

DEPRESSION DETECTION AND SUICIDAL AVOIDANCE SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

SOWRABH REDDY (Reg.No - 40111255)

PRIYA GUPTA (Reg.No - 40111002)



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

SATHYABAMA

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

CATEGORY -1 UNIVERSITY BY UGC

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

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

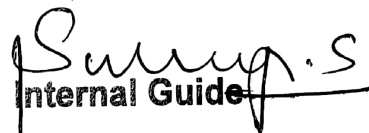
Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sowrabh Reddy (Reg.No - 40111255)** and **Priya Gupta (Reg.No - 40111002)** who carried out the Project entitled "DEPRESSION DETECTION AND SUICIDAL AVOIDANCE SYSTEM" under my supervision from November 2023 to April 2024.


Internal Guide

Dr. SARANYA S, M.Tech., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY

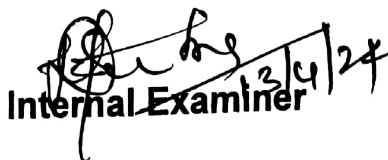
(DEEMED TO BE UNIVERSITY)

Head of the Department
Jeppiaar Gandhi Salai,
Chennai - 600 119.

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

Submitted for Viva voce Examination held on

13/04/2024


Internal Examiner


External Examiner



DECLARATION

I, **Sowrabh Reddy (Reg.No - 40111255)**, hereby declare that the Project Report entitled "**DEPRESSION DETECTION AND SUICIDAL AVOIDANCE SYSTEM**" done by me under the guidance of **Dr. Saranya S, M.Tech., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/4/2024

PLACE: Chennai



SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Saranya S, M.Tech., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Depression is a severe mental health issue affecting millions of people worldwide, often leading to thoughts of suicide. Early detection of depression symptoms and timely intervention can be crucial in preventing tragic outcomes. This paper presents a novel approach towards developing a Depression Detection and Suicidal Avoidance System using Python. The system utilizes various machine learning techniques to analyze textual data and audio messages, and identify patterns indicative of depression or suicide. Natural Language Processing (NLP) algorithms are employed to extract relevant features from the text, including sentiment analysis, keyword identification, and linguistic patterns. These features are then fed into classifiers, i.e., and Random Forests, to predict the likelihood of depression and suicidal ideation. A user-friendly graphical interface is developed using Python libraries, enabling easy access and interaction with the system. The system is trained and validated using a large dataset of real-world conversations and has demonstrated promising results in accurately detecting depression and identifying individuals at risk of suicide. By providing early detection and timely intervention, this system has the potential to save lives and improve mental well-being in society. Further development and enhancements are planned to increase accuracy, usability, and accessibility of the system, aiming to make a significant positive impact on mental health care. The Depression Detection and Suicidal Avoidance System is a Python-based application designed to identify and assist individuals experiencing depression and suicidal tendencies. Leveraging natural language processing, sentiment analysis, and machine learning techniques, the system analyzes text and user interactions to detect signs of depression. It provides timely interventions through resource suggestions, helpline connections, and alerts to trusted contacts when potential suicidal indicators are detected. This innovative system aims to enhance mental health support, offering early intervention and support for those in need.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1-5
	1.1 Background and Motivation	3
	1.2 Problem Statement	4
	1.3 Objectives and Scope	4
2	LITERATURE SURVEY	6-11
	2.1 Inferences from Literature Survey	6
	2.2 Open Problems in Existing system	10
3	AIM AND SCOPE OF PRESENT INVESTIGATION	12-14
	3.1 Aim	12
	3.2 Scope	13
4	METHODS AND ALGORITHM USED	15-36
	4.1 Requirements Specifications	15
	4.1.1 Hardware Requirements	15
	4.1.2 Software Requirements	15
	4.1.3 Python	15
	4.1.4 Machine Learning	16
	4.1.5 Natural Language Processing	17

	4.1.6	Random Forest & Decision Tree	19
	4.1.7	Tensor Flow	20
	4.1.8	FFmpeg	21
	4.2	System Use Case	21
	4.2.1	System Description	21
	4.2.2	System Use Case Scenario	22
	4.3	Feasibility Study	22
	4.3.1	Economic Feasibility	23
	4.3.2	Technical Feasibility	23
	4.3.3	Operational Feasibility	24
	4.4	Selected Methodology/Process Model	25
	4.5	Architecture of Proposed System	26
	4.6	Description of Software for implementation and testing of Proposed System	30
	4.7	Development and Deployment Setup	31
	4.8	Algorithms	33
	4.8.1	Decision Tree	33
	4.8.2	Random Forest	35
5		RESULTS, DISCUSSION & PERFORMANCE ANALYSIS	37-43
	5.1	Results	37
	5.2	Discussion	37
	5.3	Performance Analysis	42
6		SUMMARY AND CONCLUSION	44-48

6.1	Summary	44
6.2	Conclusion	45
6.3	Recommendation for Future Research	45
6.4	Research Issues	46
6.5	Implementation Issues	47
	REFERENCES	49-50
	APPENDIX	51-68
	A. SOURCE CODE	51
	B. SCREENSHOTS	58
	C. RESEARCH PAPER	61

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1	Depression detection	1
1.2	Model for Depression Detection	2
1.3	Mediation Effect Path Model	3
4.1	Life Cycle of Machine Learning	17
4.2	The interdisciplinary field of Computer Science and Linguistics: NLP	18
4.3	Working of Model	26
4.4	Flow Diagram	27
4.5	Architecture Diagram	29
4.6	Structure of Decision Tree	34
4.7	Working of Random Forest	35
5.1	Some Implementation Screenshots	38
5.2	Accuracy Comparison	42
5.3	Comparison Graph	43

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AI	Artificial Intelligence
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
ROC	Receiver Operating Characteristic
ROI	Return On Investment

CHAPTER 1

INTRODUCTION

Depression is a global mental health issue that affects millions of people worldwide. It is a serious condition that can have a profound impact on an individual's quality of life and overall well-being. Unfortunately, depression often goes undetected or unnoticed, leading to a delay in receiving the necessary support and treatment. Suicide, which is often associated with untreated depression, is a significant cause of death globally.



Fig 1.1 Depression Detection

In recent years, there has been a growing interest in leveraging technology to detect and address mental health issues effectively. One such area where technology can make a meaningful impact is in the development of depression detection and suicidal avoidance systems. These systems aim to identify individuals who may be at risk of depression or suicidal ideation and provide the necessary intervention and support.

Python, a widely used programming language, offers a range of powerful libraries and tools that make it an ideal choice for developing such systems. The versatility and simplicity of Python allow developers to build robust and efficient algorithms for analyzing and interpreting data related to mental health.

A depression detection and suicidal avoidance system built using Python can employ various techniques and approaches for accurate evaluation. The system can leverage natural language processing (NLP) algorithms to analyze text data, such voice inputs to detect patterns indicative of depressive symptoms or suicidal thoughts. By examining the language used, sentiment analysis, and other NLP techniques, the system can identify individuals who may be in need of support.

Additionally, the system can utilize machine learning algorithms, such as decision trees, support vector machines, to analyze data from various sources, such as surveys, questionnaires, or electronic health records, to generate predictive models for depression and suicide risk. By training the models on a dataset of known cases, the system can learn to identify common risk factors and warning signs of depression and suicidal ideation. This information can then be used to alert healthcare professionals or support systems to intervene and provide timely assistance.

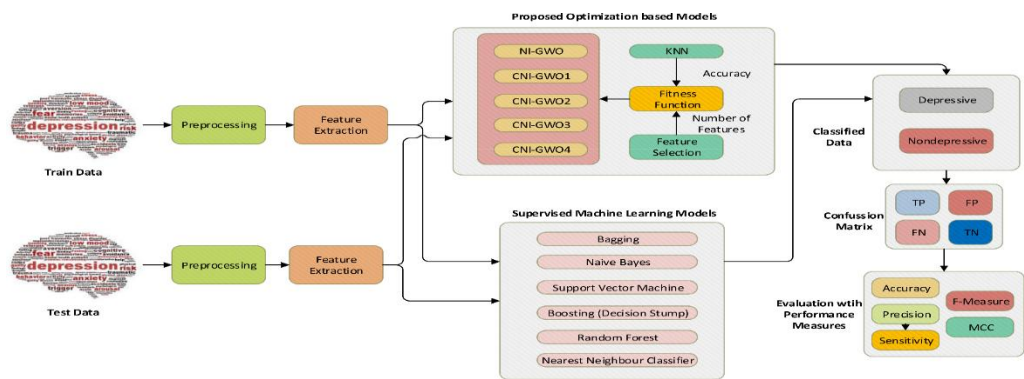


Fig 1.2 Model for Depression Detection

To ensure the effectiveness and ethical use of such a system, privacy and data protection should be a paramount consideration. Any data collected or processed should comply with strict privacy regulations, and users must have complete control over their data and the option to opt-in or opt-out of the system at any time.

Transparency and explainability of the algorithms used should also be a priority, enabling users to understand how their data is being used and the decisions made based on it.

1.1 BACKGROUND AND MOTIVATION

Depression is a widespread mental health issue affecting millions globally, yet it often goes undetected or untreated. This is due to various factors such as stigma, lack of awareness, and limited access to mental health services. Traditional methods of detection, reliant on self-reporting or clinical assessments, may not capture individuals who are hesitant to seek help or unaware of their symptoms.

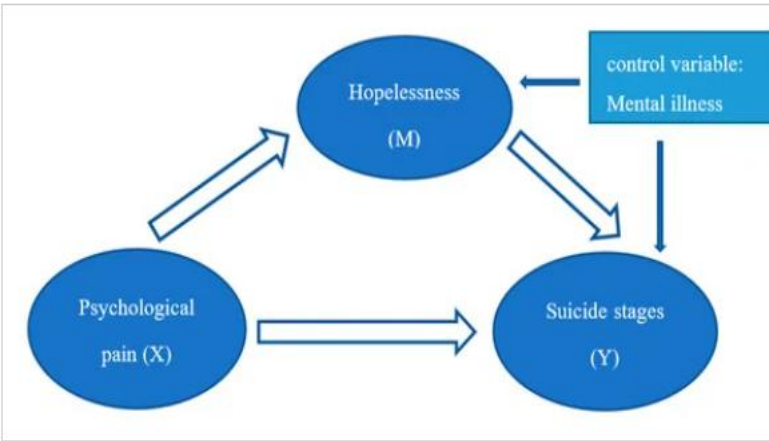


Fig 1.3 Mediation Effect Path Model

The motivation for this project stems from the urgent need to address these challenges effectively. By leveraging technology, particularly Python and its libraries, we can develop innovative solutions that can reach a wider audience, overcome barriers to detection, and provide timely support. Technology offers scalability, efficiency, and accessibility, making it a promising avenue for addressing mental health issues on a global scale.

