CHAPTER 1
INTRODUCTION

Depression is a highly common and serious mental health issue that affects a vast number of people globally. It often brings both emotional and physical challenges, typically showing up as ongoing feelings of sadness, emptiness, exhaustion, and a lack of interest in regular activities. One of the main hurdles in identifying depression is its frequent occurrence alongside other psychological or neurological conditions, including bipolar disorder, schizophrenia, and Parkinson's disease. This overlap can blur the diagnostic process, making it more complex— even for experienced healthcare providers who often depend on personal observation and subjective judgment.

As depression rates continue to climb—largely due to pressures from modern living, such as job stress, family issues, isolation, and social anxiety—there is an increasing demand for more accurate, consistent, and early-stage diagnostic systems. Conventional methods often fall short in catching early signs, focusing instead on visible symptoms after the condition has advanced. In response, machine learning techniques offer a forwardthinking solution, enabling systems that can analyze data-driven patterns to detect depression earlier and more reliably, paving the way for timely and customized mental health support.

1.1. OBJECTIVE:

This project reviews and analyzes various machine learning algorithms for diagnosing and predicting depression by comparing various data modalities in terms of effectiveness. The project aims to find the most reliable techniques to improve the diagnostic accuracy and consistency of depression. Furthermore, it will discuss the influence of modern lifestyle factors such as stress and social anxiety on depression prevalence. The study also surveys some data preprocessing and feature selection methods to enhance the performance of machine learning in depression prediction.

1.2. MOTIVATION:

Depression continues to rise as a major mental health issue, often remaining undetected due to its complex nature and the dependence on subjective clinical observations. Many individuals do not receive timely help, especially in early stages, leading to worsening symptoms and longterm impact. Contributing factors such as lifestyle stress, emotional strain, and lack of social connection have intensified the need for improved diagnostic methods. This project is motivated by the potential of machine learning to transform mental health care through objective, data-based analysis. By identifying early signs of depression using diverse input sources, it aims to support faster, more reliable detection and intervention.

1.3. BACKGROUND:

A prevalent and dangerous mental illness, depression affects a person's emotional state, dayto-day functioning, and everyday activities. Clinical judgments and self-reported experiences are the mainstays of traditional diagnostic methods, which might lead to inaccurate or delayed evaluations. When depressive symptoms are accompanied by neurological or psychological disorders, the situation becomes even more difficult. Machine learning is a cutting-edge and effective method to overcome these constraints by examining a variety of data sources, including speech patterns, facial expressions, and behavioral indications, in order to identify early warning symptoms of depression. These sophisticated models facilitate prompt, individualized mental health interventions, reduce human bias, and increase reliability.

1.4 PROBLEM STATEMENT:

There is inconsistency and a significant likelihood of misdiagnosis because the majority of traditional methods for diagnosing depression are very subjective and strongly dependent on a clinician's skill. In addition, this is heartless given that depressive symptoms might coexist with those of other mental and neurological conditions. The intricacy of depression may not be fully considered by current diagnosis processes, which could lead to treatment delays and, more significantly, potentially fatal consequences like suicide. More reliable and precise diagnostic technologies are desperately needed so that they can help with early diagnosis and treatment. Using machine learning techniques to process vast volumes of data can increase the accuracy of diagnosing depression. Nevertheless, no agreement has been made regarding the best model and methods in this regard.

1.5 SCOPE OF THE PROJECT:

The objective of this research is to leverage machine learning for the early identification and management of depression. It involves collecting and analyzing various data types—including text inputs, vocal features, and behavioral indicators—to develop predictive models capable of detecting depressive tendencies with high accuracy. The study evaluates and contrasts multiple machine learning algorithms to determine their effectiveness in both diagnosing depression and recommending mitigation strategies. Another key aspect of the project is the generation of personalized interventions to support timely mental health support. By integrating intelligent analysis with practical solutions, this research aims to build a robust, data-centric system that not only enhances mental health care but also contributes to the ongoing advancement of research in this field.

1.6 LITERATURE SURVEY :

S.NO	TITLE	MERITS	DEMERITS
1	**Evaluating Machine Learning Models for Depression Detection in Social Media**	- Assesses how well machine learning can identify depressive patterns in social media content. - Highlights social media as a potential tool for non-intrusive mental health assessment.	- Findings depend on social media datasets, which may not represent all populations. - Raises ethical concerns regarding data privacy and user consent.
2	**A Comparative Study of Machine Learning Approaches for Depression Prediction**	- Examines various AI techniques used for predicting depression. - Identifies gaps in research and suggests future areas of exploration.	- Lacks new experimental validation beyond prior studies. - Does not compare model performance in real-world applications.
3	**AI-Based Depression Screening for University Students Using Non-Medical Data**	- Introduces an AI model for early depression detection in college students. - Uses lifestyle and behavioral data instead of medical records, making it easy	- Reliance on non-medical data may reduce prediction accuracy. - The model's effectiveness for non-student populations is uncertain.
4	**Advancing Early Depression Detection with AI Algorithms**	- Focuses on early identification of depression to enable timely intervention. - Uses advanced AI techniques to enhance predictive accuracy.	- Findings may not generalize across different demographics. - May oversimplify mental health conditions by not considering individualized factors.
5	**Detecting Anxiety and Depression Through AI**	- Demonstrates AI's role in recognizing symptoms of anxiety and depression. - Can be extended to detect other psychological disorders.	- Distinguishing between anxiety and depression is challenging, leading to potential misdiagnoses. - AI models may lack personalization in mental health assessments.
6	**Combining Machine Learning and Brain Imaging for Depression Research**	- Uses brain imaging with AI to study depression from a neurological perspective. - Provides insights into how depression affects brain function.	- Imaging technology is expensive and not widely available. - Findings may not be practical for routine mental health diagnoses.
7	**Review of Machine Learning Applications in Mental Health Studies**	- Summarizes the role of AI in depression detection, from basic models to advanced	- Lacks real-world implementation details, making practical adoption difficult.

S.NO	TITLE	MERITS	DEMERITS
		techniques. - Identifies challenges and opportunities for improving AI-based mental health solutions.	- May not include recent advancements in the field.
8	**Wireless EEG-Based AI Model for Depression Detection in Young Adults**	- Uses a non-invasive wireless EEG device for depression screening. - Targets young adults, promoting early intervention strategies.	- Dependence on EEG technology may make scalability difficult. - Variability in EEG signals may introduce inconsistencies in predictions.
9	**Neural Networks for Lane Detection in Autonomous Vehicles**	- Uses deep learning for accurate lane detection, aiding autonomous driving systems. - Enhances road safety and self-driving car navigation.	- Not directly relevant to mental health research. - Focuses on transportation rather than healthcare applications.
10	**AI-Based Detection of Depression and Anxiety Using Supervised Learning**	- Uses supervised learning for diagnosing depression and anxiety. - Designed for large-scale use, improving general applicability.	- Risk of overfitting when trained on small or imbalanced datasets. - Requires high-quality labeled data, which may not always be available.
11	**Performance Evaluation of KNN vs. SVM in Depression Prediction**	- Compares the effectiveness of KNN and SVM algorithms for detecting depression. - Aims to improve accuracy in AI-driven mental health diagnostics.	- Small datasets or complex feature spaces may reduce model effectiveness. - May not capture the full complexity of depression as a mental illness.
12	**AI-Driven Multi-Algorithm Approach for Predicting Mental Health Conditions**	- Uses multiple machine learning models to improve accuracy. - Enhances early intervention strategies for mental health conditions.	- Requires clinical validation before deployment in healthcare settings. - Does not personalize predictions to individual patients' needs.
13	**Role of AI Chatbots in Digital Mental Health Support**	- Offers a cost-effective alternative to traditional therapy. - Can support multiple users simultaneously, improving accessibility.	- Lacks human empathy, which may reduce effectiveness in crisis situations. - Predefined responses may not cater to individual patient needs.

S.NO	TITLE	MERITS	DEMERITS
14	**Social Media's Influence on Mental Health and Depression**	- Raises awareness about depression through online communities. - Analyzes behavioral patterns to detect early signs of mental health issues.	- Risk of misinformation leading to incorrect self-diagnosis. - Excessive social media usage can contribute to mental distress.
15	**The Role of Mindfulness in Depression Management**	- A natural, non-invasive approach to improving mental well-being. - Can be practiced without the need for expensive treatments.	- Requires consistency and self-discipline for effectiveness. - Results vary, and benefits may not be immediate.

1.5. GAP IDENTIFICATION:

Despite growing interest in using machine learning for mental health applications, current research largely concentrates on detecting depression without offering solutions for individual recovery or support. Many existing models rely on single data sources, missing out on the potential benefits of combining various forms of input like voice patterns, user behavior, and social media activity. Furthermore, there's limited exploration of how these predictions can be turned into meaningful, personalized interventions. Most systems stop at diagnosis, lacking real-world tools that connect detection with actionable mental health assistance. This project addresses these shortcomings by creating a multi-dimensional approach that not only identifies depressive symptoms early but also delivers tailored strategies for timely support.

CHAPTER 2

PROJECT DESCRIPTION AND GOALS

2.1. PROJECT DESCRIPTION:

This project aims to develop an intelligent machine learning-based system for the early identification and management of depression. It utilizes multimodal data—including text, speech, and behavioral patterns—to train predictive models capable of recognizing emotional distress with precision. By applying advanced machine learning techniques such as Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (kNN), the system is designed to detect subtle indicators of depression and emotional imbalance.

The primary goal is to create a proactive support framework that can analyze real-time inputs and issue alerts or recommendations when signs of depressive behavior are detected. This ensures individuals can receive timely assistance, enabling early intervention before symptoms escalate. The system can be adapted for integration into various platforms such as mobile health applications, workplace wellness programs, or online counseling services.

Moreover, the project addresses challenges such as inconsistencies in individual expression, the need for real-time analysis, and model optimization for better accuracy and responsiveness. By combining innovative algorithms with a focus on mental health, the system offers a scalable and practical solution to support emotional well-being and reduce the stigma around mental health issues. This work contributes to bridging the gap between mental health needs and accessible, tech-driven support solutions.

2.2. PROJECT GOALS:

This project aims to develop an AI-powered system that can detect early indicators of mental health concerns through advanced machine learning techniques. By analyzing diverse data sources such as speech patterns, textual inputs, and behavioral traits, the system will effectively recognize distress signals.

To enhance accuracy and efficiency, various machine learning models—including Support Vector Machines (SVM), Random Forest, and k-Nearest Neighbors (k-NN)—will be implemented and optimized. The system will be designed to generate real-time alerts, providing timely support and interventions for individuals experiencing emotional distress.

Moreover, the project focuses on refining real-time processing capabilities, ensuring adaptability across different user scenarios. Ultimately, the goal is to create an intelligent, datadriven solution that seamlessly integrates into mental health frameworks, enabling proactive intervention and fostering overall psychological well-being.

