 filters of size 3×3 with ReLU activation.
    - Deeper layers use more filters (e.g., 256, 512) for complex features.
  + **Pooling Layers**: Max pooling (2×2) reduces dimensions, e.g., 48×48 → 24×24, for efficiency and reduced overfitting.
  + **Dropout Layers**: Randomly deactivate neurons during training (40% in convolutional layers, 30% in dense layers) to avoid overfitting.
  + **Flatten Layer**: Converts feature maps into a 1D vector for dense layers.

### 3. Dense Layers

* + **Layer 1**: 512 neurons with ReLU activation to combine features.
  + **Layer 2**: 256 neurons for further refinement.
  + **Output Layer**: 7 neurons (one per emotion class) with softmax activation to provide probabilities for each class.

**4. Model Compilation and Training**

* + **Loss Function**: Categorical cross entropy to compare predicted vs. actual labels.
  + **Optimizer**: Adam for adaptive learning rates and faster convergence.
  + **Metrics**: Accuracy used to evaluate model performance during training/testing.

### 5. Evaluation and Prediction

* + **Testing**: Validate on unseen data to assess generalization.
  + **Prediction**: Preprocess new images and use the model to output a probability distribution for emotion classes.

### 6. Visualization and Results

* + Display predictions alongside images for interpretability.
  + Analyze misclassifications to improve performance.

**Summary of Workflow**

* Input a grayscale image (48×48).
* Pass through convolutional and pooling layers to extract hierarchical features.
* Flatten and pass through dense layers for classification.
* Final softmax layer predicts the most likely emotion with a confidence score.

This approach combines efficient feature extraction and classification to achieve accurate emotion detection.

**Output of the model:**

1. Output 1

****

In the above snapshot, we can see we have an image of a sad person.The image is passed as an input to the model and we can see the model gives a prediction that the image depicts sad emotion. We can clearly see that the model gives the correct output.

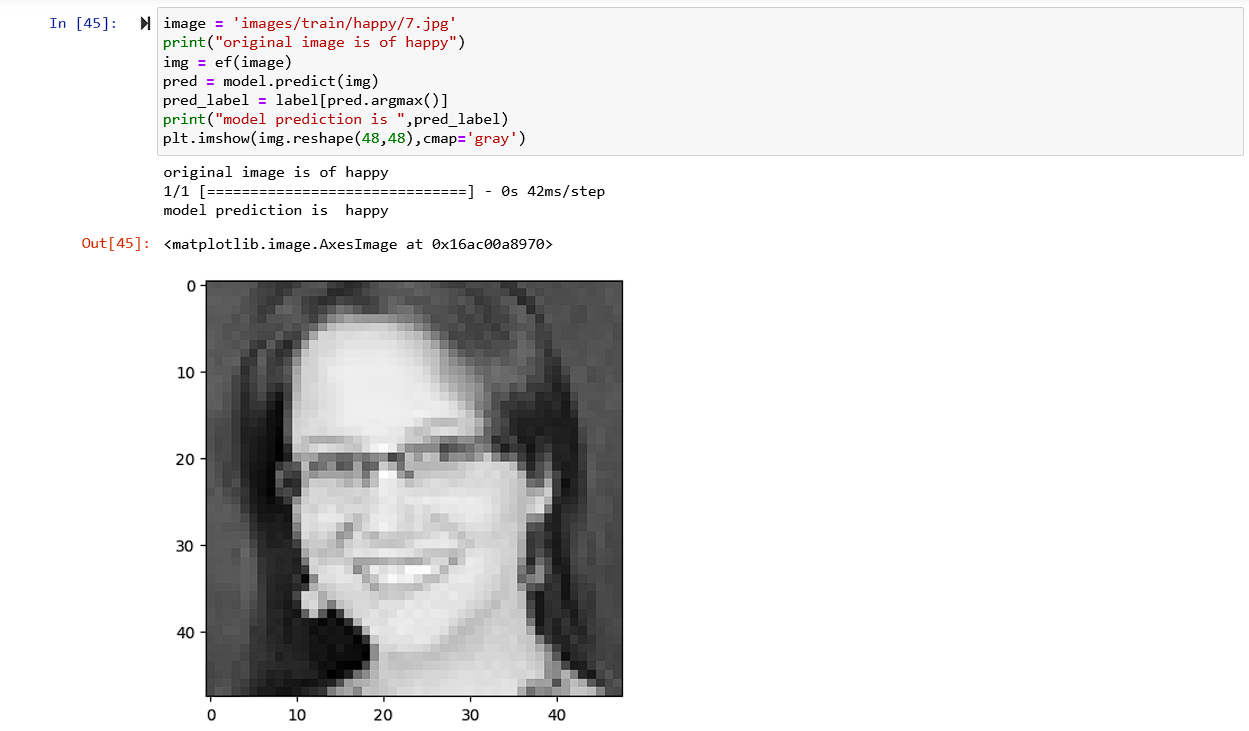
1. Output 2





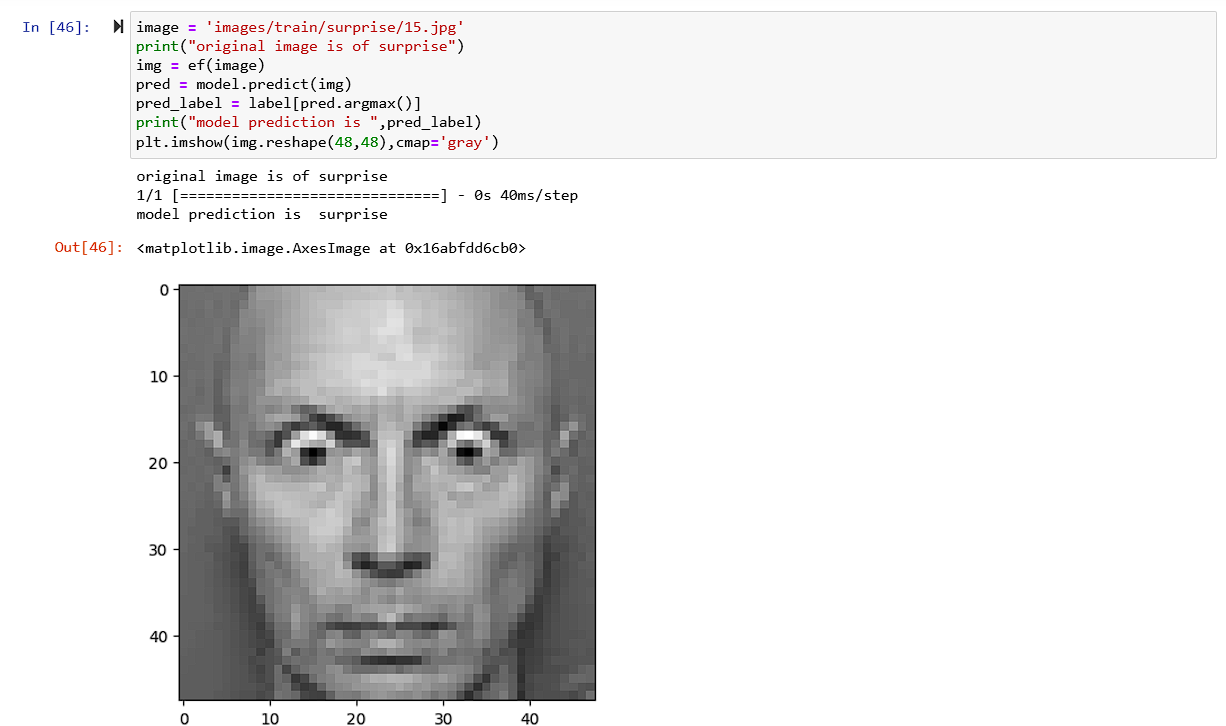
In the above snapshot, we can see we have an image of a disgusted person.The image is passed as an input to the model and we can see the model gives a prediction that the image depicts disgust emotion. We can clearly see that the model gives the correct output.

1. Output 3



In the above snapshot, we can see we have an image of a happy person.The image is passed as an input to the model and we can see the model gives a prediction that the image depicts happy emotion. We can clearly see that the model gives the correct output.

1. Output 4



In the above snapshot, we can see we have an image of a surprised person.The image is passed as an input to the model and we can see the model gives a prediction that the image depicts surprise emotion. We can clearly see that the model gives the correct output.

4.2 Textual Analysis

### Natural Language Processing (NLP) and Sentiment Analysis

**Overview**

* **NLP**: A branch of AI that enables machines to understand, process, and generate human language.
* **Sentiment Analysis**: A sub-task of NLP that identifies sentiment or emotion in text, classifying it as positive, negative, or neutral. It is widely applied in areas such as customer reviews, social media analysis, and mental health detection.

**Steps in Sentiment Analysis**

#### 1. Text Preprocessing:

* + **Tokenization**: Splits text into smaller units like words or sentences.  
     *Example*: "I love this product!" → ["I", "love", "this", "product", "!"].
  + **Lowercasing**: Standardizes text by converting to lowercase.  
     *Example*: "LOVE" → "love."
  + **Stopword Removal**: Removes common words that add little value (e.g., "the," "is").  
     *Example*: ["I", "love", "this", "product"] → ["love", "product"].
  + **Lemmatization**: Reduces words to their base form.  
     *Example*: "running," "runs" → "run."
  + **Special Character Handling**: Removes or retains emojis/punctuation based on relevance.  
     *Example*: Emojis like 😊 might indicate positive sentiment.