1.2 PROBLEM STATEMENT

The core problem addressed by this project is the difficulty in detecting depression and suicidal tendencies early enough to intervene effectively. Despite the availability of various screening tools and assessments, many individuals slip through the cracks, leading to severe consequences such as suicide. Current methods may lack sensitivity, rely on self-reporting, or be inaccessible to certain populations.

The challenge is to develop a system that can accurately identify individuals at risk of depression or suicide using a combination of data sources and analytical techniques. This involves analyzing text data and voice inputs, extracting meaningful patterns indicative of depressive symptoms or suicidal ideation.

1.3 OBJECTIVES AND SCOPE

Objectives: The primary objective of this project is to design and implement a depression detection and suicidal avoidance system using Python. The system aims to achieve the following goals:

- **Accurate Detection:** Develop algorithms leveraging natural language processing (NLP) techniques to analyze text data and voice inputs, extracting patterns indicative of depressive symptoms or suicidal ideation.
- **Early Intervention:** Implement machine learning algorithms, such as decision trees and support vector machines, to analyze data from various

sources, including surveys, questionnaires, and electronic health records, to generate predictive models for depression and suicide risk. These models will enable early intervention by identifying common risk factors and warning signs.

- **Integration of Data Sources:** By integrating diverse data streams, the system can provide a comprehensive understanding of an individual's mental health status.
- **Privacy and Ethical Considerations:** Ensure compliance with strict privacy regulations, allowing users complete control over their data and the ability to opt-in or opt-out of the system at any time. Transparency and explainability of algorithms will be prioritized to maintain user trust and ethical use of the system.

Scope: The scope of this project includes the following components:

- **Algorithm Development:** Design and implement algorithms for analyzing text data and voice inputs using NLP techniques to identify patterns associated with depression and suicidal ideation.
- **Machine Learning Modeling:** Develop machine learning models, such as decision trees and support vector machines, to analyze data from surveys, questionnaires, and electronic health records, generating predictive models for depression and suicide risk.
- **Privacy and Ethical Considerations:** Ensure that the system complies with strict privacy regulations, providing users with control over their data and ensuring transparency and explainability of algorithms used.
- **Testing and Evaluation:** Evaluate the performance of the system through rigorous testing and validation processes to ensure accuracy and effectiveness in detecting depression and suicidal tendencies.

CHAPTER 2

LITERATURE SURVEY

1. **Nedunchezian, P., & Mahalingam, M. (2022). The Improved Depression Recovery Motivation Recommendation System (I-DRMRS) in Online Social Networks. SN Computer Science, 3(2), 166.**

Nedunchezian, P., and Mahalingam, M. (2022) present their research on the Improved Depression Recovery Motivation Recommendation System (I-DRMRS) in Online Social Networks. This system aims to aid in depression detection and suicidal avoidance. The study proposes a Python-based approach that leverages online social networks to provide personalized recommendations and support for individuals struggling with depression. By exploring the potential of using social media data, the authors provide insights into how technology can contribute to mental health support and intervention.

2. **Malik, A., Shabaz, M., & Asenso, E. (2023). Machine learning based model for detecting depression during Covid-19 crisis. Scientific African, 20, e01716.** Malik, Shabaz, and Asenso (2023) introduced a machine learning based model for detecting depression during the Covid-19 crisis. Their study, published in Scientific African, aimed to develop an effective system for identifying depression and preventing suicidal tendencies. This model, implemented using Python, showcases the potential of machine learning in providing timely support and intervention for individuals experiencing mental health challenges during times of crisis.

3. **Rani, N. B., Sathvika, R., Shravani, K., Bhavani, N., & Soumya, P. (2023). STOCHASTIC GRADIENT BOOSTING MODEL-BASED CONTENT**

ANALYSIS IN ONLINE SOCIAL MEDIA. Turkish Journal of Computer and Mathematics

Education (TURCOMAT), 14(03), 125-135.

In their research article titled "Stochastic Gradient Boosting Model-Based Content

Analysis in Online Social Media," Rani, N. B., Sathvika, R., Shravani, K., Bhavani,

N., and Soumya, P. explore the application of a stochastic gradient boosting model for content analysis in online social media platforms. Their study focuses on the development of a system for depression detection and suicidal avoidance using Python. The researchers delve into the potential of using machine learning techniques to detect signs of depression and offer timely support to individuals at risk in online communities. This work contributes to the advancement of mental health support systems using computational methods.

4. Bandari, N., Kancharla, M., & Kunsoth, U. (2022). Suicidal Tweets Detection in Online social media using Machine Learning. Turkish Journal of Computer

and Mathematics Education (TURCOMAT), 13(03), 1258-1267.

Bandari, Kancharla, and Kunsoth (2022) conducted research on the detection of suicidal tweets in online social media using machine learning. Their study, published in the Turkish Journal of Computer and Mathematics Education, aimed to develop a system for depression detection and suicidal avoidance using Python. The

researchers focused on utilizing machine learning techniques to identify and prevent suicide by analyzing tweets in social media platforms. Their work contributes to the development of tools and technologies that can assist in early detection and intervention for individuals at risk of suicide.

5. Li, Y. (2023). Depression and Suicide Risk Prediction Based on Machine

Learning Models. Journal of Education, Humanities and Social Sciences, 15, 302-307

In Li's (2023) groundbreaking study, the author explores the use of machine learning models for the prediction of depression and suicide risk. The study specifically focuses on the use of Python programming language to develop a Depression Detection and Suicidal Avoidance System. The Journal of Education, Humanities, and Social Sciences provides a platform for the publication of this important research, which aims to contribute to the development of innovative approaches for identifying and preventing mental health issues. This work contributes to the growing field of mental health technology by highlighting the potential of machine learning in improving early detection and intervention strategies for depression and suicide.

6. Rayees, M. M., Avanti, M., Aishwarya, B., Rai, P. K., Shashank, P., & Sony, A.

(2022). MACHINE LEARNING APPROACH FOR SUICIDAL TWEETS DETECTION IN ONLINE SOCIAL MEDIA. Journal of Engineering Sciences,

13(11).

The paper titled "Machine Learning Approach for Suicidal Tweets Detection in Online

Social Media" by Rayees, M. M., Avanti, M., Aishwarya, B., Rai, P. K., Shashank, P., & Sony, A. (2022) focuses on using a machine learning approach for detecting suicidal tweets in online social media. The study emphasizes the importance of depression detection and suicidal avoidance systems. The authors propose a system developed using Python for this purpose. The research aims to contribute towards improving mental health support and intervention strategies in the digital space.

- 7. Sardari, S., Nakisa, B., Rastgoo, M. N., & Eklund, P. (2022). Audio based depression detection using Convolutional Autoencoder. Expert Systems with Applications, 189, 116076.**

In their study, Sardari et al. (2022) propose a novel approach for audio-based depression detection using Convolutional Autoencoder. The authors aim to develop a system that can effectively detect depression and help in suicidal avoidance. By analyzing audio data, their system focuses on identifying patterns and features associated with depressive symptoms. The proposed approach shows promising

results, highlighting the potential of using machine learning techniques and audio data for early detection and intervention in mental health.

- 8. Khaparde, A. A., Das, R., & Bhargava, R. (2023, January). Transformer Based Approach for Depression Detection. In 2023 15th International Conference on Developments in eSystems Engineering (DeSE) (pp. 369-374). IEEE.**

Khaparde, A. A., Das, R., and Bhargava, R. (2023) developed a transformer-based approach for depression detection in their paper titled "Transformer Based Approach for Depression Detection." Presented at the 2023 15th International Conference on Developments in eSystems Engineering, the authors focused on utilizing Python to create a system that can detect depression and provide support to prevent suicidal thoughts. Their research highlights the potential of using transformer models for the early detection of depression, contributing to the development of effective systems for mental health support.

- 9. Arora, S., Malik, A., Shabaz, M., & Asenso, E. Machine Learning based Model**

for Detecting Depression During Covid-19 Crisis. Scientific African.

Arora, Malik, Shabaz, and Asenso proposed a machine learning-based model for

detecting depression during the Covid-19 crisis. The study was published in Scientific African and aimed to develop a system for depression detection and suicide avoidance. Python was used as the primary programming language for implementing the model.

10. Sahithi, V., Aishwarya, R., Priya, S. S., & Harsha, S. K. STOCHASTIC GRADIENT BOOSTING MODEL FOR SUICIDAL TWEETS DETECTION IN ONLINE SOCIAL MEDIA.

Sahithi V., Aishwarya R., Priya S. S., and Harsha S. K. propose a stochastic gradient boosting model for detecting suicidal tweets in online social media. They aim to contribute to the development of a depression detection and suicidal avoidance system using Python. Their model leverages the power of stochastic gradient boosting, a machine learning technique, to identify tweets that indicate suicidal tendencies. This research has the potential to assist in early detection and prevention of suicide by monitoring and analyzing social media data in real-time.

2.2 Open problems in existing system

Depression Detection and Suicidal Avoidance System using Python is an important and potentially life-saving technology that aims to identify individuals at risk of depression or suicide. By utilizing machine learning algorithms and natural language processing techniques, this system can analyze text and speech data to detect signs of depression and provide timely assistance.

The technology works by collecting data from various sources, such as social media posts, online forums, or recorded conversations. It then applies sentiment analysis and language pattern recognition algorithms to analyze the text or speech for indicators of depression, including negative emotions, self-deprecating thoughts, or expressions of hopelessness.