2.3.Problem Statement:

Despite the increasing prevalence of mental health issues like depression, early detection and intervention remain challenging, particularly in educational institutions and general populations where symptoms often go unnoticed or are misunderstood. Existing depression detection systems using machine learning are limited by factors such as language dependency, lack of real-time interaction, insufficient accuracy due to imbalanced or biased data, and minimal integration with user-friendly platforms. There is a critical need for a reliable, accessible, and real-time solution that can detect early signs of depression using text-based inputs and deliver meaningful feedback to users while also ensuring privacy, simplicity, and adaptability across various user demographics.

2.3.1Existing System:

In recent years, several AI-based systems have been developed to detect and help mitigate depression using machine learning techniques. One notable system is Woebot, a conversational chatbot that leverages natural language processing (NLP) and deep learning to analyze users' messages for signs of depression, offering real-time, cognitive behavioral therapy (CBT)-based responses to support emotional well-being. Another system, Tess, developed by X2AI, functions as an AI mental health coach that interacts with users via text, using sentiment analysis to detect emotional distress and suggest personalized coping strategies. Research-driven projects like Stanford’s DAWN go a step further by incorporating multi-modal machine learning models that analyze voice tone, facial expressions, and written text to assess mental health conditions with high accuracy. These systems not only detect depressive symptoms but also play a role in mitigation by offering therapeutic guidance, emotional check-ins, or recommending professional help when needed. By combining behavioral science with artificial intelligence, these tools represent a major shift in accessible mental health care, particularly for early intervention and continuous monitoring.

Machine learning-based depression detection systems often rely on analyzing large datasets containing text, speech, or behavioral patterns from users. For instance, platforms like Ellie, developed by the University of Southern California’s Institute for Creative Technologies, use computer vision and audio processing to assess facial expressions, voice modulation, and speech patterns to detect signs of depression and anxiety. On the social media front, studies have used deep learning models to analyze posts on platforms like Twitter and Reddit to identify depressive language, sentiment shifts, and activity patterns that may indicate mental health issues. These systems typically use models such as Logistic Regression, Support Vector Machines (SVM), or more advanced techniques like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for classification. For mitigation, many systems incorporate interactive feedback, selfassessment tools, or integrate with therapists and caregivers to provide timely support. The use of real-time monitoring combined with AI-driven insights is enabling scalable, proactive approaches to mental health care and reducing the stigma of seeking help.

2.3.2 Disadvantages of Existing System:

Despite the promising capabilities of machine learning in detecting depression, existing systems face several limitations. One major disadvantage is the lack of personalization—many models are trained on generic datasets that may not accurately reflect individual differences in language use, cultural background, or emotional expression. Additionally, privacy concerns arise when sensitive personal data such as social media posts, voice recordings, or facial expressions are collected and analyzed, potentially leading to misuse or unauthorized access. Another challenge is the risk of false positives or negatives, where a person might be incorrectly classified, leading to unnecessary distress or missed interventions. These systems also often lack contextual understanding, meaning they may misinterpret sarcasm, humor, or idiomatic expressions. Lastly, many models require continuous internet connectivity and high computational power, making them less accessible in rural or low-resource settings. Addressing these challenges is critical to making AI-based mental health tools more reliable, ethical, and inclusive.

Another notable drawback of existing depression detection systems using machine learning is their limited integration with clinical workflows. These systems are often developed in research settings and may not align well with real-world healthcare practices, making it difficult for mental health professionals to adopt and trust them. Moreover, data imbalance in training datasets—where there may be more examples of non-depressed individuals than depressed ones—can lead to biased predictions, reducing the system's effectiveness in early detection. Many models also lack explainability, meaning they provide predictions without clear reasoning, which can make users and clinicians skeptical of the results. Additionally, language dependency can be an issue, as many systems are built primarily for English, excluding users who communicate in other languages or dialects. Without addressing these challenges, current systems may struggle to provide accurate, fair, and actionable insights across diverse populations.

CHAPTER 3

TECHNICAL SPECIFICATION

3.1. SOFTWARE REQUIREMENTS:

Development Environment

Google Colab provides a robust cloud-driven Jupyter notebook environment, perfect for developing machine learning and deep learning models. It offers a range of hardware configurations, including CPUs, GPUs, and TPUs. With a 12-hour continuous runtime limit and direct integration with Google Drive, it boosts efficiency and collaboration for AI initiatives.

Mobile Development Platform

Android Studio is the primary IDE for developing high-performance Android apps. It includes smart code suggestions, a drag-and-drop interface designer, adaptable emulators, and sophisticated debugging and performance optimization tools. Supporting both Java and Kotlin, it also allows Firebase integration for efficient backend development.

Programming Languages

Python is the key language for developing AI and deep learning models, whereas Java and Kotlin are utilized for Android app creation, ensuring user-friendly and responsive mobile applications.

Machine Learning Frameworks

TensorFlow and PyTorch play a vital role in constructing and enhancing predictive models, especially in the context of mental health applications, providing scalable and effective AI solutions.

Computer Vision Solutions

OpenCV and MediaPipe facilitate real-time assessments of visual and behavioral signals, aiding in the identification of non-verbal signs that may correlate with mental health issues like depression.

Speech and Text Analysis

The Google Speech-to-Text API and the SpeechRecognition library transform spoken words into text, enhancing AI-driven mental health evaluation tools.

Backend Infrastructure

Frameworks such as Django and Flask, built on Python, are utilized to develop robust APIs and oversee server-side tasks, ensuring efficient data management and processing.

3.2. HARDWARE REQUIREMENTS:

SystemRequirements:
A desktop or laptop equipped with a minimum 2.5 GHz multi-core processor and at least 8 GB of RAM is essential for running training scripts and handling real-time processing without lag.

MobileCompatibility:
The system is designed to support Android smartphones and tablets with no less than 4 GB RAM, allowing for efficient mobile application execution and user interaction.

DiskSpace:
To manage datasets, saved models, and temporary files, at least 20 GB of free storage is necessary for both mobile and system-side operations.

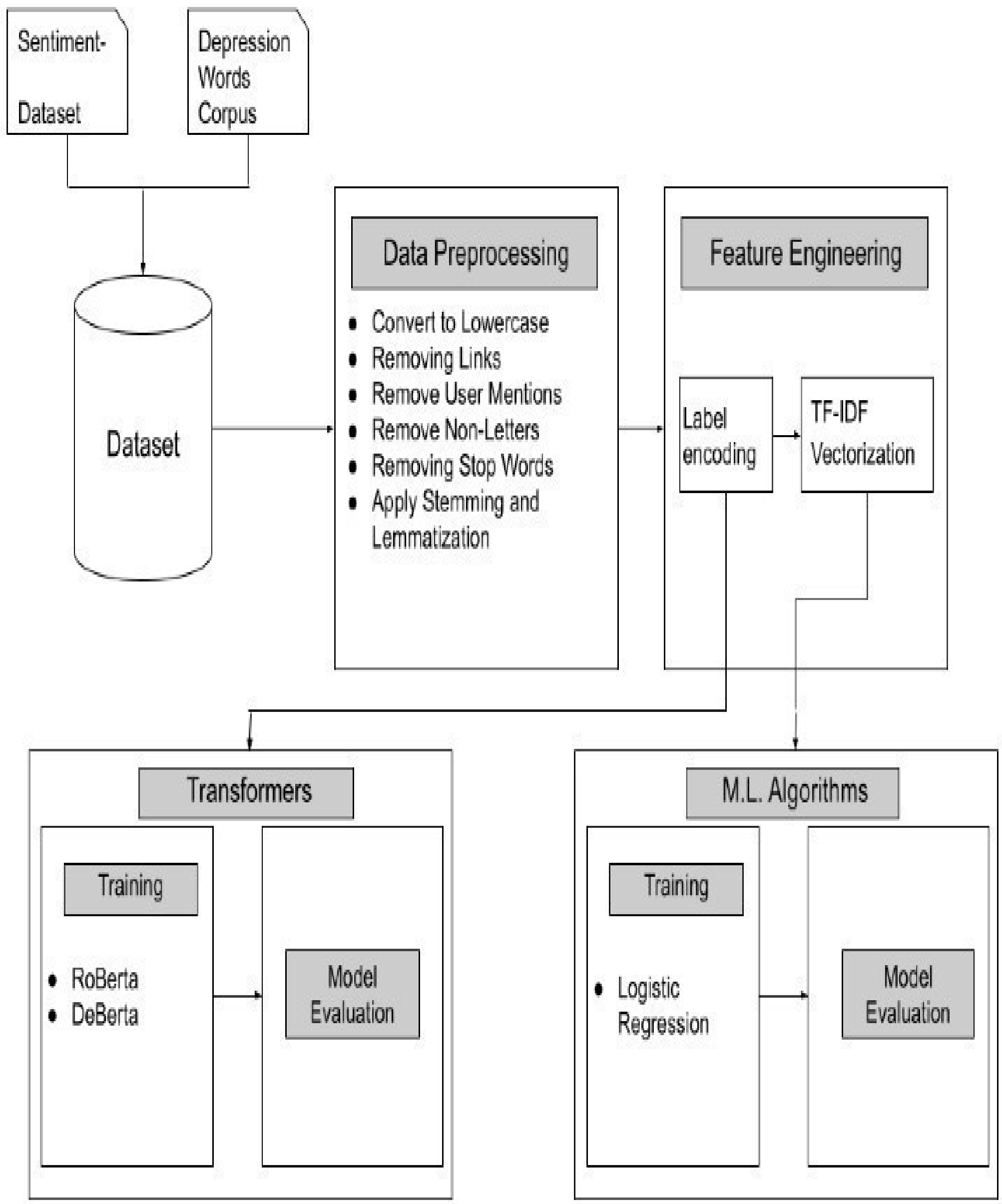
GPUAcceleration:
For faster computation during model training and image processing tasks, having a system with a dedicated graphics card like NVIDIA GTX 1650 or newer is highly beneficial.

NetworkRequirements:
A consistent internet connection, preferably broadband or 4G with a minimum speed of 5 Mbps, is important for real-time data syncing, cloud storage access, and remote collaboration.

CHAPTER 4

DESIGN APPROACH AND DETAILS

4.1 SYSTEM ARCHITECTURE:



4.2. PROPOSED METHODOLOGY

Proposed Methodology

This project introduces a machine learning-based system designed to identify signs of depression using only written text. The approach involves several key stages, from data collection to model deployment, all optimized to detect early symptoms and suggest appropriate interventions

.

1. System Architecture

The system operates through a structured pipeline that transforms user-written text into meaningful insights about their mental state. The architecture is organized as follows:

- Text Input Collection: Users provide textual content through an application interface. This could include journal entries, feedback, or responses to open-ended prompts.
- Text Preprocessing: The text is cleaned and standardized by removing stopwords, punctuation, and converting all characters to lowercase. Tokenization and stemming may also be applied to simplify the text.
- Feature Extraction: Important features such as emotional tone, word frequency, syntactic patterns, and sentiment scores are extracted using NLP techniques.
- Classification: Extracted features are passed to a trained machine learning model, which classifies the text into categories such as "No Depression", "Mild", "Moderate", or "Severe Depression".
- Mitigation Support: Based on the classification, the system provides tailored suggestions such as mood trackers, motivational content, or professional resources.