#### 2. Feature Extraction:

* + **Bag of Words (BoW)**: Represents text as a collection of word frequencies.  
     *Example*: "I love cats" → {"I": 1, "love": 1, "cats": 1}.
  + **TF-IDF (Term Frequency-Inverse Document Frequency)**: Highlights unique words across documents.  
     *Example*: A rare word in a document gets a higher weight than common words.
  + **Word Embeddings**: Captures semantic meaning in dense vectors.  
     *Example*: Word2Vec: "love" → [0.15, -0.67, 0.42, …].
  + **Sentence Embeddings**: Encodes full sentences for context understanding (e.g., using BERT).

#### 3.Model Building:

* + **Machine Learning Models**:
    - Logistic Regression for binary classification (positive/negative).
    - Naive Bayes is ideal for text due to independence assumptions.
    - Support Vector Machines (SVMs) for separating sentiment classes.
  + **Deep Learning Models**:
    - RNNs (LSTMs, GRUs) for sequential context.
    - CNNs for extracting local patterns in text.
    - Transformer models (e.g., BERT, GPT) use attention mechanisms to understand word context and nuances.

#### 4. Prediction:

* + Preprocessed input is converted to feature vectors.
  + Models output sentiment probabilities or labels.  
     *Example*: "I absolutely love this phone!" → Positive (98% confidence).



### Challenges in Sentiment Analysis

1. **Sarcasm**: Understanding indirect negativity.  
    *Example*: "Great, another delay."
2. **Context Sensitivity**: Sentiment changes with phrasing.  
    *Example*: "Not bad at all" → Positive.
3. **Domain Dependency**: Sentiment meaning varies across contexts.  
    *Example*: "Dark" → Positive for a thriller movie, negative for a hotel room.
4. **Multilingual Text**: Handling mixed-language inputs effectively.

**Applications**

* **E-commerce**: Analyzing product reviews.
* **Social Media**: Tracking public sentiment for campaigns or events.
* **Mental Health**: Detecting distress or well-being in user messages.

By combining robust NLP techniques and models, sentiment analysis provides actionable insights across industries.

**Implementation of Sentimental Analysis model on Whatsapp chats**

In our sentiment analysis project on WhatsApp chats, **Natural Language Processing (NLP)** and **Sentiment Analysis** techniques are applied to analyze

and categorize chat messages as **positive**, **negative**, or **neutral**. Let’s break down the process step-by-step, explaining how each part contributes to the detection of sentiments in WhatsApp messages.

### 1. Natural Language Processing (NLP) Overview

NLP is a field of AI that enables machines to understand, interpret, and manipulate human language. In our project, NLP is used to process and clean the WhatsApp chat data, allowing the sentiment analysis model to classify the messages effectively.

### 2. Sentiment Analysis with Naive Bayes Classifier

In sentiment analysis, the goal is to determine the sentiment (emotional tone) of a piece of text. Our project uses the **Naive Bayes Classifier**, a simple probabilistic classifier based on Bayes’ Theorem, to classify messages as positive or negative. Here's how the classifier is used in our project:

#### a. Data Preparation

The sentiment analysis model is trained on labeled movie reviews, where the reviews are either "positive" or "negative." In the code, we use the **movie\_reviews** dataset from NLTK, which contains a collection of positive and negative movie reviews. These reviews are pre-processed and used to train the Naive Bayes classifier.

#### b. Feature Extraction

The key to using Naive Bayes is extracting features (specific words or tokens) from the text. Each word in the text is treated as a binary feature: it either appears in the sentence (True) or it does not (False). For instance, in a sentence like "I love you," the features would be the words **"I"**, **"love"**, and **"you"**. These features are what the Naive Bayes classifier uses to make predictions about the sentiment of the message.

#### c. Training the Model

The model is trained on the features extracted from the labeled movie reviews. The Naive Bayes classifier uses these features to calculate the probability that a sentence belongs to a particular sentiment class (positive or negative). During training, the model learns the likelihood of a word appearing in a positive or negative review.

#### d. Classification of WhatsApp Chats

Once the model is trained, it can classify new, unseen text. For the WhatsApp chat analysis, the system applies the trained Naive Bayes classifier to each individual message in the chat, classifying it as either positive or negative based on the features extracted from the words in the message.

### 3. Analyzing WhatsApp Chats

#### a. Reading the Chat Data

The WhatsApp chat file is read line by line, and each message is extracted. The chat messages are cleaned to remove any unnecessary characters, and only the actual text content of the message is processed for sentiment analysis.

#### b. Sentiment Classification

For each message, the cleaned text is passed into the trained Naive Bayes classifier, which determines whether the message expresses a positive or negative sentiment. The classifier looks at the presence of specific words in the message and uses the learned probabilities to classify the sentiment.

#### c. Tracking Sentiments for Each User

Each message is associated with the name of the person who sent it. The system keeps a count of the number of positive and negative messages for each user. For example, if a user sends a message classified as **positive**, the count for that user’s positive messages is incremented.

#### d. Visualizing the Results

At the end of the analysis, the overall distribution of sentiments (positive vs. negative) is visualized using a pie chart, giving a clear view of how many messages were positive or negative. This visualization provides insight into the overall sentiment of the WhatsApp conversation.

### 4. Final Output

After processing all the messages, the system outputs the total number of positive and negative messages. It also prints out the sentiment classification for each message along with the name of the sender. Additionally, the pie chart representation gives a graphical view of the sentiment distribution across the chat.

### How NLP and Sentiment Analysis Work Together

1. **NLP Processing**:
   * **Text Cleaning**: Text data is cleaned and tokenized (split into words) to remove irrelevant data and focus on the words that carry meaning.
   * **Feature Extraction**: The cleaned words are converted into features that the model can process, with each word treated as a binary feature.
2. **Sentiment Classification**:
   * Using the features from the text, the Naive Bayes classifier computes the probability of a message belonging to either the "positive" or "negative" sentiment category.
   * The classifier applies learned patterns to predict the sentiment of new chat messages.
3. **Results Visualization**:
   * Sentiment counts for each user and the overall conversation sentiment are presented in a pie chart, which helps in interpreting the conversation's mood.

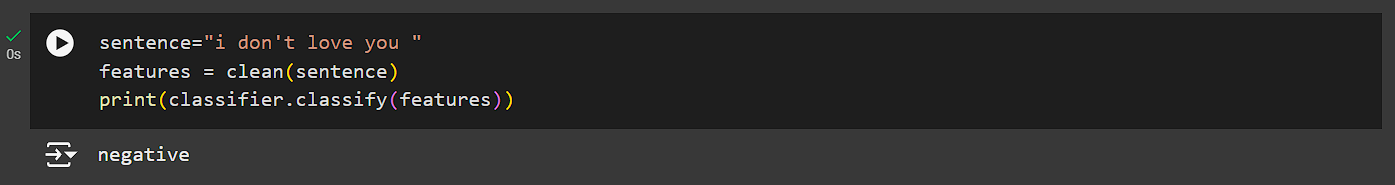
### Advantages of Using NLP and Sentiment Analysis

* **Automated Analysis**: The system automatically processes large volumes of chat data, saving time and effort.
* **Real-time Insights**: By analyzing the chat data in real-time, it provides an ongoing understanding of the emotional tone of a conversation.
* **User-specific Sentiment Tracking**: It tracks individual user sentiments, providing a detailed view of how each participant in the conversation expresses emotions.

In summary, NLP techniques like text cleaning, tokenization, and feature extraction work hand-in-hand with sentiment analysis, powered by the Naive Bayes classifier, to provide an automated solution for detecting positive and negative sentiments in WhatsApp chats.

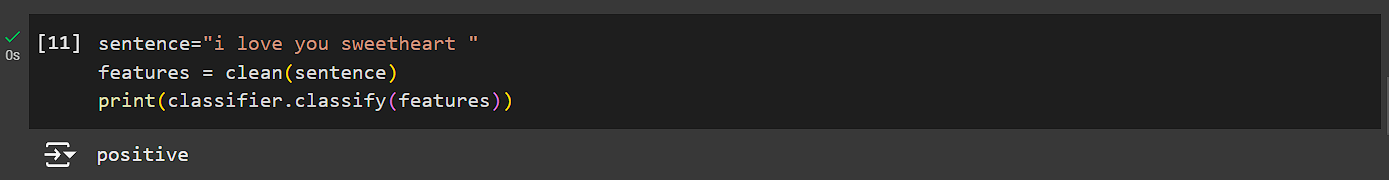
**Output of the model:**

1. Output 1



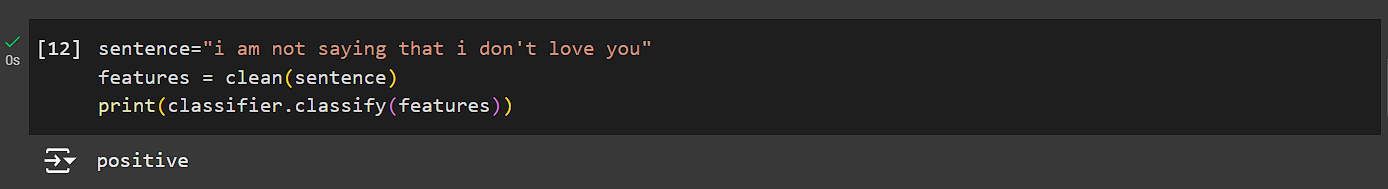
"i don't love you " clearly depicts negative sentiment and we can clearly see that the model gives a correct prediction of “negative”.