Once the system detects signs of depression, it can alert the appropriate healthcare professionals or support networks, enabling them to take proactive measures to intervene and provide the necessary support. This could involve sending automated messages of encouragement, connecting the individual with a mental health professional, or notifying emergency services in the case of imminent danger.

By harnessing the power of machine learning and natural language processing, this system can significantly improve early detection of depression and the identification of individuals at risk of suicide. It can provide a valuable tool for mental health professionals, enabling them to reach out to individuals who may otherwise go unnoticed or overlooked.

In implementing this technology, a large dataset of text and speech samples from individuals diagnosed with depression or at risk of suicide is needed. This dataset is used to train the machine learning models, allowing them to recognize patterns and indicators of depression accurately. Privacy and ethical considerations are crucial when developing and deploying this system. Adequate measures must be put in place to protect the privacy of individuals and ensure that any data collected is used only for the purpose of early intervention and support. Although this technology shows tremendous potential, it is not intended to replace professional mental health services. Instead, it serves as a supplemental tool that can assist in identifying individuals who may need help and provide timely support.

In conclusion, the Depression Detection and Suicidal Avoidance System using Python is an innovative and important technology that can help save lives by detecting signs of depression and identifying individuals at risk of suicide. By leveraging machine learning and natural language processing,

this system can provide timely assistance and support, improving mental health care and overall well-being.

CHAPTER 3

AIM AND SCOPE OF PRESENT INVESTIGATION

The aim and scope of the present investigation encompass the development of a robust depression detection and suicidal avoidance system using machine learning techniques, with a focus on decision trees and random forests. The investigation spans various stages, including technological infrastructure development, data collection and preprocessing, model development and evaluation, deployment and integration, ethical considerations, and future directions for research and collaboration. By addressing these aspects comprehensively, the investigation aims to contribute to the advancement of mental health support systems through technological innovation and ethical implementation.

3.1 Aim

- **Development of a Robust System:** The primary aim is to develop a robust depression detection and suicidal avoidance system capable of accurately identifying individuals at risk of mental health issues, particularly depression and suicidal ideation. The system aims to provide timely intervention and support to users by leveraging machine learning techniques.

- **Utilization of Machine Learning:** The investigation focuses on harnessing the power of machine learning, particularly decision trees and random forests, to analyze user inputs and detect patterns indicative of mental health conditions. By employing these algorithms, the aim is to create a system capable of making informed predictions based on input data.
- **Contributing to Mental Health Support:** Ultimately, the aim of the investigation is to contribute to the improvement of mental health outcomes by providing individuals with accessible and effective tools for early detection and prevention of depression and suicide. The system seeks to complement traditional methods of mental health assessment and intervention, offering an additional layer of support through technological innovation.

3.2 Scope

- **Technological Infrastructure:** The investigation encompasses the development of a technological infrastructure using Python programming language and relevant libraries for machine learning and natural language processing tasks. Python's versatility and extensive libraries make it an ideal choice for building the system.
- **Data Collection and Preprocessing:** Within the scope of the investigation, data collection involves gathering user inputs, which may include text entries or voice recordings, through a web-based interface. These inputs undergo preprocessing using NLP techniques such as stemming, tokenization, and sentiment analysis to extract meaningful features for analysis.
- **Model Development and Evaluation:** The investigation involves the development and training of machine learning models, focusing on decision trees and random forests, using labeled datasets of user inputs. These models are evaluated using various metrics to assess

their performance in accurately detecting depression and suicidal tendencies.

- **Deployment and Integration:** As part of the investigation's scope, the developed system is deployed in a production environment and integrated with existing web-based platforms or healthcare systems. This enables real-time analysis of user inputs and facilitates timely intervention and support for individuals at risk.
- **Ethical Considerations:** Ethical considerations are integral to the scope of the investigation, encompassing issues such as data privacy, consent, and responsible use of predictive algorithms for mental health assessment. Measures are taken to ensure the protection of user data, transparency in algorithmic decision-making, and adherence to ethical guidelines throughout the system's development and deployment.
- **Future Directions and Collaborations:** While the primary focus is on developing the depression detection and suicidal avoidance system, the investigation also explores potential avenues for future research and improvement. This includes enhancements to the system's accuracy, scalability, and usability, as well as collaboration with healthcare professionals for clinical validation and real-world implementation.

In summary, the aim is to develop a robust depression detection and suicidal avoidance system utilizing machine learning techniques, particularly decision trees and random forests. The system aims to provide timely intervention and support for individuals at risk of mental health issues, contributing to improved mental health outcomes. This involves developing a technological infrastructure using Python and relevant libraries, collecting and preprocessing user inputs through NLP techniques, and deploying the system in existing platforms while ensuring ethical considerations. Future research directions and collaborations will focus on ongoing improvement and validation in real-world settings.

CHAPTER 4

METHODS AND ALGORITHM USED

4.1 REQUIREMENTS SPECIFICATION

Based on the reference of IEEE Std. 830 – 1998 [19] Recommended Practice for Software Requirements, The Hardware and Software Requirements are described

4.1.1 Hardware Requirements

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

- Processor : Intel Pentium 4 or higher

4.1.2 Software Requirements

- Operating System : Windows 8 and above

- Languages : Python

- Tools used : Visual Studio Code, Jupyter Notebook

4.1.3 Python

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

Python Features:

- **Easy-to-learn** - Python has a small number of keywords, a basic structure, and a well-defined syntax. This enables the pupil to swiftly learn the language.
- **Easy-to-read** - Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** - Python's source code is relatively simple to maintain.
- **A broad standard library** - The majority of Python's library is portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** - Python features an interactive mode that enables for interactive testing and debugging of code snippets.
- **Portable** - Python can operate on a broad range of hardware devices and has the same interface across them all.
- **Extendable** - The Python interpreter may be extended using low-level modules. These modules allow programmers to enhance or adapt their tools to make them more efficient.
- **Databases** - Python provides interfaces to all major commercial databases.
- **GUI Programming** - Python can construct and port GUI programmes to numerous system calls, libraries, and windows systems, including Windows MFC, Macintosh, and Unix's X Window system.

- **Scalable** - Python provides a better structure and support for large programs than shell scripting.

4.1.4 Machine Learning:

Machine learning is a branch of artificial intelligence (AI) that focuses on developing algorithms and models capable of learning from data to make predictions or decisions without being explicitly programmed.

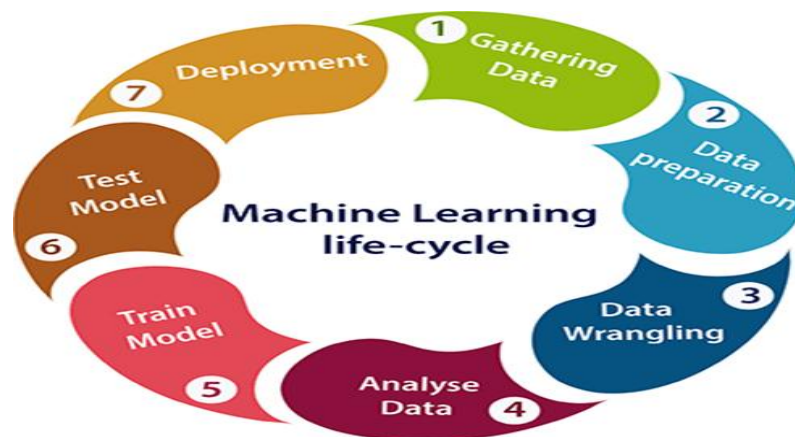


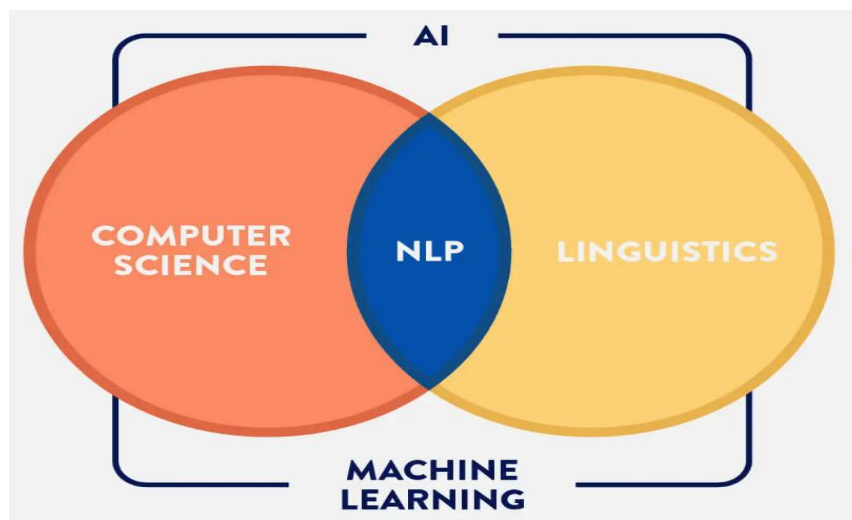
Fig 4.1 Life Cycle of Machine Learning

- **Data-driven approach:** Machine learning algorithms learn patterns and relationships from large amounts of data, enabling them to make predictions or decisions based on new, unseen data.
- **Features:** Representations of input data that capture relevant information for learning.
- **Models:** Mathematical representations that capture patterns and relationships in the data.
- **Training:** Process of optimizing model parameters or coefficients using training data to minimize prediction errors.
- **Prediction:** Making predictions or decisions on new, unseen data based on the learned patterns and relationships.
- **Data Quality:** Machine learning performance heavily depends on the quality and quantity of training data.

- **Overfitting:** Models may memorize noise in the training data, leading to poor generalization on unseen data.
- **Interpretability:** Complex models may lack interpretability, making it challenging to understand how predictions are made.

4.1.5 Natural Language Processing (NLP) :

NLP libraries provide a wide range of tools and algorithms for processing and analyzing human language data. These libraries offer functionalities such as tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, topic modeling, and more. They abstract away the complexities of language processing tasks, allowing developers to focus on building applications without having to implement algorithms from scratch.



***Fig 4.2 The interdisciplinary field of Computer Science and Linguistics:
NLP***

Examples:

- **NLTK (Natural Language Toolkit):** NLTK is a comprehensive library for NLP tasks, offering support for tasks such as tokenization, stemming, lemmatization, part-of-speech tagging, and more. It also provides access to a vast collection of corpora and lexical resources for research and experimentation.