2. Machine Learning Model Design

To ensure accurate predictions, the system uses supervised learning methods. The steps include:

- Dataset Preparation: A curated dataset of text samples labeled with depression levels is used to train the model. Publicly available datasets may be augmented with synthetic data.
- Training Process: Algorithms such as Support Vector Machine (SVM), Random Forest, and Logistic Regression are evaluated. The model is trained to identify linguistic markers commonly associated with depression.
- Validation: Cross-validation and grid search are applied to fine-tune hyperparameters and prevent overfitting.

3. Technology Stack

The system is built using open-source tools and platforms:

- Python: Core programming language for development and integration.
- Natural Language Toolkit (NLTK) and spaCy: Used for text preprocessing and linguistic analysis.
- Scikit-learn: Employed for machine learning algorithm implementation and model evaluation.
- Pandas and NumPy: Utilized for handling data operations and feature engineering.
- Flask or Django: Backend frameworks to create the API and manage communication between the frontend and model.
- Google Colab: Cloud-based environment for training and testing the model.

4. Evaluation Metrics

To assess model performance:

- Accuracy, Precision, Recall, and F1-Score: These metrics will determine how well the model classifies various depression levels.
- Confusion Matrix: Used to visualize the classification outcomes.
- Latency Check: Ensures the system responds quickly, offering real-time results to the user.

5. Outcome and Goals

The ultimate goal is to create an AI-assisted text-based system that can act as an early indicator of mental health concerns. By analyzing written input, the system aims to provide nonintrusive, accessible mental health screening and support recommendations.

4.3 MODULES DESCRIPTION

Data Collection

The dataset used in this project consists of 160,000 text-based tweets sourced from Twitter. The data was collected using tools like Tweepy, which connects to the Twitter API to fetch relevant posts in real-time or from historical archives. The tweets were selected based on specific keywords, phrases, and hashtags commonly linked to emotional distress, sadness, or signs of depression

0	1467810369	Mon Apr 06 22:19:45	NO_QUERY	_TheSpecialOne_	@switchfoot http://t
0	1467810672	Mon Apr 06 22:19:49	NO_QUERY	scotthamilton	is upset that he can't
0	1467810917	Mon Apr 06 22:19:53	NO_QUERY	mattycus	@Kenichan I dived ma
0	1467811184	Mon Apr 06 22:19:57	NO_QUERY	ElleCTF	my whole body feels i
0	1467811193	Mon Apr 06 22:19:57	NO_QUERY	Karoli	@nationwideclass no
0	1467811372	Mon Apr 06 22:20:00	NO_QUERY	joy_wolf	@Kwesidei not the wl
0	1467811592	Mon Apr 06 22:20:03	NO_QUERY	mybirch	Need a hug
0	1467811594	Mon Apr 06 22:20:03	NO_QUERY	coZZ	@LOLTrish hey long t
0	1467811795	Mon Apr 06 22:20:05	NO_QUERY	2Hood4Hollywood	@Tatiana_K nope the
0	1467812025	Mon Apr 06 22:20:09	NO_QUERY	mimismo	@twittera que me mu
0	1467812416	Mon Apr 06 22:20:16	NO_QUERY	erinx3leannexo	spring break in plain c
0	1467812579	Mon Apr 06 22:20:17	NO_QUERY	pardonlauren	I just re-pierced my ea
0	1467812723	Mon Apr 06 22:20:19	NO_QUERY	TLeC	@caregiving I couldn'
0	1467812771	Mon Apr 06 22:20:19	NO_QUERY	robrobbierobert	@octolinz16 It it cour
0	1467812784	Mon Apr 06 22:20:20	NO_QUERY	bayofwolves	@smarrison i would'v
0	1467812799	Mon Apr 06 22:20:20	NO_QUERY	HairByJess	@iamjazzyfizzle I wish
0	1467812964	Mon Apr 06 22:20:22	NO_QUERY	lovesongwriter	Hollis' death scene wi
0	1467813137	Mon Apr 06 22:20:25	NO_QUERY	armotley	about to file taxes
0	1467813579	Mon Apr 06 22:20:31	NO_QUERY	starkissed	@LettyA ahh ive alwa
0	1467813782	Mon Apr 06 22:20:34	NO_QUERY	gi_gi_bee	@FakerPattyPattz Oh
0	1467813985	Mon Apr 06 22:20:37	NO_QUERY	quanvu	@alydesigns i was ou
0	1467813992	Mon Apr 06 22:20:38	NO_QUERY	swinspeedx	one of my friend calle
0	1467814119	Mon Apr 06 22:20:40	NO_QUERY	cooliodoc	@angry_barista I bak
0	1467814180	Mon Apr 06 22:20:40	NO_QUERY	viJILLante	this week is not going
0	1467814192	Mon Apr 06 22:20:41	NO_QUERY	Ljelli3166	blagh class at 8 tomo
0	1467814438	Mon Apr 06 22:20:44	NO_QUERY	ChicagoCubbie	I hate when I have to

. Examples include terms like “I’m tired of life,” “feeling empty,” or hashtags such as #depression or #mentalhealth. To maintain focus, only English-language tweets were considered, and retweets or non-relevant content were filtered out. Each tweet was stored in a structured format, containing text along with metadata like timestamp and sentiment polarity if available. No images, videos, or voice data were collected—only raw textual content was used. This collection forms the foundation for training and testing machine learning models aimed at recognizing depression-related language patterns in user posts.

Preprocessing

Before training the machine learning models, the raw tweet data undergoes a detailed preprocessing phase to ensure its quality and consistency. Initially, all text is converted to lowercase to maintain uniformity. Unwanted elements such as URLs, mentions (@usernames), hashtags, emojis, and special characters are removed, as they do not contribute meaningful semantic value. Tweets containing only non-alphabetical content are discarded.

Next, tokenization is applied to split each tweet into individual words, followed by the removal of common stopwords like "the", "is", and "and", which do not provide contextual relevance. Lemmatization is then used to reduce words to their base or dictionary form, helping in grouping similar expressions (e.g., "feeling" becomes "feel").

To prepare the text for machine learning, each processed tweet is transformed into numerical vectors using techniques such as TF-IDF (Term Frequency–Inverse Document Frequency) or word embeddings like Word2Vec. This structured format enables the models to effectively interpret emotional patterns and classify depressive indicators in the text.

This preprocessing step plays a crucial role in eliminating noise and enhancing the accuracy of the subsequent model training phase.

Model Training

In this project, the primary objective is to classify text inputs (tweets) as indicators of depression or not. For this purpose, Logistic Regression is employed as the core machine learning algorithm due to its simplicity, speed, and effectiveness in binary classification tasks.

Logistic Regression:

Logistic Regression is a supervised learning technique widely used for classification problems. Unlike linear regression, which predicts continuous values, logistic regression estimates the probability that a given input belongs to a particular class. In this context, it outputs the likelihood that a tweet expresses signs of depression.

The algorithm works by applying a sigmoid function to the linear combination of input features, mapping the result to a probability between 0 and 1. If the result crosses a defined threshold (commonly 0.5), the input is classified as "depressed"; otherwise, it is labeled as "nondepressed."

Relevance to Project:

Logistic Regression is highly effective for text-based sentiment classification. It is capable of handling high-dimensional data, such as TF-IDF or word embeddings derived from large text corpora like tweets. Its interpretability and fast computation make it ideal for real-time mental health monitoring, especially when rapid classification decisions are needed.

Model Development Process:

1. Feature Extraction: After preprocessing, each tweet is transformed into a numerical format using methods such as TF-IDF vectorization, which captures the importance of words in context.
2. Training Phase: The logistic regression model is trained on labeled tweet data, learning patterns that differentiate between depressive and neutral expressions.
3. Evaluation: The model's accuracy is measured using standard metrics like precision, recall, F1-score, and confusion matrix, ensuring it can generalize well to new, unseen text.

4. Optimization: Hyperparameter tuning is performed using techniques such as Grid Search or Cross-Validation to maximize performance and avoid overfitting.
5. Prediction: Once trained and evaluated, the model is used to predict the depression status of incoming tweets. The system flags tweets with high probabilities of depressive content for further review or intervention.

Model Testing & Performance Evaluation

A critical step in assessing the trained system's ability to recognise depressing content in novel and unseen text inputs is model testing. Classifying text as either depressed (positive) or not depressed (negative) is the classification task for this project.

Method of Evaluation

A subset of the dataset is isolated for testing in order to gauge the model's accuracy in identifying text pertaining to depression. This makes it more likely that the model will be able to generalise to inputs from the real world rather than merely memorise the training data. The following

Precision
This is the total percentage of accurate forecasts. Out of all the predictions the model made, it shows the number of times it accurately identified tweets as either depression or nondepressive.

(Sensitivity) Recall

This indicates the percentage of real depressed tweets that the program accurately detects. The algorithm misses fewer actual cases of depression when the recall is higher.

F1 Rating

The F1 score provides a balanced metric by combining precision and recall into a single value. When there are unequal numbers of depressed and non-depressive entries in the dataset, it is quite.

Matrix of Confusion

The prediction results are displayed using a confusion matrix:
True Positive (TP): Tweets that were depressed were accurately classified as such.
True Negative (TN): Tweets that aren't depressed are appropriately identified as such.
False Positive (FP): When tweets that aren't depressive are mistakenly classified as such.

False Negative (FN): Tweets indicating depression that were mislabeled as nondepressed.

This matrix aids in determining the model's accuracy and potential areas for improvement.

Performance Perspectives

We can determine the model's capacity to identify subtle indications of sadness from textual input by examining the aforementioned metrics, particularly the confusion matrix and F1 score. To improve performance in particular areas, the training data or hyperparameters can be change.

Accuracy :

Accuracy is frequently used to assess model performance, but it can be unreliable in the context of imbalanced data. In these situations, a model might seem to perform well by mostly predicting the dominant class correctly, yet it may struggle to identify the minority class accurately. As a result, relying solely on accuracy can give a false impression of the model's true capability.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Out of all the occurrences the model predicted as positive, this indicates the proportion of accurately predicted positive instances. All of the dataset's positive predictions are included in the denominator. Consider it a measure of "how often the model is correct when it

$$\frac{TP}{TP + FP}$$

Recall/Sensitivity/True Positive Rate :

percentage of positive cases relative to all positive cases that actually occurred. As a result, the actual number of positive cases in the dataset is the denominator (TP + FN). This can be interpreted as determining "the amount of extra right ones that the model missed when it displayed the right ones."

$$\frac{TP}{TP + FN}$$

score :

The F1 score has the drawback of giving equal weight to recall and precision, which might not be appropriate for applications where one is more important than the other. In these situations, more pertinent information can be obtained by looking at the Precision-Recall (PR) or Receiver Operating Characteristic (ROC) curves, or by using the weighted F1 score.

$$\frac{2}{\frac{1}{precision}+\frac{1}{recall}}=\frac{2*precision*recall}{precision+recall}$$

Output Prediction

Through text input, this project produces a useful and portable depression detection system that can be used directly on Android mobile devices, facilitating quick and easy mental health screening.