1. Output 2



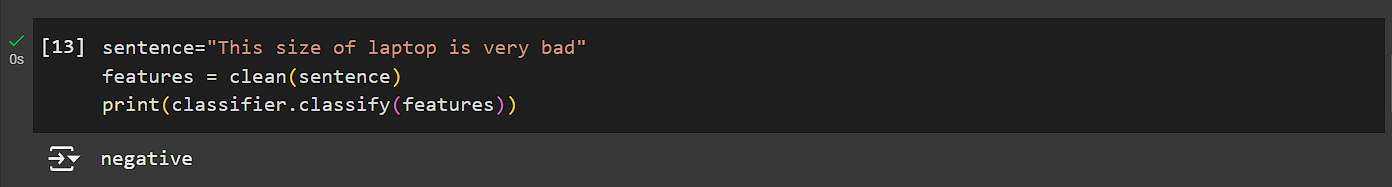
"i love you sweetheart " clearly depicts positive sentiment and we can clearly see that the model gives a correct prediction of “positive”.

1. Output 3



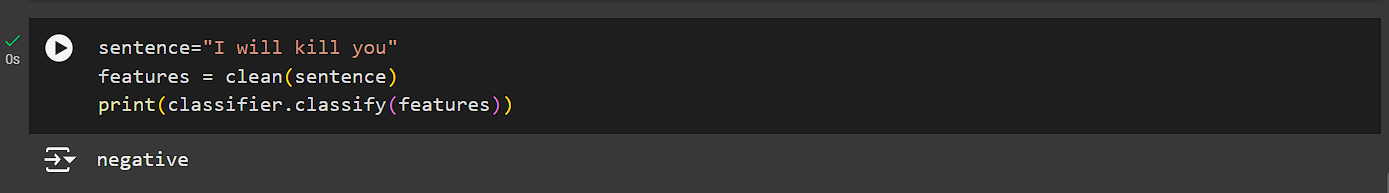
"i am not saying that i don't love you" clearly depicts positive sentiment and we can clearly see that the model gives a correct prediction of “positive”.

1. Output 4



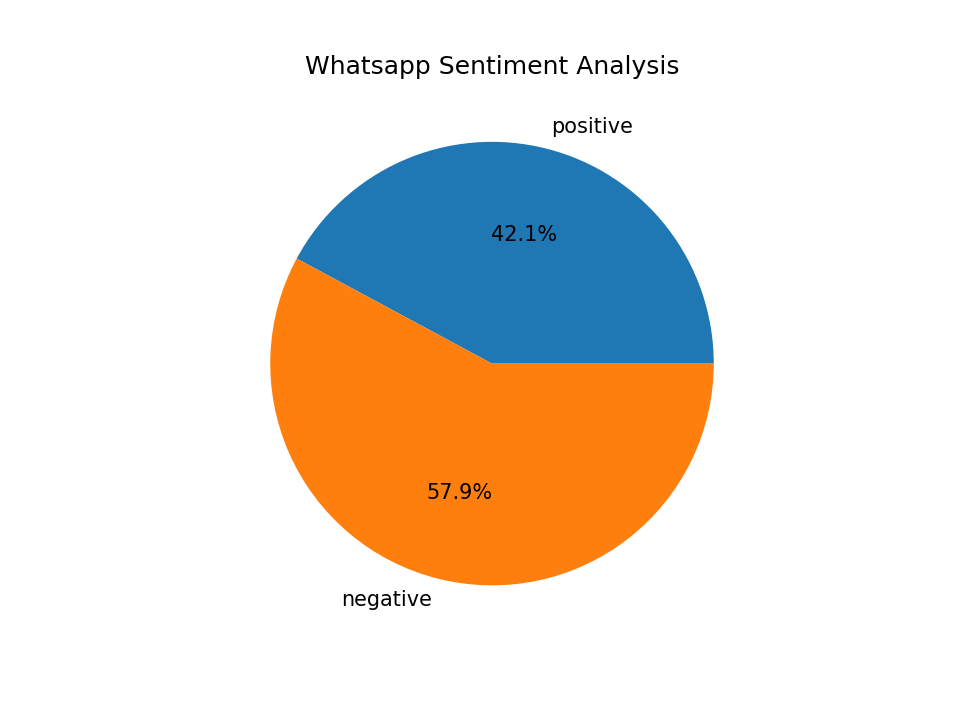
"This size of laptop is very bad" clearly depicts negative sentiment and we can clearly see that the model gives a correct prediction of “negative”.

1. Output 5



"I will kill you" clearly depicts negative sentiment and we can clearly see that the model gives a correct prediction of “negative”.

Output of the Sentimental Analysis of Whatsapp Chats:





4.3 Speech & Voice Analysis

**Librosa:**

Librosa is a Python library specifically designed for analyzing and processing audio signals. It is widely used in audio and music analysis tasks like feature extraction, audio visualization, and music classification. Librosa's flexibility and powerful API make it an essential tool for researchers and developers in audio signal processing. Librosa is designed to handle audio data and extract features such as spectrograms, chroma features, tempo, pitch, and more.

**Key Intuition Behind Librosa**

**Digital Audio Representation**:

* Audio signals are sampled at a certain rate (e.g., 44,100 Hz) and represented as a time-series signal.
* Librosa works on this time-series representation to extract meaningful features or manipulate the signal.

**Feature Extraction**:

* Audio can be represented by features such as spectrograms, chroma features, mel-frequency cepstral coefficients (MFCCs), and pitch.
* These features are crucial for tasks like audio classification, genre detection, and speech recognition.

**Mathematical Foundations**:

* Many of its functions are based on **Fourier Transforms** (for frequency analysis) and **filter banks** (e.g., mel-scale).
* Signal manipulation relies on **convolution**, **windowing**, and **time-frequency representation** (e.g., Short-Time Fourier Transform, STFT).

**Working of Librosa:**

Librosa processes audio signals represented as 1D arrays (time-domain signals). These are converted into various forms (spectrograms, chroma, MFCCs) to extract meaningful patterns. Here’s how it works step-by-step:

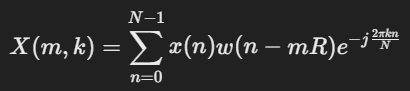
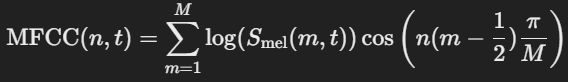
**Loading Audio Data**:

Audio files are loaded into NumPy arrays using librosa.load. The audio is resampled to a target sampling rate (default is 22050 Hz).

**Preprocessing**:

* **Trimming**: Silence is removed using librosa.effects.trim.
* **Resampling**: Rescales the signal to a specific sample rate.
* **Normalization**: Ensures the audio amplitude is consistent.

**Feature Extraction**: Librosa extracts various features that represent audio in the frequency and time domains. Key features include:

* **Spectrograms**: Representation of the frequency spectrum over time.
  + **Short-Time Fourier Transform (STFT)**: Audio signals are stationary only over short periods. STFT applies the Fourier Transform to small windows, allowing for time-frequency analysis. The STFT divides the signal into small overlapping frames and applies the Fourier Transform. 
    - X(m,k): STFT output at frame mmm and frequency bin kkk.
    - w(n): Window function (e.g., Hanning window).
    - R: Hop size (stride between frames).
  + In Librosa: librosa.stft(y).
* **Mel Spectrogram**: Human hearing is logarithmic; the Mel scale mimics this perception by spacing frequency bins logarithmically. A spectrogram mapped to the Mel scale (perceptually spaced frequencies).
  + 
    - librosa.feature.melspectrogram(y, sr=sr).
* **MFCCs (Mel-Frequency Cepstral Coefficients)**: They reduce dimensionality while retaining the most important timbral information.
  + Captures timbral texture by compressing the Mel spectrogram. 
    - librosa.feature.mfcc(y, sr=sr).

**Tempo and Beat Detection**:

* Identifies the tempo (beats per minute) and beat positions using autocorrelation and onset detection.
* Example: librosa.beat.beat\_track.

**Chroma Features**: These are compact representations of harmonic content, useful for tasks like chord recognition.

* Encodes harmonic content by mapping frequencies to 12 chromatic pitch classes.
* Computed from the STFT: 
  + librosa.feature.chroma\_stft(y, sr=sr).

**Pitch and Harmonics**:

* Pitch estimation uses algorithms like the Harmonic-Percussive Source Separation (HPSS) and autocorrelation.
* librosa.piptrack for pitch tracking.

**Visualization**:

* Librosa integrates seamlessly with Matplotlib for visualizing features like waveforms and spectrograms.

### Advantages of Librosa

1. **Ease of Use**:
   * Simple API for complex tasks (e.g., STFT, MFCC).
2. **Feature-Rich**:
   * Includes numerous features like pitch estimation, chroma features, and onset detection.
3. **Customizable**:
   * Functions allow parameter tuning (e.g., hop size, window length).
4. **Integration**:
   * Compatible with NumPy, SciPy, and machine learning libraries.
5. **Visualization Tools**:
   * Easily plot spectrograms, waveforms, and chroma diagrams.
6. **Community Support**:
   * Widely used in research and has excellent documentation.