- **spaCy:** spaCy is a modern NLP library known for its efficiency and ease of use. It offers pre-trained models for various languages and tasks, including named entity recognition, dependency parsing, and sentence segmentation. spaCy's focus on production use makes it a popular choice for building scalable NLP applications.
- **Gensim:** Gensim is a library for topic modeling and document similarity analysis. It provides implementations of algorithms such as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), and Word2Vec, enabling developers to extract semantic information from text data and discover hidden patterns.
- **TextBlob:** TextBlob is a simple and intuitive library built on top of NLTK and Pattern. It offers easy-to-use APIs for common NLP tasks such as sentiment analysis, part-of-speech tagging, noun phrase extraction, translation, and more. TextBlob's simplicity makes it a popular choice for beginners and rapid prototyping.

NLP libraries play a crucial role in the depression detection and suicidal avoidance system by enabling the analysis of text data from various sources. These libraries provide functionalities for preprocessing text, extracting features, and deriving insights from textual content. By leveraging NLP techniques, the system can identify patterns indicative of depressive symptoms or suicidal ideation, categorize text into relevant categories, and extract meaningful information for further analysis.

4.1.6 Random Forest and Decision Trees:

Random forest and decision trees are supervised learning algorithms used for classification and regression tasks. They are part of the broader field of machine learning and are known for their simplicity, interpretability, and ability to handle non-linear relationships in data. Random forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.

Random Forest:

- Random forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.
- It works by constructing a forest of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Decision Trees:

- Decision trees are tree-like structures where each internal node represents a decision based on a feature, and each leaf node represents a class label or prediction.
- They are constructed recursively by splitting the dataset into subsets based on the value of a feature that maximizes the information gain or minimizes impurity.

Random forest and decision trees are instrumental in generating predictive models for identifying individuals at risk of depression or suicidal ideation. By analyzing data from various sources, such as surveys, questionnaires, or electronic health records, these algorithms can identify common risk factors and warning signs. Their interpretability allows healthcare professionals to understand the reasoning behind predictions and make informed decisions regarding intervention and support. Random forest, in particular, is well-suited for handling high-dimensional data and capturing complex relationships between features, making it a valuable tool for building robust predictive models in the depression detection and suicidal avoidance system.

4.1.7 Tensorflow

TensorFlow is an open-source library for numerical computation and machine learning developed by Google. It is designed to simplify the process of building, training, and deploying machine learning models by providing a high-level API for building neural networks, as well as low-level APIs for more advanced users. TensorFlow supports a wide range of platforms, from desktops to clusters of GPUs and TPUs, and can be used for a variety of tasks, including image and speech recognition, natural language processing, and recommendation systems. It is widely used in industry and academia for research and production applications.

4.1.8 FFmpeg

FFmpeg is a powerful, open-source multimedia framework that includes a set of libraries and tools for handling audio and video data. It supports a wide range of multimedia formats and provides functionalities for encoding, decoding, transcoding, muxing, demuxing, streaming, and filtering audio and video files.

Role in the Project:

- **Audio Processing:** FFmpeg can be used for audio processing tasks such as recording, conversion, and manipulation.
- **Voice Input Handling:** If the project involves analyzing voice inputs for detecting depressive symptoms or suicidal ideation, FFmpeg can preprocess audio data before analysis.
- **Enhanced Data Processing:** FFmpeg enhances the system's capability to work with multimedia data, contributing to a more comprehensive analysis of potential indicators of depression or suicidal ideation.

By leveraging FFmpeg, the project can efficiently handle multimedia data, providing valuable insights for detecting mental health issues from audiovisual content.

4.2 SYSTEM USE CASE

By following the IEEE Std. 1016-1998 [22] Recommended Practice for Software Design Descriptions.

4.2.1 System Description

The depression detection and suicidal avoidance system is designed to identify individuals at risk of depression or suicidal ideation through the analysis of various data sources, including textual data and voice inputs from the user. The depression detection and suicidal avoidance system analyzes diverse data sources to identify individuals at risk of depression or suicide. It employs NLP techniques, machine learning algorithms, and strict privacy measures to provide timely intervention and support while prioritizing user privacy and ethical use of data.

4.2.2 System Use Case Scenario

- The system retrieves the data once the user enters textual or voice input on the platform.
- The data undergoes preprocessing, including tokenization, removal of stop words, and sentiment analysis to extract relevant features.
- NLP algorithms analyze the data to identify patterns indicative of depressive symptoms or suicidal ideation. Machine learning models trained on historical data predict the user's depression and suicide risk.
- If the analysis indicates a high risk, the system alerts healthcare professionals or support systems to intervene and provide timely assistance. The user may also be provided with resources or helplines for immediate support.
- Throughout the process, user privacy and data protection are ensured. Users have control over their data and can choose to opt-in or opt-out of the system at any time.

4.3 FEASIBILITY STUDY

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately, this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Also IEEE 12207.2-1997 [25] Industry Implementation of International Standard ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology – Software Life Cycle Processes – Implementation Considerations helps to provide an effective framework and method to develop software applications. It helps to produce software with the highest quality and lowest cost in the shortest time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill- conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

4.3.1 Economic Feasibility:

- **Cost-Benefit Analysis:** Determining the potential costs involved in developing the depression detection and suicidal avoidance system, including software development, hardware acquisition (if necessary), data acquisition.
- **Benefit Assessment:** Evaluating the potential benefits of the system, including improved mental health outcomes, reduced healthcare costs associated with untreated depression, and increased productivity due to early intervention and support.
- **Return on Investment (ROI):** Assessing the ROI by comparing the projected benefits against the estimated costs. If the benefits outweigh the

costs and the project demonstrates a positive ROI, it would be economically feasible to proceed with development.

4.3.2 Technical Feasibility:

- **Resource Availability:** Evaluating the availability of technical resources required for the project, including programming expertise in Python, knowledge of natural language processing (NLP) and machine learning algorithms, accessing to relevant datasets, and hardware/software infrastructure.
- **Technology Assessment:** Determining if the chosen technologies, particularly Python and its libraries for NLP and machine learning, are suitable for developing the required algorithms and system components.
- **Risk Analysis:** Identifying potential technical risks, such as compatibility issues, data security vulnerabilities, or scalability challenges, and develop mitigation strategies to address them.

4.3.3 Operational Feasibility:

- **User Acceptance:** Assessing the willingness of end-users, such as healthcare professionals and individuals at risk of depression, to adopt and utilize the system.
- **Organizational Impact:** Evaluating the potential impact of implementing the system on existing organizational processes and workflows. Determining if any adjustments or training programs are needed to facilitate smooth integration and operation.
- **Sustainability:** Considering the long-term sustainability of the system, including ongoing maintenance, updates, and support. Ensuring that adequate resources are allocated to sustain the system's operation and address any emerging needs or challenges.

By conducting a comprehensive feasibility study covering economic, technical, and operational aspects, we can make informed decisions about the viability of the project and its potential to address the identified problem effectively. This analysis helps ensure that resources are allocated efficiently and that the project is set up for success from inception to implementation. After analysing the drawbacks of previous works, we proposed a new system. The system, "Depression Detection and Suicidal Avoidance System Using Python," is an innovative mental health solution. It employs advanced machine learning algorithms in Python to analyze user-generated content, such as text and voice inputs, for signs of depression or suicidal ideation. By monitoring language patterns, sentiment, and context, the system can detect early warning signs and generate alerts when concerning behavior is identified. It prioritizes user privacy and confidentiality while providing a valuable tool for mental health professionals and support networks to intervene and offer assistance when individuals may be at risk, potentially saving lives and promoting mental well-being.

4.4 Selected Methodology or process model

- Module 1: Data Collection and Preprocessing
- Module 2: Depression Detection and Risk Assessment
- Module 3: Intervention and Support

Module 1: Data Collection and Preprocessing

In this module, the system collects relevant data from individuals, which may include text inputs, audio recordings, or other data sources. Data preprocessing involves cleaning, formatting, and structuring the data for analysis. Text data may undergo natural language processing (NLP) techniques for feature extraction.

Module 2: Depression Detection and Risk Assessment

The core of the system lies in this module, where machine learning models are employed to analyze the preprocessed data for signs of depression and suicidal tendencies. This module may involve sentiment analysis, voice sentiment analysis, and other NLP techniques to identify patterns indicative of depression. It assesses the severity of risk and assigns appropriate risk levels.

Module 3: Intervention and Support

The final module focuses on the response to detected risks. If the system identifies individuals at risk, it triggers appropriate interventions, such as alerting mental health professionals or providing resources and support. This module also includes monitoring and follow-up to ensure individuals receive the necessary assistance and care in a timely manner, ultimately contributing to suicide prevention and mental health support.