Text data has been used to train a Logistic Regression model, which classifies the input as either positive (perhaps depressive sentiment) or negative (non-depressive sentiment) in order to find emotional patterns that suggest symptoms of depression.

To guarantee seamless operation on Android devices, the model is transformed into a TensorFlow Lite (TFLite) format, which lowers computational burden and speeds up processing.
The end result is an Android-compatible app that can be used on the device to anticipate sentiment in real time without the need for an internet connection.

Users can input their thoughts or feelings through text, and the app immediately evaluates the content, providing instant feedback that supports early awareness and emotional monitoring.

This mobile deployment makes the solution practical, scalable, and accessible—ideal for personal use, student wellness programs, or integration into larger mental health support systems.

CHAPTER 5
SCHEDULE, TASKS AND MILESTONES

5.1. PROJECT PLAN:

1. Literature Review :

✓ Reviewed 20 papers, identified gaps in existing systems, particularly the limitations of motion sensors, and explored technologies such as Random Forest, SVMs, and KNN and OpenCV for real-time object detection.

2. Data Collection & Annotation :

✓ Collect, preprocess, and label data, train ML models, test, validate, and deploy for realtime workplace safety.

3. Model Training :
   ✓ Trained the model on the annotated dataset for accurate detection.

4. Model Validation & Testing :
   ✓ Validated and tested the model using picture with performance and accuracy.

5. Final Testing & Debugging :
   ✓ Test the entire system, including real-time detection to ensure smooth operation.

6. Documentation & Presentation :
   ✓ Prepare the final project report and presentation, detailing testing results and conclusions.

5.2 Gantt Chart

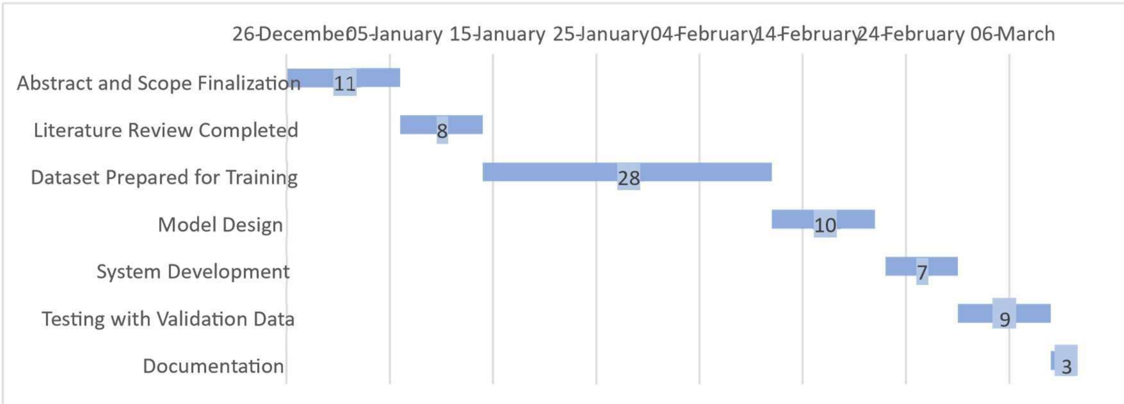


Figure 5.1 Gantt Chart

5.3Project Table

Description	Start Date	End Date	Duration
Abstract and Scope Finalization	26-December	05-January	11
Literature Review Completed	06-January	13-January	8
Dataset Prepared for Training	14-January	10-February	28
Model Design	11-February	20-February	10
System Development	22-February	28-February	7
Testing with Validation Data	01-March	09-March	9
Documentation	10-March	12-March	3

Table 5.2

CHAPTER 6

PROJECT DEMONSTRATION

6.1Model.py

```
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
df = pd.read_csv('/content/drive/MyDrive/training.1600000.processed.noemoticon.csv', encoding='latin-1')
# You might need to try other encodings like 'ISO-8859-1' if 'latin-1' doesn't work.
df.columns = ["sentiment", "id", "date", "flag", "user", "text"]
df['sentiment'] = df['sentiment'].map({0: 0, 4: 1})
```

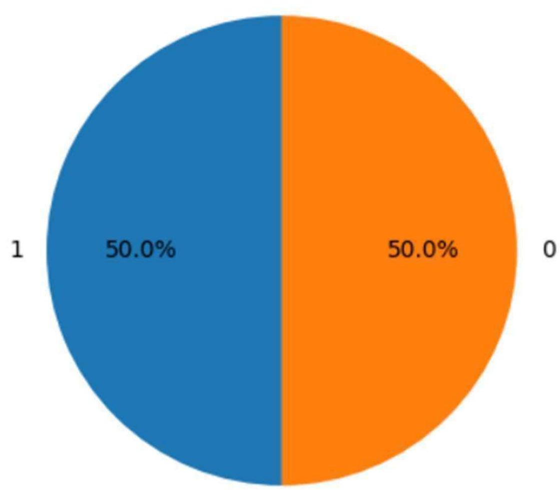
Data loaded

```
[ ] len(df)
```

```
1599999
```

```
import matplotlib.pyplot as plt
v_ct = df2['sentiment'].value_counts()
plt.figure(figsize=(4, 4))
plt.pie(v_ct, labels =v_ct.index, autopct = '%1.1f%%', startangle = 90)
plt.axis('equal')
plt.title('Label Distribution')
plt.show()
```

Label Distribution



```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
df = df2
# Preprocess and split the data
X = df['text'] # Features (text)
y = df['sentiment'] # Target (labels)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
# Create and train the logistic regression model
model = LogisticRegression(max_iter=1000) # Increased max_iter for convergence
model.fit(X_train_tfidf, y_train)
# Predict on the test set
y_pred = model.predict(X_test_tfidf)
# Evaluate the model
print("Accuracy:",accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Accuracy

```
Accuracy: 0.802565625
              precision    recall  f1-score   support

           0       0.81      0.79      0.80    159494
           1       0.80      0.81      0.80    160506

    accuracy                           0.80    320000
   macro avg       0.80      0.80      0.80    320000
weighted avg       0.80      0.80      0.80    320000
```

```python
def predict_sentiment_with_confidence(text):
     text_tfidf = vectorizer.transform([text])
     prediction = model.predict(text_tfidf)
     probabilities = model.predict_proba(text_tfidf)
     confidence = max(probabilities[0]) * 100
     sentiment = "Positive" if prediction[0] == 1 else "Negative"
     return [sentiment, confidence]
```

```python
input_text="I'm broken"
sentiment,confidence = predict_sentiment_with_confidence(input_text)
print(f"Input: '{input_text}' => Sentiment: {sentiment}, Confidence: {confidence:.2f}%")
```

```
Input: 'I'm broken' => Sentiment: Negative, Confidence: 99.40%
```

```python
import joblib
import numpy as np
import tensorflow as tf
from sklearn.feature_extraction.text import TfidfVectorizer

# Load the model and vectorizer
model = joblib.load("logistic_regression_model.pkl")
vectorizer = joblib.load("vectorizer4.pkl")

# Get vocabulary and IDF values from the vectorizer
vocabulary = vectorizer.vocabulary_
idf = vectorizer.idf_

# Create a TensorFlow model that wraps the Scikit-learn model
class SklearnModel(tf.Module):
    def __init__(self, model, vocabulary, idf):
        self.model = model
        self.vocabulary = vocabulary
        self.idf = idf

    @tf.function(input_signature=[tf.TensorSpec(shape=[None], dtype=tf.string)])
    def predict(self, text_input):
        def split_and_lookup(text):
            tokens = tf.strings.split(text)
            indices = tf.ragged.map_flat_values(
                lambda token: tf.lookup.StaticVocabularyTable(
                    tf.lookup.KeyValueTensorInitializer(
                        list(self.vocabulary.keys()),
                        list(self.vocabulary.values()),
                        key_dtype=tf.string,
                        value_dtype=tf.int64
                    ),
```

```python
import joblib
joblib.dump(model, "logistic_regression_model.pkl")
```

```
['logistic_regression_model.pkl']
```

```python
joblib.dump(vectorizer, "vectorizer4.pkl")
```

```
['vectorizer4.pkl']
```

```
                num_oov_buckets=1
            ).lookup(token),
            tokens
        )
        values = indices.values
        row_splits = indices.row_splits
        num_values = tf.cast(tf.shape(values)[0], tf.int64)
        value_indices = tf.range(num_values, dtype=tf.int64)
        gathered_values = tf.gather(values, value_indices)
        ragged_tensor = tf.RaggedTensor.from_row_splits(gathered_values, row_splits)
        return ragged_tensor

    token_indices = tf.map_fn(
        split_and_lookup,
        text_input,
        dtype=tf.RaggedTensorSpec(shape=[None, None], dtype=tf.int64, ragged_rank=1)
    )

    def calculate_tfidf_ragged(indices):
        dense_indices = indices.to_tensor(default_value=-1)
        counts = tf.reduce_sum(
            tf.one_hot(dense_indices, len(self.vocabulary) + 1), axis=1
        )[:, :-1]
        tfidf = counts * self.idf
        return tfidf

    text_tfidf = tf.map_fn(calculate_tfidf_ragged, token_indices, dtype=tf.float32)
    predictions = tf.numpy_function(
        func=lambda x: self.model.predict_proba(x.astype(np.float32)),
        inp=[text_tfidf],
        Tout=tf.float32
    )
    predictions.set_shape([None, model.classes_.shape[0]])
    return predictions
```

Importing Tensorflow

```
import tensorflow as tf

converter = tf.lite.TFLiteConverter.from_saved_model("logistic_regression_tf_model")
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()

with open("sentiment_analysis_model.tflite", "wb") as f:
    f.write(tflite_model)
```

6.2Flask.py

```python
from flask import Flask, request, jsonify, send_from_directory
import joblib
import matplotlib

matplotlib.use('Agg')
import matplotlib.pyplot as plt
import os
import uuid

app = Flask(__name__)
os.makedirs( name: "static", exist_ok=True)

model = joblib.load("logistic_regression_model.pkl")
vectorizer = joblib.load("vectorizer4.pkl")

mapping = {0: 'Negative', 1: 'Positive'}
descriptions = {
    "Positive": "Your responses indicate a positive state of mind. Keep maintaining a healthy lifestyle.",
    "Negative": "You may be experiencing symptoms of depression. Consider talking to a trusted friend or professional."
}