### Applications

1. **Speech Recognition**: Feature extraction for phoneme detection.
2. **Music Analysis**: Chord recognition, tempo detection, and genre classification.
3. **Emotion Recognition**: Extract prosodic features from voice signals.
4. **Audio Classification**: Scene recognition, environmental sound analysis.

**Speech Recognition: Detailed Explanation**

The **SpeechRecognition** library is a simple yet powerful Python library used for converting spoken language into text. It provides a high-level API to perform speech recognition using various engines and APIs, such as Google Speech Recognition, IBM Watson, Sphinx, and more.

### Key Intuitions Behind Speech Recognition

1. **Speech as a Signal**:
   * Speech is an analog sound wave that is digitized into a time-series signal through **sampling**.
2. **Speech-to-Text Conversion**:
   * Speech signals are converted into text by extracting features such as phonemes, mapping them to words using language models, and applying probability-based corrections for accuracy.
3. **Mathematical Foundation**:
   * Speech recognition relies on **Hidden Markov Models (HMMs)**, **Neural Networks**, and **Dynamic Time Warping (DTW)**.
   * Modern systems use **Recurrent Neural Networks (RNNs)** and **Transformer-based models** for enhanced recognition.

**Working of SpeechRecognition Library**

#### 1. Importing and Initializing

#### 2. Capturing Audio

Audio input is taken from a microphone or an audio file.

#### 3. Recognizing Speech

The recognizer converts audio data into text using a chosen engine:

#### 4. Working with Audio Files

The library supports recognizing speech from pre-recorded files:

#### 5. Handling Noise

The adjust\_for\_ambient\_noise() function reduces background noise for better accuracy.

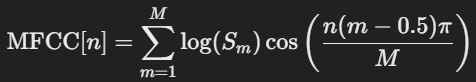
**Internal Functioning of SpeechRecognition**

**Audio Signal Processing**:

* Converts analog audio into a digital signal.
* Applies noise reduction and filtering.

**Feature Extraction**:

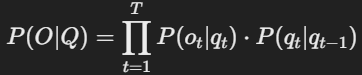
* Extracts **Mel-Frequency Cepstral Coefficients (MFCCs)**, **spectral features**, and **pitch**.
* MFCC is a representation of the short-term power spectrum based on the mel scale.

MFCC Calculation:  


* Sm​: Mel-scale filter bank energies.

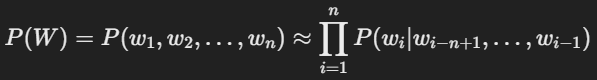
**Acoustic Modeling**:

* Maps extracted features to phonemes using **Hidden Markov Models (HMMs)**.
* HMMs model sequences of speech data as probabilistic transitions:



* + O: Observations (features).
  + Q: States (phonemes).

**Language Modeling**:

* Uses **n-grams** or deep learning models to form coherent words/sentences: 

**Decoding**:

* Finds the most likely sequence of words given the probabilities from acoustic and language models.

### Advantages of the SpeechRecognition Library

1. **Ease of Use**:
   * Simple Python API for complex speech-to-text tasks.
2. **Multi-Engine Support**:
   * Supports Google Speech Recognition, IBM Watson, Microsoft Azure, CMU Sphinx, and more.
3. **Real-Time Processing**:
   * Handles real-time audio input seamlessly.
4. **Cross-Platform**:
   * Works on Windows, macOS, and Linux.
5. **Pre-Trained APIs**:
   * Leverages advanced models from cloud services for high accuracy without the need for training.
6. **Noise Handling**:
   * Includes functions to handle background noise, improving recognition.
7. **File Compatibility**:
   * Supports multiple audio formats like WAV, AIFF, and FLAC.
8. **Customizability**:
   * Allows for tuning and using custom engines if required.



### Applications

1. **Virtual Assistants**:
   * Core functionality in Alexa, Siri, and Google Assistant.
2. **Accessibility**:
   * Enabling speech-to-text for individuals with disabilities.
3. **Voice-Based Systems**:
   * Call centers, automated response systems, and voice-activated devices.
4. **Language Learning**:
   * Provides real-time feedback for pronunciation.
5. **Data Analysis**:
   * Transcribes audio for meeting minutes or video content analysis.



**Implementation of SpeechRecognition Library in Speech & Voice Analysis**

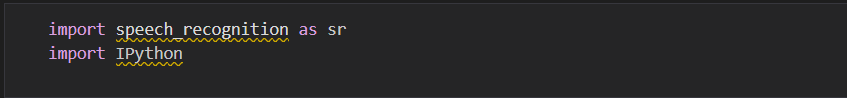
1. **Audio Acquisition:**
   * The audio files (e.g., male.wav, female.wav) are passed to the SpeechRecognition library for processing.
2. **Preprocessing:**
   * Background noise is reduced.
   * The audio is standardized for better recognition.

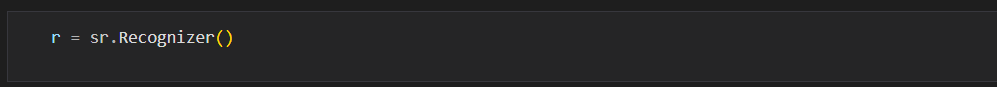


1. **Feature Extraction:**
   * Speech features are extracted using internal functions and algorithms (e.g., MFCCs, spectrograms).
2. **Speech-to-Text Conversion:**
   * The processed audio data is sent to Google’s Web Speech API.
   * The API decodes the audio into phonemes and maps them to words based on the language model.
3. **Output:**
   * The transcribed text is returned and can be analyzed further using NLP for sentiment/emotion detection.

### Working Process in Summary

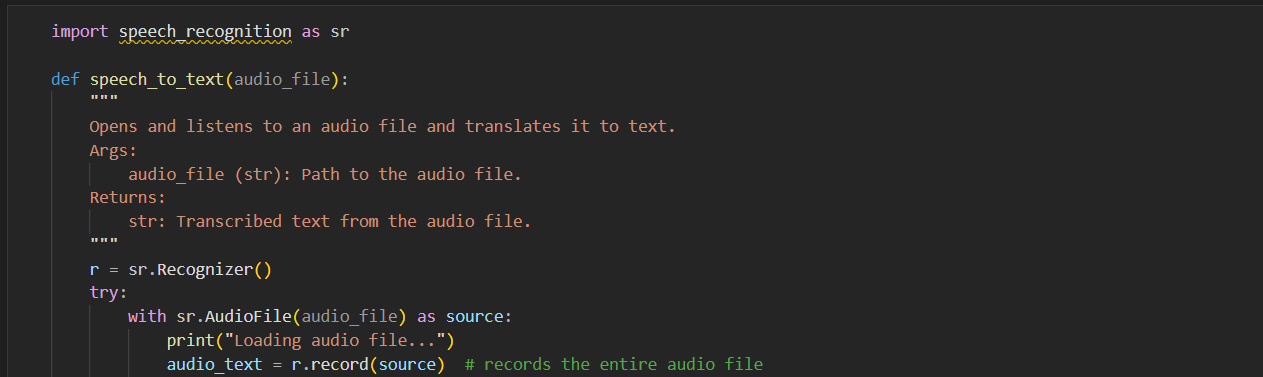
1. Loading and Initializing the Recognizer

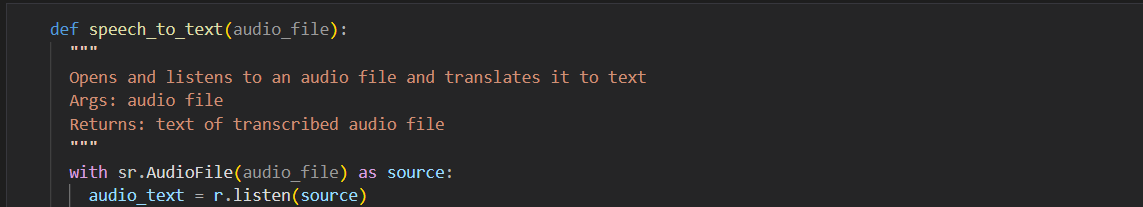




**Recognizer**: The primary object used to process audio input and recognize speech. It contains methods for audio recording and processing.

2. Handling Audio Input







* The **AudioFile** method loads the audio file and opens it in a read-only mode.
* The **record()** method captures the entire audio stream, converting it into a digital signal representation.
* The **listen()** function in the SpeechRecognition library is used to capture audio data from a live microphone input. It processes the audio signal and converts it into a format suitable for speech recognition.

#### 3. Feature Extraction

* The library extracts key features from the audio, such as:
  + **Mel Frequency Cepstral Coefficients (MFCCs)**: Encodes the short-term power spectrum of speech.
  + **Pitch and Energy**: Measures voice pitch and intensity.
  + **Temporal Patterns**: Captures speech dynamics over time.

These features are then converted into a format understandable by the recognition engine.

4. Speech Recognition



The **Google Web Speech API** is used to perform the actual speech-to-text conversion.