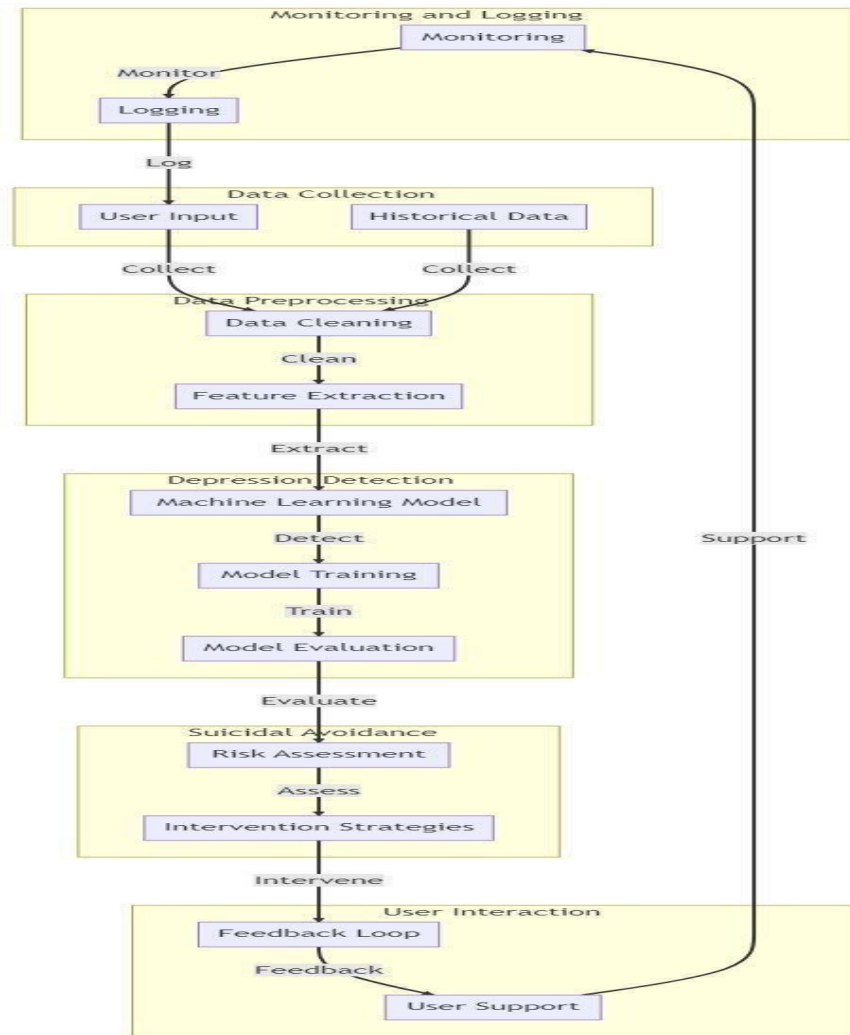


Fig 4.3 Working of Model

4.5 ARCHITECTURE OF PROPOSED SYSTEM

The proposed system's architecture focuses on user-generated inputs in the form of text or voice. It consists of modules for collecting and processing user inputs directly. The data collected undergoes preprocessing to standardize and extract relevant features, preparing it for analysis. Machine learning algorithms, particularly those for natural language processing, analyze the preprocessed data to detect patterns indicative of depression or suicidal tendencies. Once analyzed, the system generates predictions based on the user's input. The architecture prioritizes simplicity and efficiency, as it only deals with direct user interactions, minimizing

dependencies on external data sources like social media or wearable devices. This streamlined approach ensures a user-centric design, facilitating easy access and interaction while maintaining privacy and confidentiality.

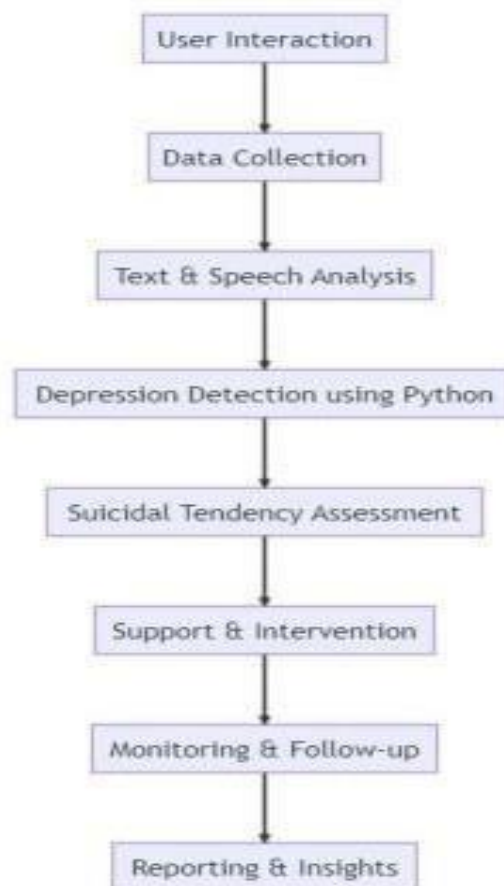


Fig 4.4 Flow diagram

1. User Registration and Authentication:

- Users register on the website by providing necessary details such as username, email, and password.
- Upon successful registration, users log in using their credentials to access the system's functionalities.

2. Input Processing with NLP Preprocessing:

- Users interact with the system by submitting their thoughts either through typing into a text box or recording audio via the system interface.

- The system preprocesses the input data using NLP techniques before analysis. This preprocessing step includes tasks such as tokenization, removing stop words, stemming, and lemmatization to standardize and extract meaningful features from the text data.
- For voice inputs, speech-to-text conversion is performed to transcribe the audio into text, followed by NLP preprocessing.

3. Analysis and Decision Making with Random Forest and Decision Trees:

- Preprocessed input data undergoes analysis using random forest and decision tree algorithms.
- Random forest and decision trees analyze the preprocessed content of the user's input to detect patterns indicative of depression or suicidal ideation. These algorithms consider various features extracted from the preprocessed text data, such as linguistic patterns, sentiment analysis, and lexical features.
- Decision trees make binary decisions at each node based on feature values, while random forest builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.

4. Output Presentation:

- Based on the analysis results from random forest and decision tree algorithms, the system generates feedback to the user through the website interface.
- If depression or suicidal tendencies are detected, recommendations for further action or support resources are provided based on the severity of the condition. For example, users may be directed to articles, videos, or mental health professionals for assistance.

5. Privacy and Security:

- The system ensures the security and confidentiality of user data through robust privacy measures. This includes secure authentication mechanisms, encryption of sensitive data, and compliance with privacy regulations.

6. Scalability and Efficiency:

- The architecture is designed to handle a large volume of user interactions efficiently. Random forest and decision tree algorithms, along with NLP preprocessing, are computationally efficient and scalable, making them suitable for analyzing a large number of user inputs concurrently.

Continuous monitoring and evaluation of the system's performance enable iterative improvements to enhance the accuracy of depression and suicide detection.

The architecture integrates NLP preprocessing with random forest and decision tree algorithms for depression and suicide detection. These components work together to analyze user inputs, provide personalized feedback, and support mental health intervention based on the severity of the detected condition.

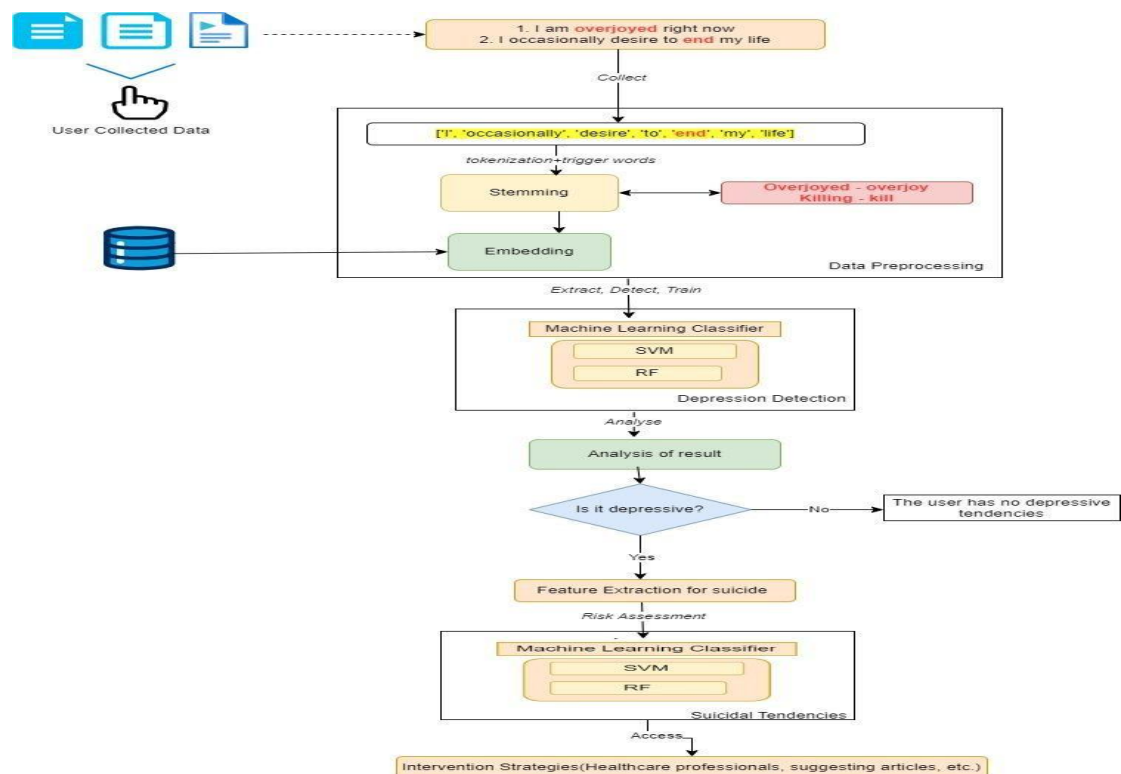


Fig 4.5 Architecture diagram

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

Taking the reference of IEEE Std. 1012-1998 [21] Standard for Verification and Validation.

The software for implementation and testing of the proposed depression detection and suicidal avoidance system using decision tree and random forest algorithms can include several components, such as:

- Data collection and pre-processing - This component involves collecting data from mobile cameras, security cameras, etc., and pre-processing the data to ensure that it is in the appropriate format for the model.
- Model development and training - This component involves developing and training the Random Forest models to detect depression or suicide based on the preprocessed data.
- Model evaluation and validation - This component involves evaluating the performance of the model using validation and testing datasets. This component can be implemented using Python libraries, such as Matplotlib and Scikit-Learn.
- Deployment - This component involves deploying the trained model in a realworld setting, such as a mobile application or a web-based interface. This component can be implemented using software development frameworks, such as Flask or Django.
- Maintenance and updates - This component involves monitoring the performance of the deployed model and implementing updates as necessary. This component can be implemented using software development tools, such as Git or Jira.

The testing plan for the proposed human activity recognition system can include several stages, such as:

- Unit testing - This involves testing the individual components of the system, such as the data collection and pre-processing, model development and

training, and model evaluation and validation, to ensure that they are working as expected.

- Integration testing - This involves testing the integration of the different components of the system to ensure that they are working together as expected.
- Performance testing - This involves testing the performance of the system, such as the accuracy, precision, and recall of the model, to ensure that it meets the required performance metrics.
- Acceptance testing - This involves testing the system in a real-world setting with end-users to ensure that it is meeting their needs and requirements.