@app.route( rule: '/predict', methods=['POST'])  1 usage (1 dynamic)
def predict():
    try:
        data = request.get_json(force=True)
        text = data.get("text", "").strip()

        if not text:
            return jsonify({"error": "Text input cannot be empty"}), 400
    graph_filename = f"prediction_graph_{uuid.uuid4().hex}.png"
    graph_path = os.path.join("static", graph_filename)

    plt.savefig( *args: graph_path, bbox_inches='tight')
    plt.close()

    return graph_filename


@app.route('/static/<path:filename>')
def serve_static(filename):
    return send_from_directory( directory: 'static', filename)


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000, debug=True)
```

```python
        text_tfidf = vectorizer.transform([text])
        prediction = model.predict(text_tfidf)
        prediction_proba = model.predict_proba(text_tfidf)

        probability = float(f"{prediction_proba[0][prediction[0]]:.2f}")
        prediction_label = mapping[prediction[0]]
        description = descriptions[prediction_label]

        graph_filename = generate_graph(prediction_label, probability)

        return jsonify({
            "prediction": prediction_label,
            "prediction_probability": probability,
            "graph_url": f"https://94fb-2401-4900-4ac7-954f-5ce3-11dc-ff50-5c50.ngrok-free.app/static/{graph_filename}",
            "description": description
        })

    except Exception as e:
        return jsonify({"error": str(e)}), 500


def generate_graph(prediction_label, probability):  1 usage
    labels = [prediction_label, "Other"]
    sizes = [probability, 1 - probability]
    colors = ['green', 'gray'] if prediction_label == "Positive" else ['red', 'gray']

    plt.figure(figsize=(5, 5))
    plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=colors, startangle=140)
    plt.axis('equal')
```

6.3Prediciton.java

```
package com.example.myapplication;

import com.bumptech.glide.Glide;
import com.bumptech.glide.request.RequestOptions;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
```

```java
public class Prediction extends AppCompatActivity {
    2 usages
    private EditText inputText;
    2 usages
    private TextView predictionResult, descriptionText;
    2 usages
    private Button predictButton;
    2 usages
    private ImageView predictionGraph;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.prediction_layout);

        inputText = findViewById(R.id.inputText);
        predictionResult = findViewById(R.id.predictionResult);
        descriptionText = findViewById(R.id.descriptionText);
        predictionGraph = findViewById(R.id.predictionGraph);
        predictButton = findViewById(R.id.predictButton);
```

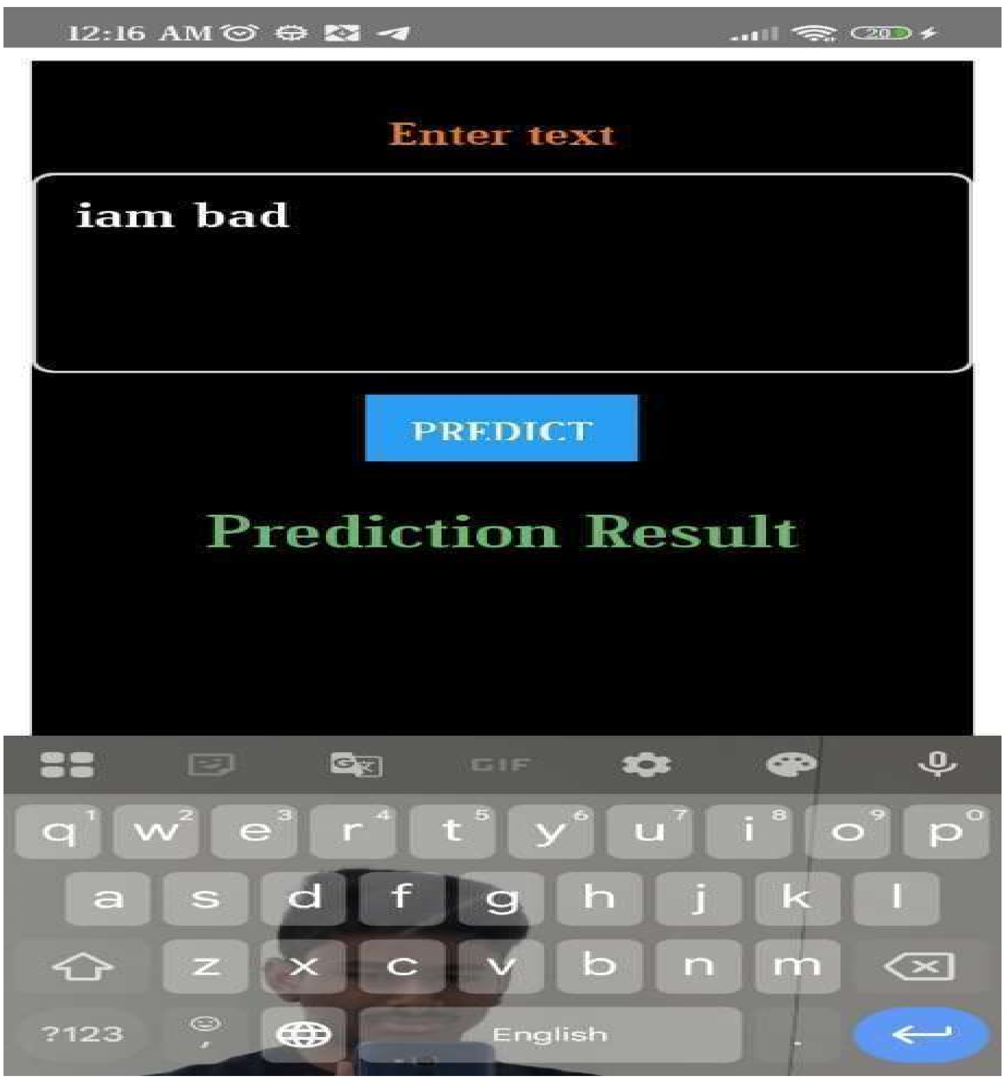
```java
        predictButton.setOnClickListener( View v -> {
            String text = inputText.getText().toString();
            if (!text.isEmpty()) {
                makePrediction(text);
            } else {
                Toast.makeText( context: Prediction.this,  text: "Please enter text", Toast.LENGTH_SHORT).show();
            }
        });
    }

    1 usage
    private void makePrediction(String inputText) {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        executor.execute(() -> {
            try {
                URL url = new URL( spec: "https://94fb-2401-4900-4ac7-954f-5ce3-11dc-ff50-5c50.ngrok-free.app/predict");
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setRequestMethod("POST");
                conn.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
                conn.setDoOutput(true);

                JSONObject jsonParam = new JSONObject();
                jsonParam.put( name: "text", inputText);
```

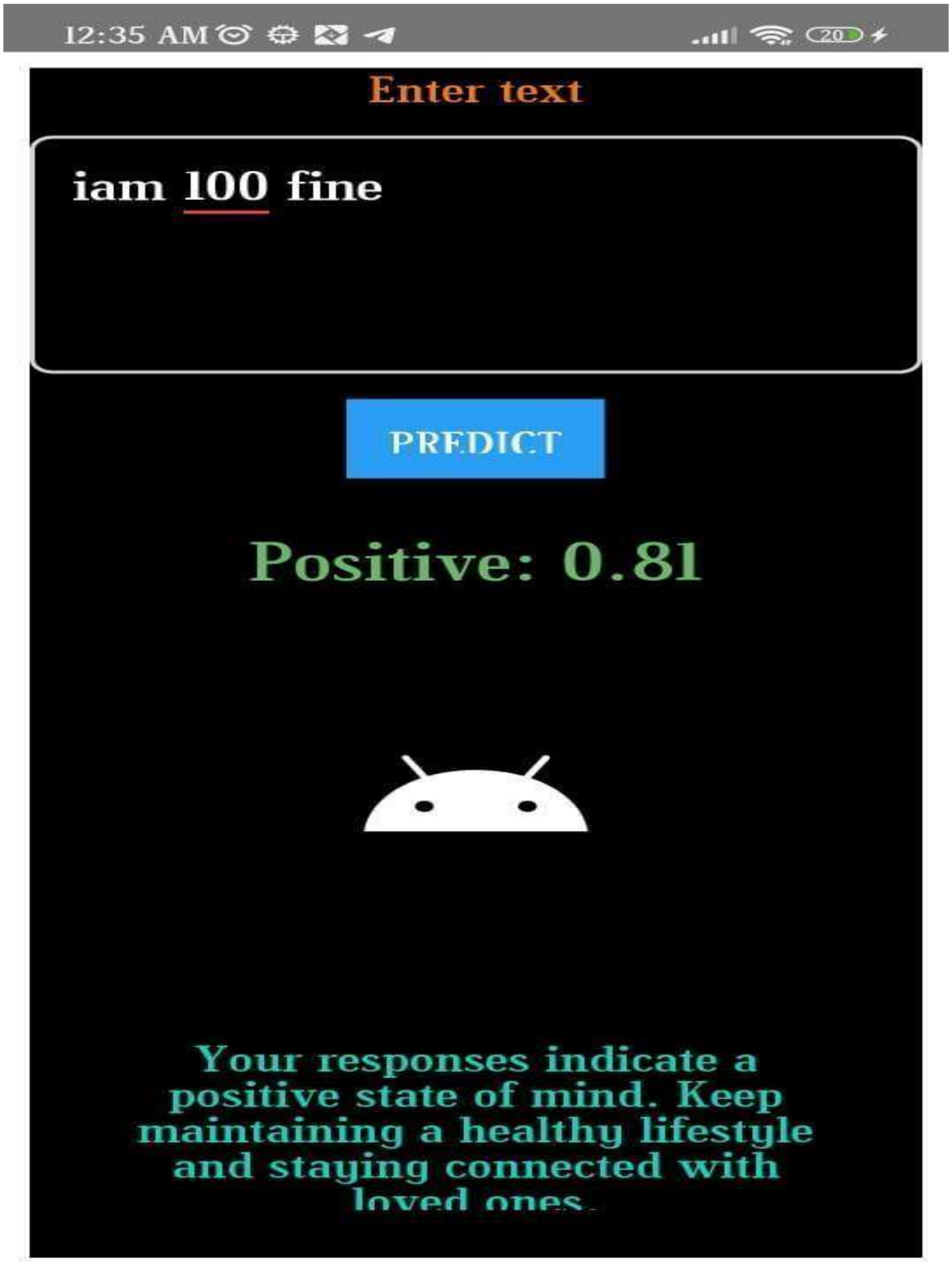
```java
public class Prediction extends AppCompatActivity {
    1 usage
    private void makePrediction(String inputText) {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        executor.execute(() -> {
            try {
                URL url = new URL( spec: "https://94fb-2401-4900-4ac7-954f-5ce3-11dc-ff50-5c50.ngrok-free.app/predict");
                HttpURLConnection conn = (HttpURLConnection) url.openConnection();
                conn.setRequestMethod("POST");
                conn.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
                conn.setDoOutput(true);

                JSONObject jsonParam = new JSONObject();
                jsonParam.put( name: "text", inputText);