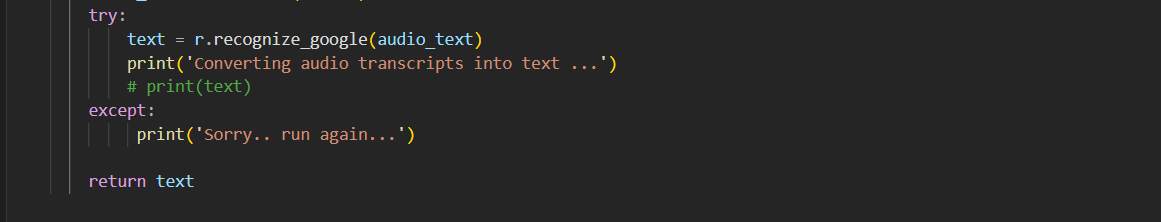
**Key Processes in Recognition:**

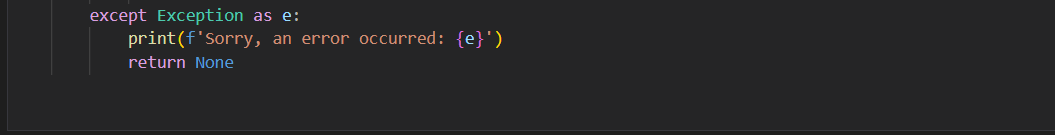
* **Acoustic Modeling**: Maps audio features (e.g., MFCCs) to phonemes (basic units of sound).
* **Language Modeling**: Combines recognized phonemes into valid words and sentences using a probabilistic model.

#### 5. Handling Errors

The library includes mechanisms to handle errors gracefully:

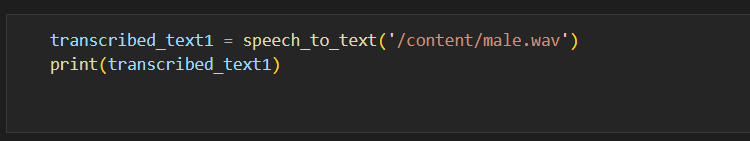
* **Unintelligible Speech**: If the speech is unclear, an exception is raised.
* **Noise or Silence**: Can be filtered using preprocessing methods.

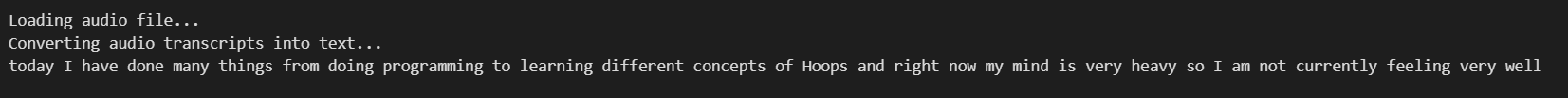




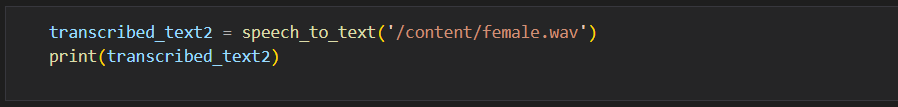
6. Output

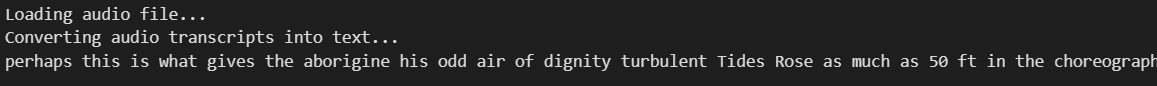
male.wav



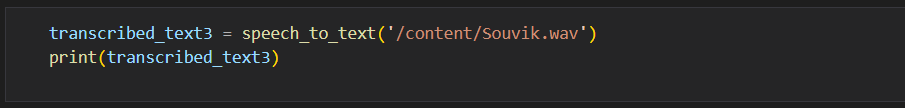


Female.wav

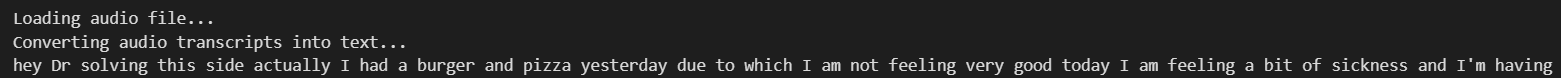




Souvik.wav









4.4 Multi modal analysis

**Gemini api**

**Using the Gemini API: A Quick Guide**

The Gemini API by Google AI provides advanced language models for tasks like text generation, translation, and creative content creation. Here’s how to get started:

1. **Obtain API Access:**
   * Create a Google Cloud Platform (GCP) account.
   * Enable the Vertex AI API and set up a project with billing.
   * Obtain an API key for authentication.
2. **Choose a Model:**
   * Gemini offers multiple models tailored for different tasks.
   * Select based on factors like model size, speed, cost, and task requirements.
3. **Prepare the Prompt:**
   * Craft a clear, concise, and specific prompt for the task.
   * Example: To generate a poem about a robot on Mars, use: *"Write a poem about a curious robot exploring Mars."*
4. **Send the Request:**
   * Use one of the following methods:
     + **HTTP Requests**: Directly interact with the API.
     + **Client Libraries**: Simplify integration with language-specific libraries.
     + **Google Cloud Console**: Test prompts in a user-friendly interface.
5. **Process the Response:**
   * The model generates a response based on the prompt.
   * Parse and integrate the output into the application or project.

### Key Tips for Effective Use

* **Prompt Engineering**: A well-crafted prompt leads to better results. Experiment for optimal phrasing.
* **Model Selection**: Match the model’s capabilities with the project’s needs.
* **Error Handling**: Ensure robust mechanisms for API errors or unexpected outputs.
* **Ethical Use**: Avoid generating harmful, biased, or misleading content.

By following these steps, we can harness the Gemini API’s capabilities to enhance our projects with AI-driven language tasks.

**Implementation of gemini api**

In our multimodal analysis project, we are combining data from three different sources — text, voice, and facial expressions — to analyze the emotional state of an individual, specifically to detect potential mental health concerns like anxiety or depression. Here’s a step-by-step explanation of how the process works, based on the provided code and implementation:

### 1. Multimodal Data Inputs

The system takes inputs from three different modalities (types of data):

* **Sentiment Analysis of Text**: This input comes from the sentiment analysis of WhatsApp texts. For example, "70% negative" means that the text analysis revealed a significant portion of the conversation reflects negative sentiment.
* **Voice Emotion Detection**: This comes from an emotion recognition system analyzing the tone of voice in speech. In this case, "Sad" indicates that the detected emotional state from the voice is sadness.
* **Facial Expression Analysis**: This input is derived from analyzing the individual's facial expressions, detecting emotions like "Anxious."

These three inputs are essential because they provide different perspectives on the person's emotional state, allowing for a more holistic analysis.

### 2. The Role of the Gemini API

Once the three inputs (text sentiment, voice emotion, and facial expression) are prepared, they are passed together as a **prompt** to the **Gemini API**. The Gemini API, which is a generative AI model by Google, is tasked with processing the multimodal data and synthesizing it into a comprehensive emotional analysis.

The **prompt** we create combines these inputs in a clear, structured format that the API can understand:

**Input:**

Sentiment Analysis of WhatsApp Texts: {a}

Voice Emotion Detection: {b}

Facial Expression Analysis: {c}

**Task:**

Analyze the individual’s emotional state based on the provided multimodal data, assess potential mental health concerns like depression or anxiety, and provide appropriate recommendations.

Here:

* {a} represents the sentiment analysis result.
* {b} represents the emotion detected in the voice.
* {c} represents the emotion derived from facial expressions.

### 3. API Response Generation

The **Gemini model** then processes the combined inputs (sentiment, voice, and facial data) and generates a detailed response, which includes:

* **Multimodal Analysis**: The model analyzes the three types of input and integrates them to provide an overall assessment of the individual’s emotional state. For example, the presence of "70% negative" sentiment in text, sadness in the voice, and anxiety in the facial expressions could indicate distress or mental health concerns.
* **Mental Health Concerns**: Based on the patterns in the data, the model identifies potential issues like depression, generalized anxiety disorder (GAD), or other conditions. The analysis highlights possible mental health risks such as persistent sadness, excessive worry, or anxiety.
* **Recommendations**: After assessing the emotional state, the model suggests actions, like self-care techniques, consulting a mental health professional, and engaging with a support system. This guidance aims to help the individual manage their emotional distress.



### 4. Key Steps in the Process

1. **Input Collection**: Gather multimodal data (text, voice, and facial expression) from different sources.
2. **Preprocessing**: Analyze the sentiment of the text, detect emotions in the voice, and interpret facial expressions.
3. **Multimodal Integration**: Combine the insights from all three sources into a structured prompt for the Gemini API.
4. **API Response**: The Gemini model processes the combined data and generates a comprehensive emotional analysis, including potential mental health concerns and suggestions for action.
5. **Output**: Present the analysis and recommendations to the user.

### 5. Benefits of Using Multimodal Analysis

* **Holistic View**: By combining data from multiple sources (text, voice, and facial expressions), the model can form a more complete and accurate picture of the individual's emotional state.
* **Early Detection**: The system can help detect early signs of emotional distress, such as anxiety or depression, by analyzing patterns in behavior across different modalities.
* **Personalized Recommendations**: The system can provide tailored suggestions to help improve emotional well-being, offering both self-care strategies and professional guidance.

In summary, the integration of text sentiment analysis, voice emotion detection, and facial expression analysis through the Gemini API enables an advanced approach to understanding an individual's emotional state. This multimodal approach enhances the accuracy and depth of emotional analysis, leading to better insights and more effective recommendations for mental health support.