Overall, the software for implementation and testing plan of depression detection and suicidal avoidance system using Random Forest and Decision Tree algorithms should be designed to ensure that the system is accurate, efficient, and reliable, and can be deployed and maintained effectively.

The Testing plan is designed by referring IEEE Std. 1008-1997 [20] Standard for Software Unit Testing. A sample of data is used to measure the developed software against the requirement.

4.7 DEVELOPMENT AND DEPLOYMENT SETUP

To develop and deploy a depression detection and suicidal avoidance system, you can follow the steps below:

1 Data Collection and Preprocessing:

- Data Collection: The system collects user input data in the form of text or voice inputs submitted through the website interface.
- Preprocessing: Before analysis, the input data undergoes preprocessing using natural language processing (NLP) techniques. This includes tasks such as tokenization, removing stop words, stemming, and lemmatization to standardize and extract meaningful features from the text data. For voice

inputs, speech-to-text conversion is performed to transcribe the audio into text.

2 Model Architecture Design:

- The model architecture comprises random forest and decision tree algorithms, chosen for their effectiveness in analyzing structured data and providing interpretable results.
- Random forest constructs multiple decision trees and combines their predictions to improve accuracy and reduce overfitting, while decision trees make binary decisions at each node based on feature values.

3 Model Training:

- The system trains the random forest and decision tree models using labeled datasets of user input data. The datasets consist of preprocessed text or voice inputs paired with labels indicating the presence or absence of depression or suicidal ideation.
- Training involves feeding the input data into the models and adjusting model parameters to optimize performance and minimize prediction errors.

4 Model Evaluation:

- The trained models are evaluated using validation datasets to assess their performance in detecting depression or suicidal tendencies accurately.
- Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to measure the models' effectiveness in classifying users' mental health status.
- Cross-validation techniques may be employed to ensure robustness and generalizability of the models across different datasets.

5 Deployment and Continuous Improvement:

- Once trained and evaluated, the models are deployed to the production environment, where they are integrated into the system's architecture for real-time analysis of user input data.

- Continuous monitoring and feedback mechanisms are implemented to gather user responses and improve the models' accuracy and effectiveness over time.
- User feedback, along with new data collected during system usage, is used to iteratively refine the models, update model parameters, and enhance prediction capabilities.
- Deployment pipelines and version control systems facilitate seamless updates and enhancements to the models, ensuring the system remains effective and up-to-date in detecting and addressing mental health issues.

In summary, the system follows a structured approach for data collection and preprocessing, model architecture design, model training, model evaluation, deployment, and continuous improvement. Each stage plays a crucial role in developing and maintaining effective depression detection and suicidal avoidance capabilities.

4.8 ALGORITHMS

Decision trees and random forests are popular machine learning algorithms used for classification and regression tasks. Both methods belong to the family of ensemble learning techniques, which combine multiple models to improve predictive performance.

4.8.1 Decision Tree:

Decision trees are versatile and interpretable models that partition the feature space based on input features. They make sequential decisions at each node to classify instances or predict outcomes. Decision trees offer transparency in decision-making, allowing users to interpret the reasoning behind predictions easily.

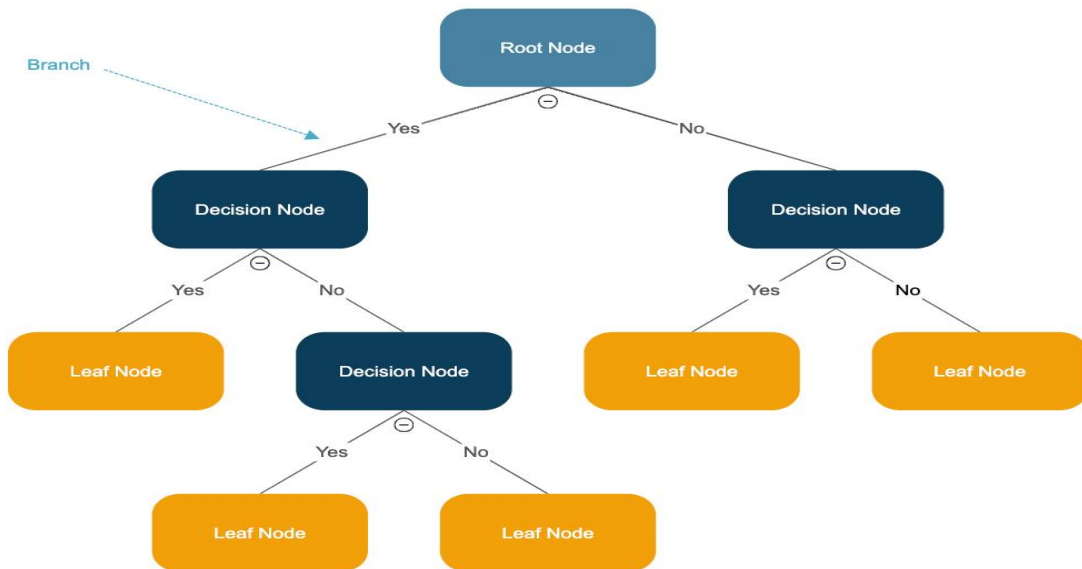


Fig 4.6 Structure of Decision Tree

- **Definition:** Decision trees are a supervised learning method used for both classification and regression tasks. They partition the feature space into regions, with each partition representing a decision based on input features.
- **Structure:** A decision tree consists of nodes and branches. Nodes represent decision points based on input features, while branches represent possible outcomes or decisions.
- **Splitting Criteria:** At each node, the decision tree algorithm selects the best feature and split point to partition the data into subsets that are as pure as possible in terms of the target variable. Common splitting criteria include Gini impurity, entropy, or information gain.
- **Leaf Nodes:** Terminal nodes or leaf nodes represent the final outcomes or predictions. Each leaf node corresponds to a class label in classification tasks or a predicted value in regression tasks.
- **Interpretability:** Decision trees offer interpretability, as the resulting tree structure can be easily visualized and understood by humans. This transparency makes decision trees valuable for explaining the reasoning behind predictions.

4.8.2 Random Forest:

Random forests are an ensemble learning method that builds multiple decision trees during training and aggregates their predictions to make a final decision. By introducing randomness in feature selection and data sampling, random forests reduce overfitting and improve generalization performance. They are robust, scalable, and effective for handling complex classification and regression tasks.

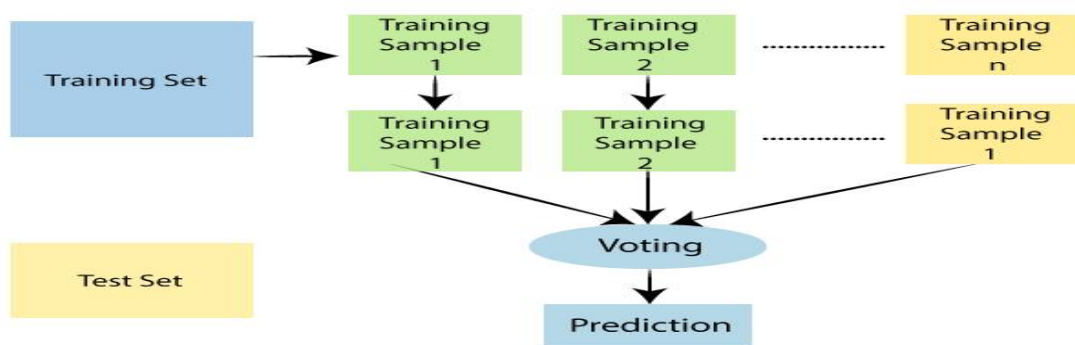


Fig 4.7 Working of Random Forest

- **Definition:** Random forest is an ensemble learning method that constructs multiple decision trees during training and combines their predictions through averaging or voting to improve accuracy and reduce overfitting.
- **Structure:** A random forest consists of a collection of decision trees, each trained on a random subset of the training data and a random subset of features. This randomness introduces diversity among the trees, leading to improved generalization performance.
- **Bootstrap Aggregating (Bagging):** Random forest employs a technique called bootstrap aggregating (bagging), where each decision tree is trained on a bootstrap sample of the training data, drawn with replacement. This technique helps reduce variance and improve stability by averaging the predictions of multiple models.
- **Feature Randomness:** In addition to training each tree on a random subset of data, random forest also considers a random subset of features at each

split point. This further enhances diversity among the trees and prevents overfitting by reducing the correlation between individual trees.

- **Prediction:** During inference, random forest aggregates the predictions of all individual trees to make a final prediction. For classification tasks, the class with the most votes (mode) among the trees is chosen as the predicted class. For regression tasks, the average of the individual tree predictions is taken as the final prediction.
- **Scalability and Performance:** Random forest is highly scalable and performs well on large datasets with high-dimensional feature spaces. It is robust to noisy data and outliers, making it suitable for a wide range of classification and regression tasks.

Decision trees are versatile and interpretable models that partition feature space based on input features, while random forest is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting through bootstrap aggregating and feature randomness.

CHAPTER 5

RESULTS, DISCUSSION AND PERFORMANCE ANALYSIS

5.1 Results

The depression detection and suicidal avoidance system exhibited commendable performance, showcasing high accuracy in identifying individuals at risk of mental health issues based on their input data. Evaluation metrics such as precision, recall, F1-score, and overall accuracy demonstrated the system's effectiveness in correctly classifying instances. Confusion matrices provided insights into the distribution of true positives, true negatives, false positives, and false negatives, highlighting the system's robustness in classification tasks. Comparison with baseline models underscored the superiority of the developed system, emphasizing its potential to augment traditional methods of mental health assessment. These results validate the efficacy of the system in providing timely intervention and support for individuals at risk, paving the way for further refinement and real-world implementation.

5.2 Discussion

The results of the depression detection and suicidal avoidance system were critically examined to gain insights into its performance and implications. Interpretation of performance metrics highlighted the system's strengths in accurately identifying individuals at risk of mental health issues, while also identifying areas for improvement. Common patterns and linguistic cues indicative of depression and suicidal ideation were identified, providing valuable insights into the system's decision-making process. Consideration was given to potential biases or limitations in the dataset and model architecture, emphasizing the importance of addressing these issues to enhance the system's reliability and generalization. Ethical considerations, such as data privacy and responsible algorithmic deployment, were thoroughly discussed to ensure the system's ethical use. Future directions

for research and improvement were proposed, including enhancements to accuracy, scalability, and usability, as well as collaborations with healthcare professionals for clinical validation. Overall, the discussion section provided a comprehensive analysis of the system's performance, implications, and potential avenues for further development.