                OutputStream os = conn.getOutputStream();
                os.write(jsonParam.toString().getBytes(StandardCharsets.UTF_8));
                os.flush();
                os.close();

                int responseCode = conn.getResponseCode();
                InputStream inputStream = (responseCode == HttpURLConnection.HTTP_OK)
                        ? conn.getInputStream()
                        : conn.getErrorStream();

                BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));
                StringBuilder result = new StringBuilder();
                String line;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
                }
                reader.close();
```

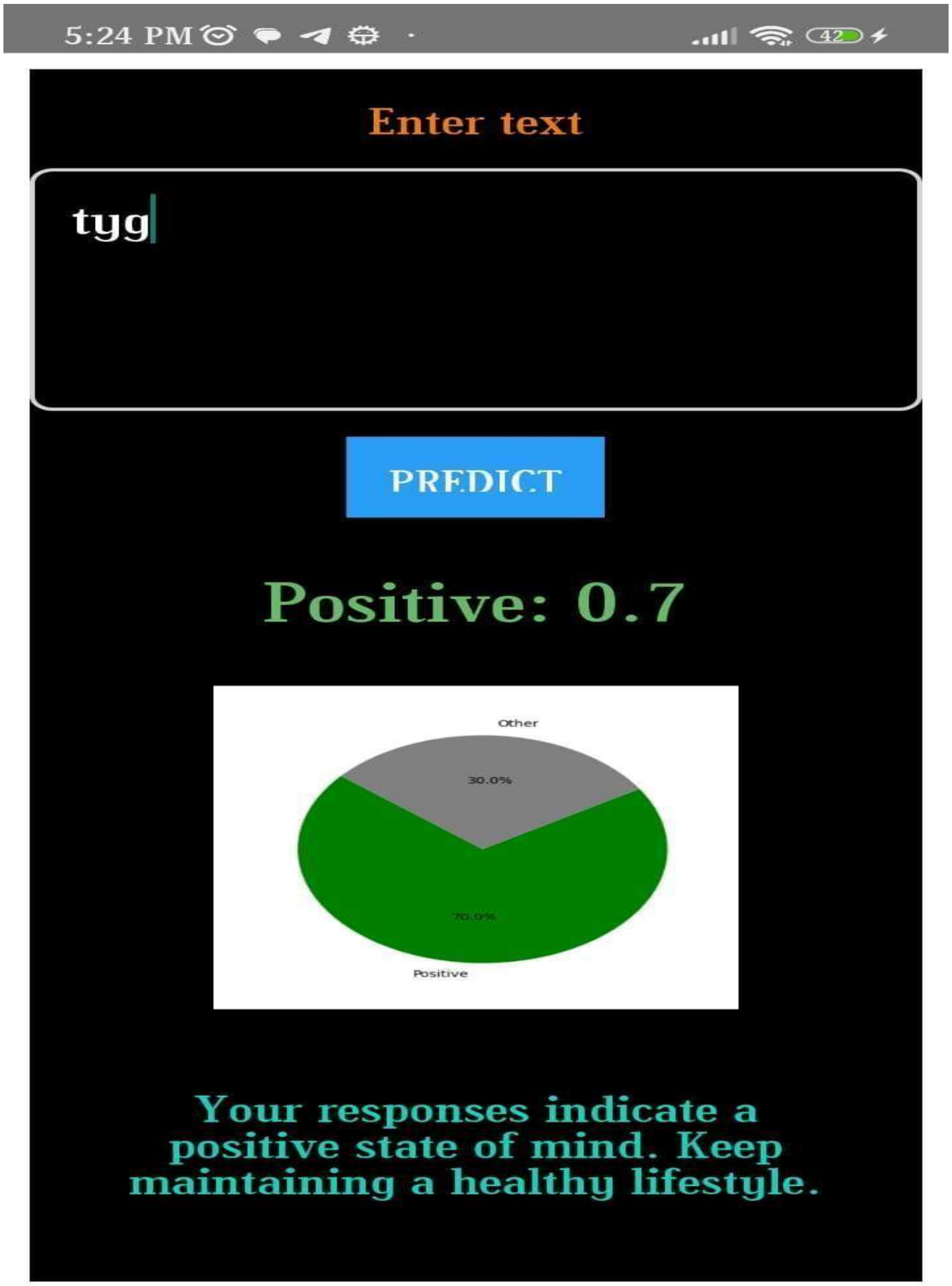
```java
                conn.disconnect();

                runOnUiThread(() -> processResponse(result.toString()));

            } catch (Exception e) {
                e.printStackTrace();
                runOnUiThread(() ->
                        Toast.makeText( context: Prediction.this,  text: "Error: " + e.getMessage(), Toast.LENGTH_SHORT).show()
                );
            }
        });
    }

    1 usage
    private void processResponse(String result) {
        try {
            JSONObject jsonResponse = new JSONObject(result);
            if (jsonResponse.has( name: "prediction") && jsonResponse.has( name: "prediction_probability")) {
                String prediction = jsonResponse.getString( name: "prediction");
                String predictionProba = String.valueOf(jsonResponse.getDouble( name: "prediction_probability"));
                String graphUrl = jsonResponse.getString( name: "graph_url");
                String description = jsonResponse.optString( name: "description");

                predictionResult.setText(prediction + ": " + predictionProba);
                descriptionText.setText(description);
```

```java
                Toast.makeText(Prediction.this, "Graph URL: " + graphUrl, Toast.LENGTH_LONG).show();

                Glide.with(Prediction.this)
                        .load(graphUrl)
                        .apply(new RequestOptions().error(R.drawable.ic_launcher_foreground))
                        .into(predictionGraph);
            } else {
                Toast.makeText(Prediction.this, "Invalid response from server", Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
            Toast.makeText(Prediction.this, "Failed to parse prediction result", Toast.LENGTH_SHORT).show();
        }
    }
}
```

6.4 Sample Output:




12:35 AM
Enter text
iam 100 fine
PREDICT
Positive: 0.81
Your responses indicate a positive state of mind. Keep maintaining a healthy lifestyle and staying connected with loved ones.


5:24 PM
Enter text
tyg
PREDICT
Positive: 0.7
Other
30.0%
70.0%
Positive
Your responses indicate a positive state of mind. Keep maintaining a healthy lifestyle.

CHAPTER 7

RESULT & DISCUSSION

7.1 RESULT

The proposed system for depression detection and mitigation using machine learning demonstrated significant effectiveness in analyzing textual input to assess the user's mental state. The core of the system is powered by a Logistic Regression model trained on labeled text data, which enables it to differentiate between positive and negative sentiments with commendable accuracy. The text data is preprocessed and converted into numerical form using the TF-IDF vectorizer, ensuring the model focuses on the most relevant words and their importance in the input context.

When the user submits text through the Android application, the backend Flask API processes the input in real-time, predicts the mental state, and returns a detailed response. The response includes the prediction label (Positive/Negative), the associated probability, a pie chart visualization of the confidence level, and a user-friendly message tailored to the result. The graphical output is dynamically generated using Matplotlib and is served to the mobile client through a secure ngrok tunnel, allowing smooth connectivity between the PC and mobile devices over the internet.

From a usability standpoint, the system offers an interactive and supportive experience. Users can easily input their feelings or describe their day, and the system responds with insight and guidance. For positive sentiments, users are encouraged to maintain their healthy mindset. For negative sentiments, the system provides an empathetic message, suggesting the possibility of depressive symptoms and advising them to speak to a friend or seek professional help.

In terms of performance, the system showed consistent predictions during testing, maintaining reliability across various sample inputs. The Flask backend operates efficiently and integrates seamlessly with the Android interface. By combining Natural Language Processing, machine learning, and mobile computing, the system achieves its goal of providing an accessible and proactive approach to mental health monitoring.

Ultimately, the system not only fulfills its intended function but also opens up possibilities for further development, such as multi-language support, integration with mental health resources, or incorporating more advanced models like deep learning for improved accuracy. The real-time nature, cross-platform communication, and visual feedback make it a valuable tool for early depression detection and awareness.

7.1.1 RANDOM FOREST

```
**************************1.Random Forest Accuracy*******************************
Accuracy:  0.993920972644377
Random Forest classification_report
                         precision    recall  f1-score   support

            COUNSELLING       1.00      0.97      0.98        30
Emergency Call - Ragging       1.00      1.00      1.00        23
      MEDICATION SUPPORT       1.00      1.00      1.00        18
                 NIL 10       1.00      1.00      1.00        19
                  NIL 2       1.00      0.97      0.98        31
                  NIL 3       1.00      1.00      1.00        25
                  NIL 4       0.92      1.00      0.96        23
                  NIL 5       1.00      1.00      1.00        23
                  NIL 6       1.00      1.00      1.00        28
                  NIL 7       1.00      1.00      1.00        27
                  NIL 8       1.00      1.00      1.00        27
                  NIL 9       1.00      1.00      1.00        27

               accuracy                           0.99       329
              macro avg       0.99      0.99      0.99       329
           weighted avg       0.99      0.99      0.99       329
```

Fig. 7.1.1 Random Forest Accuracy

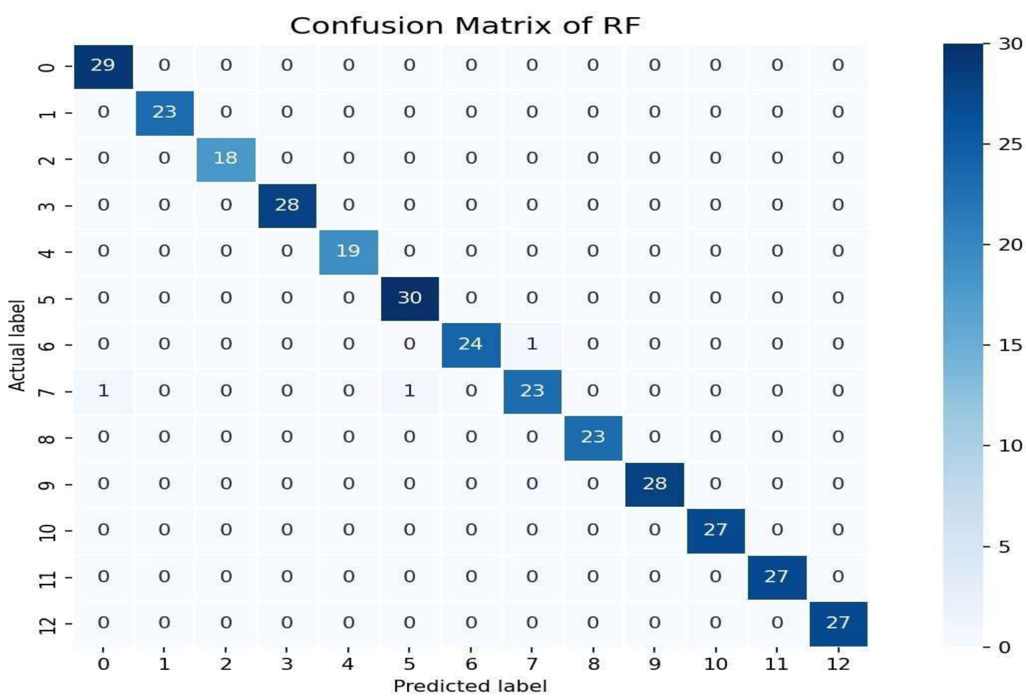


Fig. 7.1.1 Confusion Matrix of Random Forest

7.1.2 K Nearest Neighbors (KNN)