**Output of Multimodal Analysis:**

Here is the code snippet where the output of the other models are given as input to gemini api:

genai.configure(api\_key="AIzaSyDDQunQPh9EqoIhizXvJD2jI3Dcf\_YZtgA")

model = genai.GenerativeModel("gemini-1.5-flash")

Sentiment = "70% negative"

Voice\_Emotion = "Sad"

Facial\_Emotion = "Anxious"

prompt = """Prompt:

Input:

Sentiment Analysis of WhatsApp Texts: {a}

Voice Emotion Detection: {b}

Facial Expression Analysis: {c}

Task:

Based on the provided multimodal data, analyze the individual's emotional state. Assess the potential for mental health concerns such as depression or anxiety. Provide a comprehensive analysis and suggest appropriate actions.""".format(a=Sentiment, b=Voice\_Emotion, c=Facial\_Emotion)

response = model.generate\_content(prompt)

print(response.text)



Chapter 5

Backend & Frontend

5.1 Procedure - Backend

Mentalytics : AI for Mental Health Monitoring

****



5.2 Frontend Overview

The frontend of the **Mentalytics: AI for Mental Health Monitoring** application is designed to provide an interactive and user-friendly interface for various mental health monitoring tasks. It is built using **HTML**, **CSS**, **Bootstrap**, and **JavaScript (with jQuery)**. This interface allows users to interact seamlessly with the application’s core functionalities, such as **Facial Emotion Analysis**, **WhatsApp Chat Sentiment Analysis**, **Speech Emotion Analysis**, and **Multimodal Analysis**.

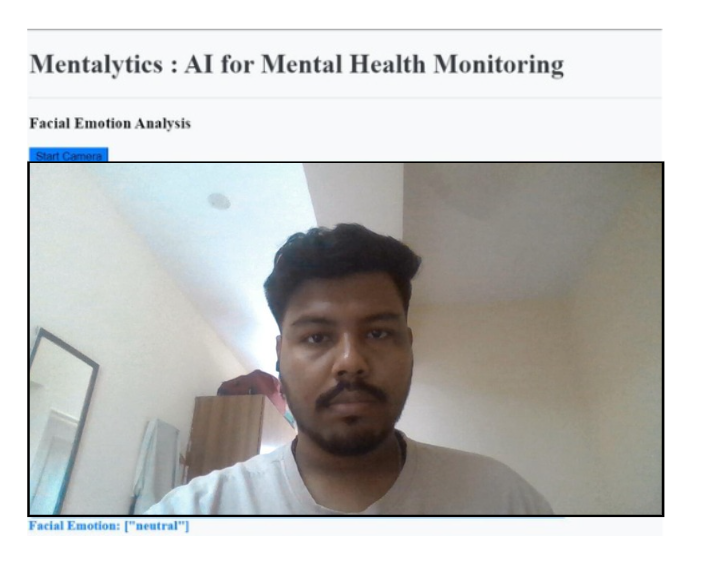
#### Features

1. **Responsive Design**:
   * Utilizes **Bootstrap 4.5.2** for a mobile-first, responsive design.
   * Ensures compatibility across devices with varying screen sizes.
2. **Modular Structure**:
   * Divides functionalities into clearly defined sections with intuitive navigation.
3. **Interactive Elements**:
   * Buttons to trigger actions like starting the camera, uploading files, and performing analyses.
   * Real-time feedback for results, displayed dynamically without page reloads.
4. **User-Friendly Styling**:
   * Styled using a combination of Bootstrap classes and custom CSS.
   * Clean and professional look with a focus on accessibility and ease of use.
5. **Dynamic Media Handling**:
   * Displays live video streams for facial emotion detection.
   * Supports file uploads for analyzing WhatsApp chats and speech emotions.
6. **Visualization**:
   * Displays sentiment analysis results graphically (e.g., bar plots) using embedded base64-encoded images.

#### Key Components

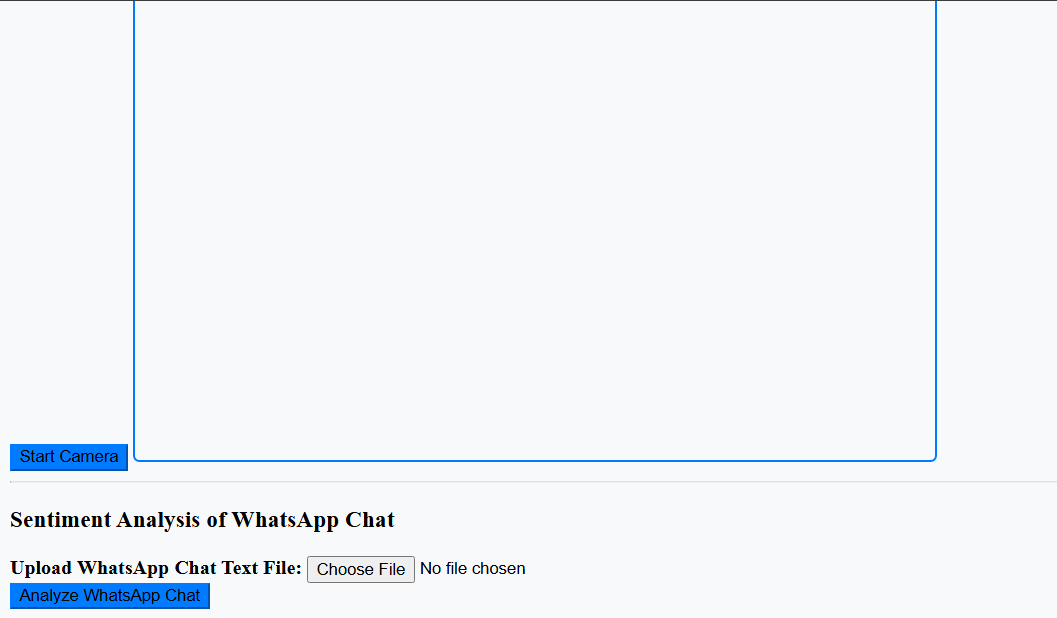
1. **Facial Emotion Analysis**:
   * Integrates live camera feed using navigator.mediaDevices.getUserMedia.
   * Displays real-time detected emotions below the video feed.

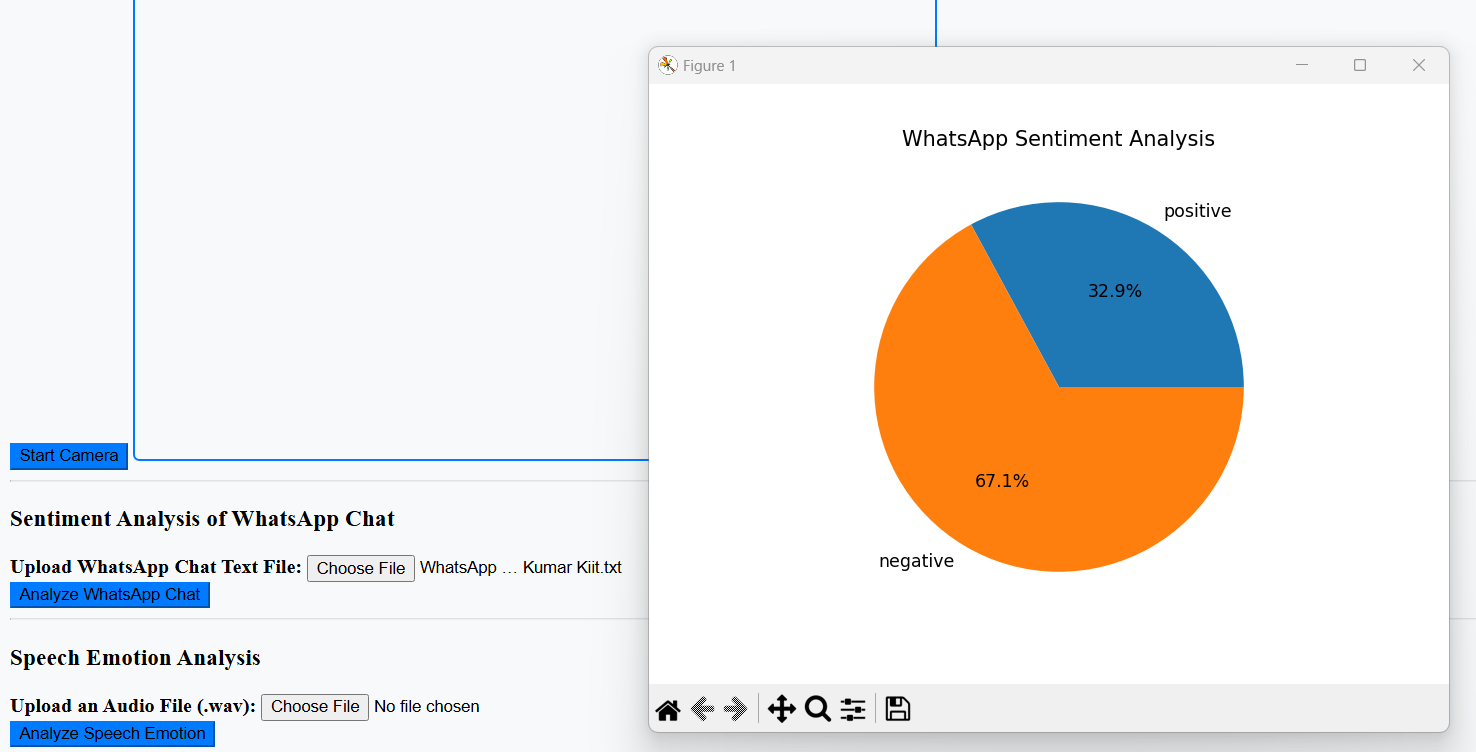




#### 

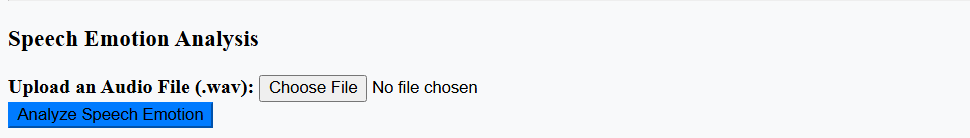
1. **WhatsApp Chat Sentiment Analysis**:
   * File upload functionality to analyze .txt files.
   * Displays both sentiment scores and a sentiment distribution plot.