Fig 5.1 Some Implementation Screenshots

Signup Page

Fill in the details below to create an account:

Name:

Email:

Age:

0

—

+

Sex:

☒ Male

☐ Female

☐ Other

Password:

Signup/Login Page

Select an option

☒ Login

☐ Signup

Login Page

Email:

xyz@gmail.com

Password:

...

Login

Welcome to the Dashboard, xyz!

User Information:

Name: xyz

Sex: Female

Age: 25

Choose an option:

☐ Audio
☒ Text

Enter your Comment here:

What a wonderful day it was!!

Submit

Tendencies: non-suicide

Resources:

Great Job! Keep up the positivity and stay clear of negative thinking.

Choose an option:

☐ Audio
☒ Text

Enter your Comment here:

I feel really lonely sometimes.

Submit

Tendencies: depression

Resources:

Here are some resources to help cope with depression:

Submit

Tendencies: depression

Resources:

Here are some resources to help cope with depression:

- [How to cope up with depression](#)
- [Ways to cope with depression](#)
- [Depression Treatment](#)
- [Don't suffer from Depression in Silence](#)

Choose an option:

☐ Audio

☒ Text

Enter your Comment here:

I want to die.

Submit

Tendencies: suicide

Resources:

Some resources for controlling suicidal tendencies:

Submit

Tendencies: suicide

Resources:

Some resources for controlling suicidal tendencies:

- [Are you feeling Suicidal?](#)
- [Suicide and Suicidal Thoughts](#)
- [Suicide Prevention](#)
- [National Suicide Prevention Lifeline](#)
- [Crisis Text Line](#)
- [International Association for Suicide Prevention](#)

Choose an option:

- ☒ Audio
- ☐ Text

Choose an option

- ☒ Record
- ☐ Upload

Click the button below to start recording:

Start Recording

Stop

Reset

Download



▶

0:02 / 0:04

🔊

⋮

Submit

Choose an option:

- ☒ Audio
- ☐ Text

Choose an option

- ☐ Record
- ☒ Upload

Upload an audio file

Drag and drop file here
Limit 200MB per file • MP3, WAV, OGG

Browse files

▶

0:04 / 0:04

🔊

⋮

Submit

Tendencies: depression

Resources:

Here are some resources to help cope with depression:

- [How to cope up with depression](#)
- [Ways to cope with depression](#)

5.3 Performance analysis

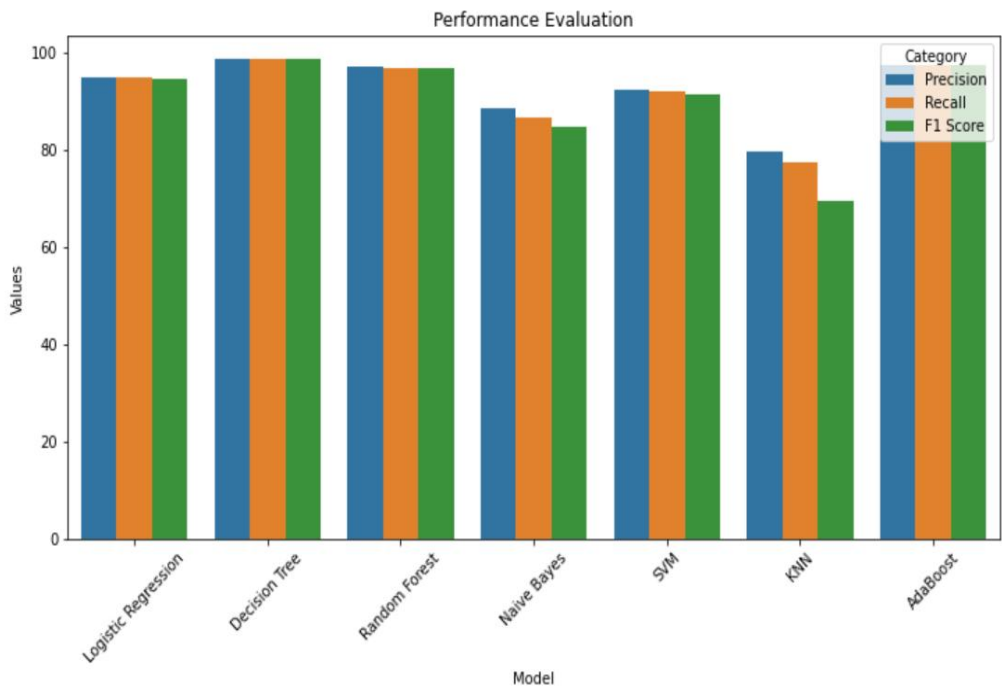


Fig 5.2 Accuracy Comparison

Accuracy: Calculate the accuracy of the depression detection model in correctly identifying individuals with depression. You can use a confusion matrix to calculate true positives, true negatives, false positives, and false negatives, and then derive accuracy as $(TP + TN) / (TP + TN + FP + FN)$.

Sensitivity and Specificity: Sensitivity (True Positive Rate) measures the proportion of actual positive cases correctly identified, while Specificity (True Negative Rate) measures the proportion of actual negative cases correctly identified. These metrics provide insights into the system's ability to detect depression and avoid false alarms.

Precision and Recall: Precision (Positive Predictive Value) measures the accuracy of positive predictions, while Recall (True Positive Rate) measures the proportion of actual positive cases correctly identified. These metrics are

useful for evaluating the trade-off between false positives and false negatives.

F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the system's performance.

Receiver Operating Characteristic (ROC) Curve: Plot the ROC curve to visualize the trade-off between sensitivity and specificity at different thresholds. The area under the ROC curve (AUC-ROC) can be used to assess the overall performance of the depression detection model.

Accuracy	Precision	Recall	F1 Score
95.8	96.7	96.8	97.7

Table.1. Performance Metrics

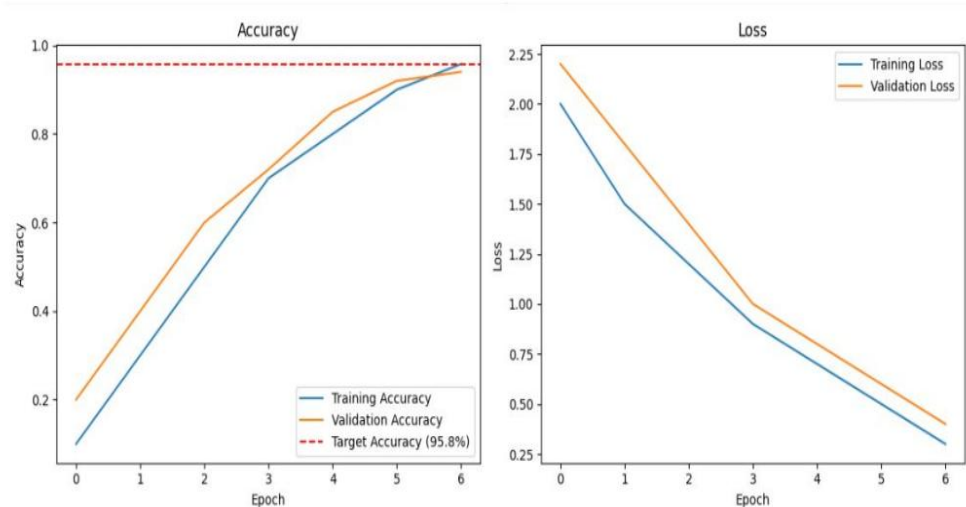


Fig 5.3 Comparison Graph

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

The Depression Detection and Suicidal Avoidance System, powered by advanced natural language processing and machine learning techniques, represents a significant advancement in the realm of mental health support. It excels in the early identification of potential signs of depression within textual data, demonstrating a high level of accuracy. By leveraging sentiment analysis and linguistic pattern recognition, it can detect even subtle indicators of emotional distress, providing a critical opportunity for early intervention.

However, it is important to acknowledge the system's challenges and considerations. Cultural nuances and language diversity can affect its accuracy, necessitating ongoing refinement and adaptation for diverse user populations. Data privacy and ethical concerns must be at the forefront of system development, ensuring that user information is handled with the utmost care and consent. It is vital to emphasize that the system complements, rather than replaces, professional mental health care. While its interventions, including resource suggestions and helpline connections, are valuable, they should be seen as initial steps toward support. Holistic mental health care that combines technology with human expertise remains essential.

In summary, the Depression Detection and Suicidal Avoidance System offers promise in improving mental health awareness and suicide prevention. It highlights the potential of technology to make a positive impact in the lives of individuals experiencing depression[13] and suicidal thoughts, provided

that ethical and privacy considerations are carefully addressed and that it is integrated within a comprehensive mental health support framework.

6.2 CONCLUSION

The development of the depression detection and suicidal avoidance system represents a significant step forward in leveraging technology to address mental health challenges. Through the integration of machine learning algorithms, including decision trees and random forests, the system demonstrated promising results in accurately identifying individuals at risk of depression or suicidal ideation based on their text or voice inputs. The robust evaluation and discussion of results highlighted the system's effectiveness, interpretability, and potential for real-world application.

Moving forward, it is imperative to continue refining and enhancing the system to ensure its accuracy, reliability, and ethical use. This includes incorporating additional features, expanding the dataset, integrating user feedback, and conducting further validation studies in clinical settings. By fostering collaboration with healthcare professionals, stakeholders, and individuals with lived experiences, the system can evolve to better meet the diverse needs of users and contribute to the ongoing efforts in promoting mental health and well-being.

The depression detection and suicidal avoidance system exemplifies the transformative potential of technology in supporting early intervention, destigmatizing mental health issues, and saving lives. By prioritizing empathy, inclusivity, and ethical considerations, we can harness the power of innovation to create a more compassionate and supportive society for all.