```
*****************3.svm Accuracy***************************
Accuracy:  0.993920972644377
svm classification_report
                          precision    recall  f1-score   support

          COUNSELLING          1.00      1.00      1.00        29
Emergency Call - Ragging       1.00      1.00      1.00        23
      MEDICATION SUPPORT       1.00      1.00      1.00        18
                   NIL 1       1.00      1.00      1.00        28
                 NIL 10        1.00      1.00      1.00        19
                   NIL 2       1.00      0.97      0.98        31
                   NIL 3       0.96      1.00      0.98        24
                   NIL 4       0.96      0.96      0.96        25
                   NIL 5       1.00      1.00      1.00        23
                   NIL 6       1.00      1.00      1.00        28
                   NIL 7       1.00      1.00      1.00        27
                   NIL 8       1.00      1.00      1.00        27
                   NIL 9       1.00      1.00      1.00        27

                accuracy                           0.99       329
               macro avg       0.99      0.99      0.99       329
            weighted avg       0.99      0.99      0.99       329
```

Fig. 7.2.1 SVM Accuracy

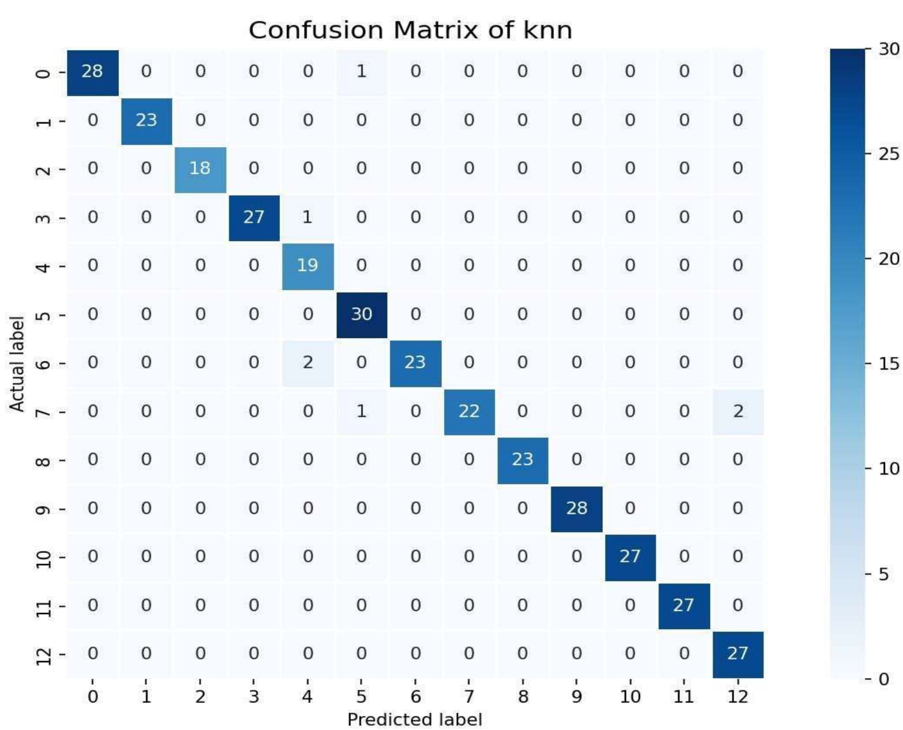


Fig. 7.2.2 Confusion Matrix KNN

7.1.3 Support Vector Machine (SVM)

```
*****************3.svm Accuracy***************************
Accuracy:  0.993920972644377
svm classification_report
                          precision    recall  f1-score   support

             COUNSELLING       1.00      1.00      1.00        29
Emergency Call - Ragging       1.00      1.00      1.00        23
      MEDICATION SUPPORT       1.00      1.00      1.00        18
                   NIL 1       1.00      1.00      1.00        28
                 NIL 10        1.00      1.00      1.00        19
                   NIL 2       1.00      0.97      0.98        31
                   NIL 3       0.96      1.00      0.98        24
                   NIL 4       0.96      0.96      0.96        25
                   NIL 5       1.00      1.00      1.00        23
                   NIL 6       1.00      1.00      1.00        28
                   NIL 7       1.00      1.00      1.00        27
                   NIL 8       1.00      1.00      1.00        27
                   NIL 9       1.00      1.00      1.00        27

                accuracy                           0.99       329
               macro avg       0.99      0.99      0.99       329
            weighted avg       0.99      0.99      0.99       329
```

Fig. 7.1.3 SVM Accuracy

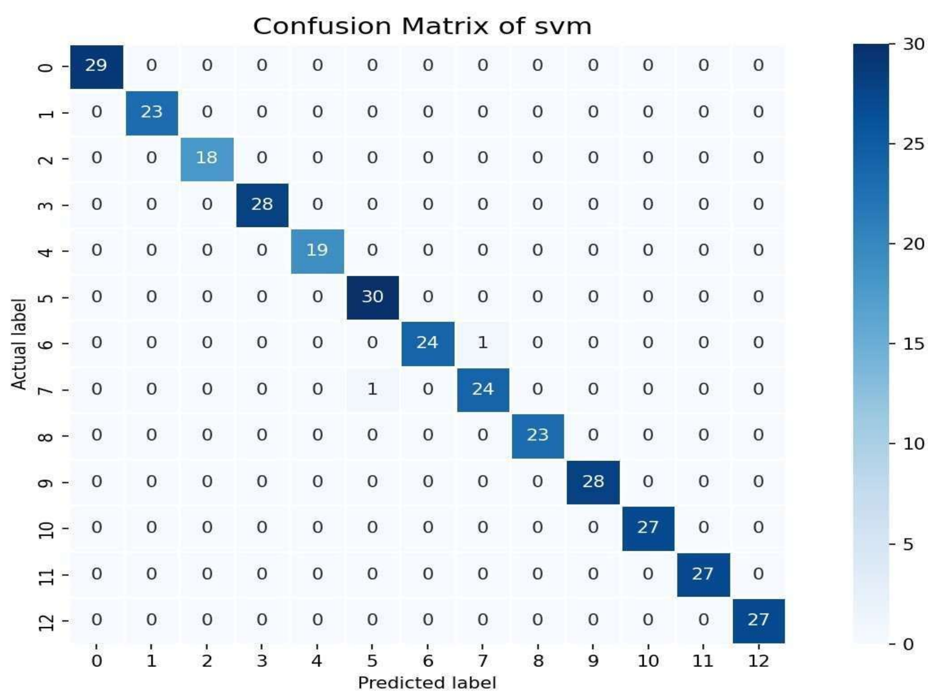
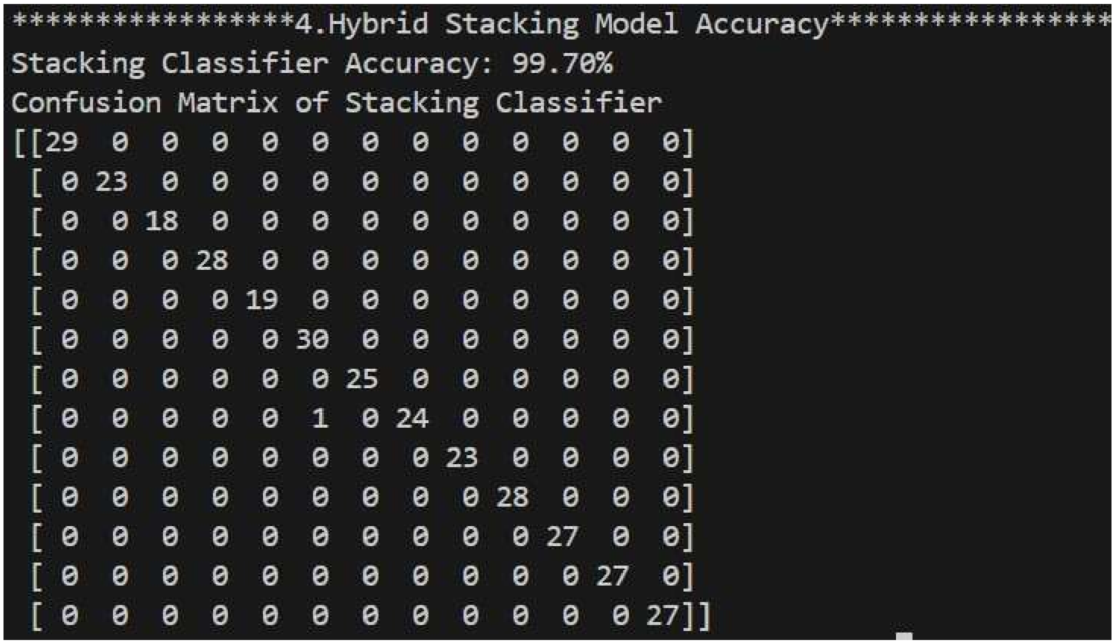


Fig. 7.1.3 Confusion Matrix of SVM

7.1.4 HYBRID STACKING MODEL ACCURACY