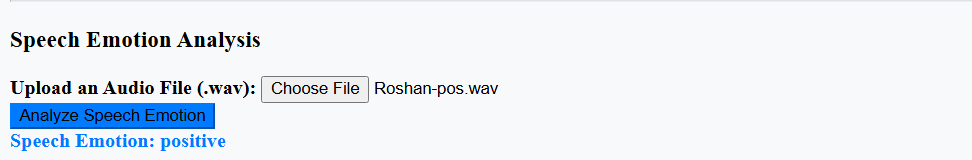






1. **Speech Emotion Analysis**:
   * Upload functionality for .wav files.
   * Outputs the detected emotional state of the speaker.





1. **Multimodal Analysis**:
   * Aggregates results from all analyses.
   * Displays a comprehensive emotional assessment derived from multimodal data.

#### User Flow

1. **Landing Page**:
   * A clean and welcoming interface with the title: "Mentalytics: AI for Mental Health Monitoring".
   * Sections are divided into **Facial Emotion Analysis**, **WhatsApp Chat Analysis**, **Speech Analysis**, and **Multimodal Analysis**.
2. **Input Actions**:
   * Users can start the camera, upload files, or trigger multimodal analysis by clicking buttons.
3. **Real-Time Feedback**:
   * Results are dynamically fetched and displayed using JavaScript and AJAX calls to the backend endpoints.



#### Technologies Used

* **HTML5**: For structuring the webpage.
* **CSS3**: For custom styles, ensuring a polished appearance.
* **Bootstrap 4**: For responsive layout and pre-defined components like buttons, forms, and grids.
* **JavaScript**: For handling user interactions, file uploads, and AJAX calls.
* **jQuery**: Simplifies DOM manipulation and AJAX requests.

#### Code Highlights

1. **Live Camera Integration**:

// Start Camera and Analyze Facial Emotion in Real Time

startCameraButton.addEventListener('click', () => {

navigator.mediaDevices.getUserMedia({ video: true })

.then(stream => {

video.srcObject = stream;

video.style.display = 'block';

analyzeFacialEmotion();

})

.catch(err => {

console.error("Error accessing camera: ", err);

});

});

1. **File Upload Handling**:

<!-- Sentiment Analysis of WhatsApp Chat Section -->

<div class="mt-5">

<h3>Sentiment Analysis of WhatsApp Chat</h3>

<form id="whatsapp-chat-form" enctype="multipart/form-data">

<div class="form-group">

<label for="chat-file">Upload WhatsApp Chat Text File:</label>

<input type="file" class="form-control-file" id="chat-file" name="file" accept=".txt" required>

</div>

<button type="button" class="btn btn-primary" onclick="analyzeWhatsAppChat()">Analyze WhatsApp Chat</button>

</form>

<div id="whatsapp-result" class="mt-3">

<img id="sentiment-plot" class="mt-3" style="display: none;" /> </div> </div>

1. **Dynamic Results Display**:

// WhatsApp Chat Analysis

function analyzeWhatsAppChat() {

const formData = new FormData(document.getElementById('whatsapp-chat-form'));

fetch('/analyze\_whatsapp\_chat', {

method: 'POST',

body: formData

})





.then(response => {

if (!response.ok) {

return response.json().then(errorMessage => {

throw new Error(errorMessage.error || 'Unknown error occurred');

});

}

return response.json();

})

.then(data => {

if (data.error) {

document.getElementById('whatsapp-result').innerText = 'Error: ' + data.error;

} else {

// Display sentiment counts

document.getElementById('whatsapp-result').innerText = `Positive: ${data.positive}, Negative: ${data.negative}`;

// Display the sentiment plot image

const img = document.getElementById('sentiment-plot');

if (img) {

img.src = 'data:image/png;base64,' + data.image;

img.style.display = 'block';

}

}

})

.catch(error => {

document.getElementById('whatsapp-result').innerText = 'Error: ' + error.message;

});

}

## 5.3 Result Obtained

The **frontend interface** of the *Mentalytics: AI for Mental Health Monitoring* system offers an intuitive, user-friendly platform with the following features:

1. **Facial Emotion Analysis**:
   * A live video feed is initiated via the webcam using the **Start Camera** button.
   * The system processes the video feed in real-time to identify emotions such as happiness, sadness, anger, fear, etc.
   * The detected emotions are displayed dynamically on the screen.
2. **Sentiment Analysis of WhatsApp Chats**:
   * Users upload a WhatsApp chat file in .txt format through a dedicated upload form.
   * Upon processing, the system:
     + Displays the **sentiment results**, including the count of positive and negative sentiments.
     + Generates a **sentiment distribution plot**, which is displayed directly in the interface
3. **Speech Emotion Analysis**:
   * Users upload an audio file (.wav format) via an input form.
   * The application identifies the **emotion expressed in the audio**, such as neutral, happy, or angry, and displays the result.
4. **Multimodal Emotional Analysis**:
   * This feature combines results from facial, speech, and WhatsApp sentiment analysis.
   * A comprehensive emotional analysis report is generated and displayed, providing insights into the individual's emotional state.

### Key Outputs:

* Emotion detection for live video feed.
* Sentiment distribution plots for textual data.
* Emotion results for speech audio.
* Comprehensive insights for multimodal data integration.

## Analysis and Observations

The analysis highlights the following capabilities and outcomes of the frontend system:

1. **Real-Time Processing**:
   * The webcam feed for facial emotion detection is processed every two seconds, ensuring a seamless real-time experience.
2. **Cross-Modal Integration**:
   * Results from multiple modalities (facial expressions, voice tone, and text sentiment) are successfully integrated to provide a holistic view of the individual's emotional state.
3. **Accuracy and Visualization**:
   * The sentiment analysis of WhatsApp chats not only provides numerical sentiment counts but also generates visual plots for better interpretability.
   * Speech emotion analysis results correlate effectively with other modalities for validation.
4. **User Experience**:
   * The **Bootstrap 4.5-based interface** ensures responsiveness and compatibility across devices.
   * Feedback from the system is immediate, enhancing usability.
5. **Areas of Improvement**:
   * Facial emotion detection could benefit from enhanced models for better accuracy under varying lighting conditions.
   * Expanding support for different audio formats could make the speech analysis module more versatile.

### Observations:

The frontend successfully serves as an interface for an AI-driven mental health monitoring system. The integration of real-time and asynchronous data processing, combined with visual results, ensures user engagement while maintaining system functionality.



CONCLUSION

In the context of this project, this report has presented a comprehensive exploration of an innovative AI-powered system designed to monitor and assess emotional states through multimodal data integration, encompassing facial expressions, voice patterns, and textual sentiment analysis. By employing advanced machine learning techniques and natural language processing, the system provides accurate and timely emotion detection, enhancing the potential for early intervention in mental health management.

The project highlights the importance of addressing the nuances of emotional expressions across different contexts and the ethical considerations necessary for handling sensitive data. Our findings demonstrate that integrating diverse data sources not only improves the accuracy of emotional assessments but also enables personalized feedback, fostering self-awareness and proactive mental health care.

As we stand on the precipice of technological advancements, the insights drawn from this research advocate for the adoption of AI solutions in mental health monitoring. The potential for early detection of conditions such as depression and anxiety underscores the need for innovative approaches in healthcare. We encourage further development and application of this technology, inviting stakeholders across academic, clinical, and organizational settings to collaborate in refining these systems for greater impact.

Ultimately, this multifaceted approach offers a pathway toward a deeper understanding of mental health, exemplifying how AI can empower individuals and communities to navigate emotional landscapes more effectively. Together, we can harness these tools to promote mental wellness, ensuring that technological progress translates into meaningful support for those in need.



***References***

1. *Jurafsky, D., & Martin, J. H. Speech and Language Processing. Pearson, 2021.*
2. *Goldberg, Y. Neural Network Methods for Natural Language Processing. Morgan & Claypool Publishers, 2017.*
3. *Manning, C. D., Raghavan, P., & Schütze, H. Introduction to Information Retrieval. Cambridge University Press, 2008.*
4. *Bishop, C. M. Pattern Recognition and Machine Learning. Springer, 2006.*
5. *Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.*
6. *LeCun, Y., Bengio, Y., & Hinton, G. "Deep Learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.*
7. *Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.*
8. *Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, pp. 4171-4186, 2019.*
9. *Chollet, F. Keras Documentation. Retrieved from* [*https://keras.io*](https://keras.io)*, 2015.*
10. *Kingma, D. P., & Ba, J. "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.*
11. *Google Cloud AI. "Gemini API Documentation." Retrieved from https://cloud.google.com/vertex-ai/docs.*
12. *Panda, P., et al. "Efficient Emotion Recognition Using CNN Architectures," International Journal of Advanced Computer Science and Applications, vol. 9, no. 8, pp. 150-157, 2018.*
13. *Liu, B. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012.*
14. *Pang, B., & Lee, L. "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1-135, 2008.*
15. *Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach, 4th Edition. Pearson, 2020.*



**Contribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Roll Number** | **Name** | **Contribution** |
| 1 | 2105511 | Aakash Kumar | Front-End Design, Facial Emotion Recognition |
| 2 | 2105550 | Krishanu Roy | Speech Analysis Model |
| 3 | 2105570 | Roshan Aryan | Sentimental Analysis |
| 4 | 2105583 | Souvik Basak | Facial Emotion Recognition, Gemini API |
| 5 | 2105585 | Suvankar Panigrahi | Sentimental Analysis |
| 6 | 2105594 | Vijay Vishal | Speech Analysis Model |

**Abstract:**

This research introduces an AI-powered system for real-time emotion detection using multimodal data such as facial expressions, voice patterns, and text. The system employs machine learning techniques to classify emotions like happiness, sadness, anger, and fear. Facial landmarks, MFCCs, and emotion lexicons are extracted as features, with CNNs processing images and RNNs modeling temporal audio and text dependencies. A multimodal fusion approach enhances accuracy and robustness, with models trained on diverse datasets. Offering personalized feedback, such as relaxation recommendations or mental health alerts, the system aims to improve well-being by providing timely interventions and fostering better mental health management.



**Aakash Kumar**

**2105511**

**Individual Contribution and Findings:** My primary role in this project was to develop the facial recognition model and design the frontend of the project, ensuring a seamless user interaction experience.

**Facial Recognition Model Development:** I worked on building a robust facial recognition model utilizing Convolutional Neural Networks (CNN). The process involved preprocessing image datasets, including resizing, normalization, and augmentation, to enhance model performance.

**Frontend Development:** I designed and developed the frontend of the project to provide a user-friendly interface. This included creating an interactive dashboard for users to input data and view the multimodal analysis results. I implemented responsive design principles to ensure compatibility across different devices and incorporated intuitive navigation features for improved usability.

**Technical Findings and Experience:** Working on the facial recognition model strengthened my understanding of deep learning techniques, particularly CNNs, and their application in real-world scenarios. The frontend development experience allowed me to enhance my skills in web development tools and frameworks, focusing on creating a cohesive and visually appealing interface.

**Individual Contribution to Project Report Preparation:** I contributed to the project report by documenting the methodologies behind the facial recognition model, including data preprocessing, model training, and evaluation techniques.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will demonstrate the functionality of the facial recognition model and the frontend interface. I will also be prepared to address questions about the model's development and user interaction aspects.



**Krishanu Roy**

**2105550**

**Individual Contribution and Findings:** My primary role in this project was to develop the Speech Analysis model and contribute to the methodologies section of the project report.

**Speech Analysis Model Development:** I worked on creating the Speech Analysis model, which involved processing and analyzing speech data to detect emotional cues and patterns. I implemented audio feature extraction techniques, such as Mel-frequency cepstral coefficients (MFCCs), to transform the raw speech data into a form suitable for machine learning. The model was then trained to classify emotional states based on speech features. I focused on improving the model’s accuracy by experimenting with different classifiers and tuning hyperparameters to optimize performance.

**Methodologies Section of the Report:** I focused on documenting the methodologies used in the project, particularly in the Speech Analysis component. This involved clearly outlining the steps taken for data preprocessing, feature extraction, and model selection. I explained the rationale behind the choice of algorithms, the training process, and the evaluation metrics used to assess model performance.

**Technical Findings and Experience:** Developing the Speech Analysis model allowed me to enhance my understanding of audio processing techniques and machine learning applications in speech recognition. Working on the methodologies section of the report improved my ability to communicate technical concepts in a structured and clear manner, ensuring that the details of the model’s development were well-documented.

**Individual Contribution to Project Report Preparation:** I contributed to the project report by writing the methodologies section, detailing the processes involved in the Speech Analysis model’s development.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will explain the workings of the Speech Analysis model, demonstrating how it processes speech data to detect emotional states. I will also be prepared to answer questions about the model’s implementation and the methodologies used in its development.

**Roshan Aryan**

**2105570**

**Individual Contribution and Findings:** My primary role in this project was to develop the WhatsApp chat sentiment analysis model and contribute to the methodologies section of the project report.

**WhatsApp Chat Sentiment Analysis Model Development:** I worked on creating a model to analyze sentiments in WhatsApp messages and classify them as positive, negative, or neutral. This involved applying Natural Language Processing (NLP) techniques, such as tokenization, stop-word removal, and vectorization, to preprocess the text data. Using the Naive Bayes classifier, I trained the model on labeled datasets and fine-tuned it to achieve a high level of accuracy.

**Technical Findings and Experience:** Building the sentiment analysis model allowed me to deepen my knowledge of NLP techniques and machine learning algorithms. I gained practical experience working with libraries such as NLTK and Scikit-learn, enhancing my ability to process and analyze textual data effectively.

**Individual Contribution to Project Report Preparation:** I contributed to the methodologies section of the project report by providing a detailed explanation of the WhatsApp chat sentiment analysis model. This included outlining the preprocessing techniques, model selection rationale, and evaluation metrics used, ensuring that the report accurately reflected the technical workflow and findings.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will discuss the methodology behind the sentiment analysis model and explain its key components, such as data preprocessing and classification techniques. I will also be prepared to address questions regarding the technical aspects and outcomes of the model.



**Souvik Basak**

**2105583**

**Individual Contribution and Findings:** My primary role in this project was to develop the facial recognition model using convolutional neural networks (CNNs) and to implement the Gemini API for multimodal emotional analysis.

**Facial Recognition Model Development:** I designed and implemented the core facial recognition model to identify emotions from facial expressions. This involved training a CNN on a diverse dataset to accurately classify emotions. I also focused on optimizing the model’s architecture and tuning hyperparameters to enhance its performance.

**Integration of Multimodal Analysis API:** I integrated the Gemini API to perform multimodal analysis by combining facial recognition results with text and voice inputs. This required ensuring seamless communication between the API and our system to provide real-time emotional insights.

**Technical Findings and Experience:** Developing the facial recognition model allowed me to enhance my understanding of convolutional neural networks and their application in emotion detection. The integration of the Gemini API provided experience in working with multimodal systems, deepening my knowledge of combining multiple data streams for a unified analysis.

**Individual Contribution to Project Report Preparation:** I contributed to the implementation section of the report where I explained how CNN works and its implementation in the Facial Recognition model. I also explained the architecture of Natural Language Processing and its implementation in Whatsapp Chat Sentimental Analysis.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will demonstrate the facial recognition model and its functionality. I will also explain how the multimodal analysis workflow operates and be prepared to address questions related to the model and API integration.

**Suvankar Panigrahi**

**2105585**

**Individual Contribution and Findings:** My primary role in this project was to develop the WhatsApp chat sentiment analysis model and create the presentation for the overall project.

**WhatsApp Chat Sentiment Analysis Model Development:** I worked on building a sentiment analysis model for classifying WhatsApp messages into positive, negative, or neutral sentiments. This involved preprocessing chat data, utilizing Natural Language Processing (NLP) techniques, and implementing a Naive Bayes classifier for sentiment classification.

**Technical Findings and Experience:** Developing the sentiment analysis model enhanced my understanding of text preprocessing and sentiment classification. I gained valuable experience working with NLP libraries, such as NLTK, and applying machine learning algorithms to real-world text data.

**Presentation Development:** I designed and created the PowerPoint presentation for the project, ensuring it effectively conveyed the objectives, methodologies, results, and contributions. I focused on visual clarity, concise content, and smooth transitions to ensure an engaging and informative presentation.

**Individual Contribution to Project Report Preparation:** I contributed details about the development of the sentiment analysis model, which were included in the methodology section of the project report. My input ensured that the report accurately captured the technical workflow and findings of this component.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will provide an overview of the entire project and explain the WhatsApp sentiment analysis model.

**Vijay Vishal**

**2105594**

**Individual Contribution and Findings:** My primary role in this project was to develop the speech analysis model for emotional recognition and contribute to the project report, focusing on the implementation details of the speech analysis model.

**Speech Analysis Model Development:** I worked on designing and training the speech analysis model, which involved preprocessing audio data, extracting key features such as pitch, tone, and intensity, and using machine learning techniques to classify emotions accurately. This required selecting appropriate algorithms and fine-tuning them to ensure robust performance across diverse audio samples.

**Technical Findings and Experience:** Developing the speech analysis model allowed me to enhance my skills in audio data preprocessing and feature extraction. I also gained practical experience in implementing machine learning for speech emotion recognition, refining my ability to design and optimize models for real-world applications.

**Individual Contribution to Project Report Preparation:** I significantly contributed to the implementation section of the project report, focusing on detailing the workflow, technical challenges, and solutions involved in developing the speech analysis model. This ensured the report comprehensively highlighted my contributions to this aspect of the project.

**Individual Contribution for Project Presentation and Demonstration:** During the project presentation, I will demonstrate the speech analysis model, showcasing how it processes audio inputs to detect emotions accurately. I will also be prepared to answer technical questions related to its implementation and performance.