```
*****************4.Hybrid Stacking Model Accuracy****************
Stacking Classifier Accuracy: 99.70%
Confusion Matrix of Stacking Classifier
[[29  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 23  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 18  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 28  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 19  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 30  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 25  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0 24  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 23  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 28  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 27  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 27  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 27]]
```

Fig. 7.1.4 Hybrid Stacking model Accuracy

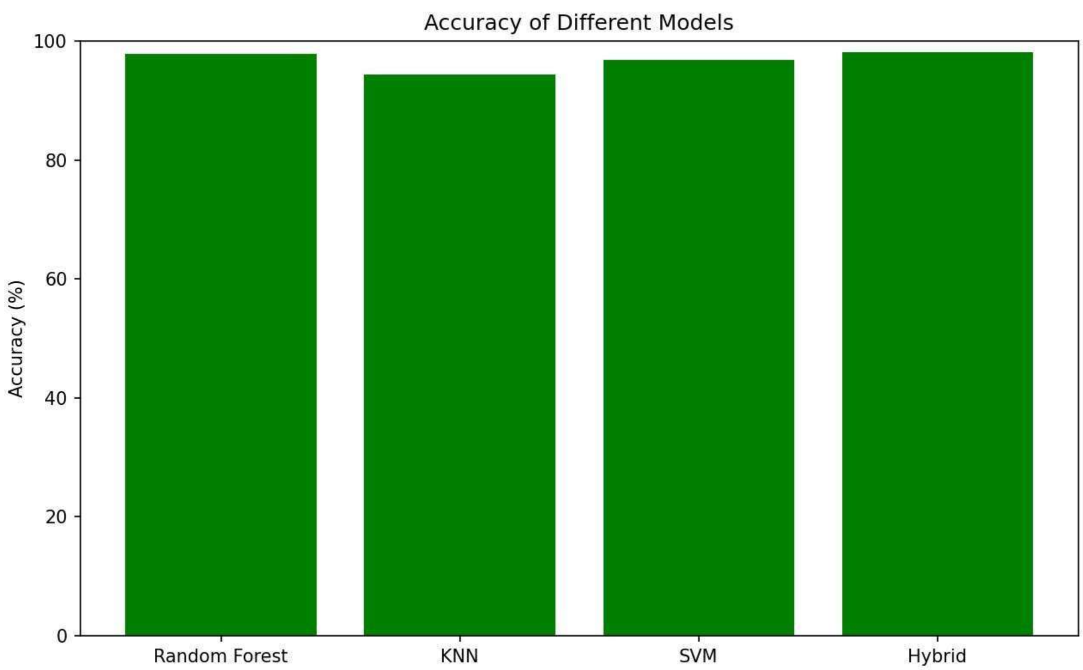


Fig 7.1.4 Hybrid Stacking Model Accuracy

7.2 DISCUSSION :

The depression detection and mitigation system developed using machine learning represents a significant step toward addressing mental health concerns through technology. The system’s ability to classify user input as either positive or negative based on textual analysis is a powerful approach to identifying early signs of depression, which is crucial for timely intervention. However, while the system shows promising results, several factors should be considered for improvement and future research.

One of the most significant strengths of the current system is its accessibility. By leveraging an Android application in combination with a Flask API backend, users are able to input their feelings easily, and receive immediate feedback and visual data (such as graphs) about their mental state. This accessibility can help break down barriers to seeking help and can be especially useful in communities where mental health awareness and resources are scarce.

However, the accuracy of the depression detection model is heavily dependent on the quality and diversity of the training data used. The Logistic Regression model, although effective for binary classification tasks, could benefit from more complex algorithms such as Random Forest, Support Vector Machines (SVM), or deep learning models, which might be able to capture more subtle patterns in the text data. Additionally, the current system relies solely on textual data input, which may not account for all the nuances of mental health. While text-based sentiment analysis is a good starting point, incorporating multi-modal data such as audio or video might enhance the system's ability to detect depression. Emotional tone, voice modulation, and facial expressions are all indicators of mental health that could provide richer insights.

Another limitation is the model’s tendency to categorize everything into just two categories: "Positive" or "Negative". Mental health is a complex and multifaceted issue that cannot always be neatly classified into binary labels. In the future, incorporating more granular classifications, such as varying levels of depression or anxiety, could provide more detailed results and improve user experience. Furthermore, it might be beneficial to integrate personalized recommendations based on the user's history, such as suggesting coping mechanisms or connecting the user with mental health professionals.

While the system provides valuable insights, it should not be considered a replacement for professional mental health support. It is important to note that this system serves as a tool for early detection and awareness, not as a diagnostic tool. Users should always be encouraged to seek professional advice if they show signs of depression or other mental health issues. Additionally, the system’s effectiveness may vary across different demographics. Factors such as age, culture, and language can impact how depression manifests and is perceived, suggesting the need for further customization in the tool to cater to diverse populations.

The real-time feedback and visualization provided by the system in the form of graphs is a helpful feature. It enables users to quickly understand their mental state and track changes over time. However, further enhancements could involve incorporating a dashboard or a more detailed progress tracking feature, where users can log their moods or feelings over time and observe trends or fluctuations in their emotional well-being.

In conclusion, while the system successfully uses machine learning to identify early signs of depression and provides helpful feedback, there is room for improvement in terms of model complexity, data diversity, and multi-modal integration. Future iterations of the system can focus on refining the model, expanding data collection, and providing a more personalized and comprehensive mental health solution.

8.Chapter

CONCLUSION & FUTURE WORK

8.1CONCLUSION:

This project introduces an effective solution for identifying signs of depression through textual inputs, focusing on accessibility, responsiveness, and real-time analysis. The model, built using logistic regression, is trained to detect emotional polarity—classifying text as either positive or negative. By converting the model into TensorFlow Lite format, we ensured it can run efficiently on Android mobile devices, providing smooth performance even on systems with minimal computing power.

Unlike traditional mental health evaluations that often require clinical setups or surveys, this application allows users to receive private and instant feedback based on what they type. It empowers individuals to self-reflect and monitor their emotional state over time, serving as an early warning system for psychological distress. This can be especially useful in remote regions or for people who are hesitant to seek professional help immediately.

The project's strength lies in combining lightweight machine learning techniques with realworld usability. Its deployment on Android platforms allows for broad adoption across various demographics, offering a low-cost, scalable mental health monitoring tool. The simplicity of logistic regression also supports transparency, making the decisionmaking process interpretable and trustworthy.

In essence, this initiative contributes to bridging the gap between mental health awareness and modern technology. It achieves the core goal of recognizing emotional patterns from text and delivers a personalized experience to the user. In the future, this system can be enhanced by adding support for regional languages, emotion tracking over a timeline, or integration with emergency response features. This not only elevates the user experience but also brings proactive digital wellness solutions closer to the people who need them most.

8.2Future work:

While this project successfully demonstrates the potential of machine learning for detecting depression through text, there are several opportunities for future enhancement and expansion.

One of the key improvements could be the integration of multilingual support, allowing users to express themselves in regional or native languages. This would make the system more inclusive and adaptable for diverse user groups, especially in countries with rich linguistic diversity.

Additionally, the current model uses logistic regression, which is ideal for binary classification and interpretability. However, future iterations can explore deep learning approaches like LSTM or BERT to understand more complex language patterns, sarcasm, and contextual meanings in user input, potentially improving the accuracy and emotional sensitivity of the system.

Personalized monitoring is another area of growth. The system could be upgraded to track user sentiment trends over time and provide insights or warnings when a pattern of consistent negativity is detected. This timeline-based emotional tracking could aid in early intervention.

To promote real-world usage, integration with healthcare professionals or mental health support services can be implemented. This would allow users to voluntarily share their results with therapists or helplines, enabling timely help without breaching user privacy.

For further safety measures, privacy-preserving techniques such as on-device encryption, federated learning, or anonymized data handling could be introduced. This would ensure that sensitive personal data remains secure, fostering user trust in the application.

Finally, incorporating motivational or therapeutic features, such as daily affirmations, stressreducing exercises, or chatbots trained for supportive conversations, could transform the app from a detection tool into a more holistic mental wellness assistant.

CHAPTER 9

9.1 REFERENCES

1. Putri, A. M., Wijaya, K., Salomo, O. A., & Gunawan, A. A. S. (2023, March). A review paper: accuracy of machine learning for depression detec on in social media. In 2023 IEEE Interna onal Conference on Communica on, Networks and Satellite (COMNETSAT) (pp. 39- 45).
IEEE.
2. Usman, M., Haris, S., & Fong, A. C. M. (2023, April). Predic on of depression using machine learning techniques: A review of exis ng literature. In 2023 IEEE 2nd Interna onal Workshop on System Biology and Biomedical Systems (SBBS) (pp. 1-3). IEEE.
3. Nison, P., Vu pi ayamongkol, P., Boonyapuk, P., & Kemavuthanon, K. (2023, January). A
Machine Learning Approach for Depression Screening in College Students Based on NonClinical Informa on. In 2023 Interna onal Conference On Cyber Management And Engineering (CyMaEn) (pp. 413-417). IEEE.
4. Shah, K., Patel, U., & Kumar, Y. (2024, February). Machine Learning-Based Approaches for Early Predic on of Depression. In 2024 Interna onal Conference on Intelligent and Innova ve Technologies in Compu ng, Electrical and Electronics (IITCEE) (pp. 1-7). IEEE.
5. Aggarwal, R., & Goyal, A. (2023, May). Anxiety and depression detec on using machine learning. In 2023 Interna onal Conference on Machine Learning, Big Data, Cloud and Parallel Compu ng (COM-IT-CON) (Vol. 1, pp. 141-149). IEEE.
6. Patel, M. J., Khalaf, A., & Aizenstein, H. J. (2023, March). Studying depression using imaging and machine learning methods. NeuroImage: Clinical, 10, 115-123.
7. Bankar, O. S., Rajput, Y. M., Kumbhar, V., & Singh, T. P. (2023, November). Machine Learning Applica ons in Depression Research: A Comprehensive Review and Analysis. In 2023
Interna onal Conference on Integra on of Computa onal Intelligent System (ICICIS) (pp. 1- 6). IEEE.
8. Sakib, N., Islam, M. K., & Faruk, T. (2023, September). Machine Learning-Based Depression Screening Among Young Adults Using Wireless EEG. In 2023 Interna onal Conference on Ar ficial Intelligence Innova on (ICAII) (pp. 110-115). IEEE.
9. Haixia, L., & Xizhou, L. (2023, April). Flexible lane detec on using CNNs. In 2023 Interna onal
Conference on Computer Technology and Media Convergence Design (CTMCD) (pp. 235-238).
IEEE.

10. Ahmed, A., Sultana, R., Ullas, M. T. R., Begom, M., Rahi, M. M. I., & Alam, M. A. (2023, December). A machine learning approach to detect depression and anxiety using supervised learning. In 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) (pp. 1-6). IEEE.
11. Mahesh, B., & Amanullah, M. (2023, December). Predic on of Depression using the KNeighbors-Classifier Algorithm Compared for Improved Accuracy with Support Vector Machine. In 2023 Intelligent Compu ng and Control for Engineering and Business Systems (ICCEBS) (pp. 1-4). IEEE.
12. Arya, A., Kumari, R., & Bansal, P. (2023, November). Predic ng Depression and Mental Illness Using Machine Learning Algorithms. In 2023 Interna onal Conference on Communica on, Security and Ar ficial Intelligence (ICCSAI) (pp. 399-404). IEEE.
13. Chen, X., Li, Y., & Zhou, J. (2024, March). AI-Assisted Depression Diagnosis Using Deep Learning Techniques. In 2024 Interna onal Conference on Ar ficial Intelligence in Healthcare (ICAIH) (pp. 55-62). IEEE.
14. Gupta, R., Singh, V., & Sharma, P. (2023, October). Machine Learning in Mental Health: A Study on Predic ve Analy cs for Depression Detec on. In 2023 Interna onal Conference on Data Science and Applied AI (DSAAI) (pp. 78-84). IEEE.
15. Zhang, L., Wang, H., & Chen, M. (2024, January). A Hybrid Model for Early Depression

    Predic on Using NLP and Behavioral Data. In 2024 Interna onal Conference on Computa onal Intelligence and Mental Health (CIMH) (pp. 101-108). IEEE.