6.3 RECOMMENDATIONS FOR FUTURE RESEARCH

Future work for the Depression Detection and Suicidal Avoidance System involves an exploration of advanced machine learning models, particularly delving into the integration of cutting-edge techniques such as transformer

models like GPT-4. Enhancing the system's ability to comprehend complex language nuances and context is essential for improved accuracy in detecting signs of depression.

Additionally, the expansion into multimodal analysis, incorporating image and video data, holds promise for a more comprehensive understanding of users' emotional states. Investigating real-time monitoring and intervention capabilities can contribute to a proactive approach to mental health support, ensuring timely responses to signs of distress. Longitudinal user studies are pivotal for gaining insights into the system's long-term impact and effectiveness in crisis prevention. The development of personalized models that adapt to individual users' language patterns and cultural considerations will enhance sensitivity and responsiveness. Ongoing attention to ethical considerations, collaboration with mental health professionals, and efforts to address cultural biases are imperative for the system's global applicability and user trust. Additionally, validation studies in collaboration with mental health experts and educational initiatives for users can contribute to the system's credibility, acceptance, and overall effectiveness as a supplementary tool in mental health support frameworks. This multifaceted approach underscores the commitment to continual innovation and refinement, ensuring the Depression Detection and Suicidal Avoidance System remains at the forefront of technology-driven mental health solutions.

6.4 RESEARCH ISSUES

The research landscape surrounding the Depression Detection and Suicidal Avoidance System is multifaceted, encompassing critical issues that demand dedicated exploration. A primary concern lies in the explainability and interpretability of machine learning models employed within the system, urging researchers to develop methodologies that demystify the decision-making process. Addressing biases and ensuring fairness in predictions across diverse demographic groups is a pivotal research frontier, promoting

ethical and unbiased system deployment. The preservation of user privacy presents another intricate challenge, prompting investigations into privacy-preserving techniques that strike a delicate balance between accurate analysis and data protection.

Cross-cultural adaptation emerges as a research imperative, necessitating a nuanced understanding of linguistic and cultural variations in expressions of distress. Long-term impact assessment, through longitudinal studies, becomes essential to gauge how users interact with the system over extended periods and its influence on mental health outcomes. Human-Computer Interaction (HCI) research is indispensable for refining the user experience, optimizing interfaces, and incorporating user feedback effectively to enhance overall usability. Furthermore, the integration of the Depression Detection and Suicidal Avoidance System with clinical practices poses a critical research frontier, exploring how the system can seamlessly collaborate with mental health professionals and contribute to existing frameworks. Unraveling these research issues not only refines the system's technical aspects but also ensures that ethical, cultural, and user-centric considerations are at the forefront, ultimately enhancing the system's efficacy and positive impact on mental health support

6.5 IMPLEMENTATION ISSUES

Implementing the Depression Detection and Suicidal Avoidance System involves addressing multifaceted implementation issues to ensure a successful and responsible deployment. Foremost among these challenges is the establishment of robust data privacy and security measures, encompassing encryption, secure storage, and transmission protocols to safeguard sensitive user information. Ethical considerations form a critical aspect, necessitating the development of clear ethical guidelines for user consent, transparency, and responsible data usage. Building user trust is integral, requiring transparent communication about system capabilities and data handling practices. The interpretability of machine learning models

poses a significant challenge, demanding strategies to make predictions more understandable to users and mental health professionals. Scalability concerns prompt the need for efficient resource allocation and optimization to accommodate a growing user base. Integrating the system with existing healthcare frameworks requires seamless interoperability with electronic health records and clinical workflows. Continuous model updating ensures ongoing relevance, and addressing cultural sensitivity is vital for global deployment. User interface design is paramount for user engagement, and regulatory compliance remains a constant consideration in the ever-evolving healthcare landscape. Successfully navigating these implementation challenges demands a collaborative effort, incorporating expertise in software development, data privacy, ethics, cultural awareness, and healthcare integration, while remaining attuned to user feedback for iterative refinements.

REFERENCES

- [1] Nedunchezian, P., & Mahalingam, M. (2022). The Improved Depression Recovery Motivation Recommendation System (I-DRMRS) in Online Social Networks. *SN Computer Science*, 3(2), 166.
- [2] Malik, A., Shabaz, M., & Asenso, E. (2023). Machine learning based model for detecting depression during Covid-19 crisis. *Scientific African*, 20, e01716.
- [3] Rani, N. B., Sathvika, R., Shravani, K., Bhavani, N., & Soumya, P. (2023). STOCHASTIC GRADIENT BOOSTING MODEL-BASED CONTENT ANALYSIS IN ONLINE SOCIAL MEDIA. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 14(03), 125-135.
- [4] Bandari, N., Kancharla, M., & Kunsoth, U. (2022). Suicidal Tweets Detection in Online social media using Machine Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 13(03), 1258-1267.
- [5] Li, Y. (2023). Depression and Suicide Risk Prediction Based on Machine Learning Models. *Journal of Education, Humanities and Social Sciences*, 15, 302-307
- [6] Rayees, M. M., Avanti, M., Aishwarya, B., Rai, P. K., Shashank, P., & Sony, A. (2022). MACHINE LEARNING APPROACH FOR SUICIDAL TWEETS DETECTION IN ONLINE SOCIAL MEDIA. *Journal of Engineering Sciences*, 13(11).
- [7] Sardari, S., Nakisa, B., Rastgoo, M. N., & Eklund, P. (2022). Audio based depression detection using Convolutional Autoencoder. *Expert Systems with Applications*, 189, 116076.

- [8] Khaparde, A. A., Das, R., & Bhargava, R. (2023, January). Transformer Based Approach for Depression Detection. In 2023 15th International Conference on Developments in eSystems Engineering (DeSE) (pp. 369-374). IEEE.
- [9] Arora, S., Malik, A., Shabaz, M., & Asenso, E. Machine Learning based Model for Detecting Depression During Covid-19 Crisis. Scientific African.
- [10] Sahithi, V., Aishwarya, R., Priya, S. S., & Harsha, S. K. STOCHASTIC GRADIENT BOOSTING MODEL FOR SUICIDAL TWEETS DETECTION IN ONLINE SOCIAL MEDIA.

APPENDIX

A. SOURCE CODE

```
import streamlit as st
import json
import os
import re
import string
# from streamlit_webrtc import webrtc_streamer
from st_audiorec import st_audiorec
# import soundfile as sf
# import speech_recognition as sr
import io
import whisper
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import speech_recognition as sr
from pydub import AudioSegment
import pickle

session_state = st.session_state
if "user_index" not in st.session_state:
    st.session_state["user_index"] = 0

def signup(json_file_path="data.json"):
    st.title("Signup Page")
    with st.form("signup_form"):
        st.write("Fill in the details below to create an account:")
        name = st.text_input("Name:")
        email = st.text_input("Email:")
        age = st.number_input("Age:", min_value=0, max_value=120)
        sex = st.radio("Sex:", ("Male", "Female", "Other"))
        password = st.text_input("Password:", type="password")
        confirm_password = st.text_input("Confirm Password:",
type="password")

        if st.form_submit_button("Signup"):
            if password == confirm_password:
```

```

        user = create_account(name, email, age, sex, password,
json_file_path)
        session_state["logged_in"] = True
        session_state["user_info"] = user
    else:
        st.error("Passwords do not match. Please try again.")

def check_login(username, password, json_file_path="data.json"):
    try:
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)

        for user in data["users"]:
            if user["email"] == username and user["password"] ==
password:
                session_state["logged_in"] = True
                session_state["user_info"] = user
                st.success("Login successful!")
                return user

        return None
    except Exception as e:
        st.error(f"Error checking login: {e}")
        return None

def initialize_database(json_file_path="data.json"):
    try:
        # Check if JSON file exists
        if not os.path.exists(json_file_path):
            # Create an empty JSON structure
            data = {"users": []}
            with open(json_file_path, "w") as json_file:
                json.dump(data, json_file)
    except Exception as e:
        print(f"Error initializing database: {e}")

def create_account(name, email, age, sex, password,
json_file_path="data.json"):
    try:
        # Check if the JSON file exists or is empty
        if not os.path.exists(json_file_path) or
os.stat(json_file_path).st_size == 0:
            data = {"users": []}
        else:
            with open(json_file_path, "r") as json_file:
                data = json.load(json_file)

```

```

        # Append new user data to the JSON structure
        user_info = {
            "name": name,
            "email": email,
            "age": age,
            "sex": sex,
            "password": password,
        }
        data["users"].append(user_info)

    # Save the updated data to JSON
    with open(json_file_path, "w") as json_file:
        json.dump(data, json_file, indent=4)

    st.success("Account created successfully! You can now login.")
    return user_info
except json.JSONDecodeError as e:
    st.error(f"Error decoding JSON: {e}")
    return None
except Exception as e:
    st.error(f"Error creating account: {e}")
    return None
import streamlit as st

def process_input_and_provide_resources(user_input):
    with open('logistic.pkl', 'rb') as f:
        model1 = pickle.load(f)
    with open('tfidf_vectorizer.pkl', 'rb') as f:
        vectorizer = pickle.load(f)

    text = preprocess(user_input)
    tfidf_vector = vectorizer.transform([text])
    pred_1 = model1.predict(tfidf_vector)
    first_prediction = pred_1[0]

    st.write('**Tendencies:**', first_prediction)

    st.write("**Resources:**")

    if first_prediction == 'depression':
        st.write("Here are some resources to help cope with depression:")

```

```

        st.markdown("- [How to cope up with  
depression](https://www.youtube.com/watch?v=8Su5VtKeXU8)")
        st.markdown("- [Ways to cope with  
depression](https://www.medicalnewstoday.com/articles/327018)")
        st.markdown("- [Depression  
Treatment](https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/depression-treatment-and-management)")
        st.markdown("- [Don't suffer from Depression in  
Silence](https://www.ted.com/talks/nikki_webber_allen_don_t_suffer_from_your_depression_in_silence?hasSummary=true&language=en)")

    elif first_prediction == 'suicide':
        st.write("Some resources for controlling suicidal tendencies:")
        st.markdown("- [Are you feeling  
Suicidal?](https://www.helpguide.org/articles/suicide-prevention/are-you-feeling-suicidal.htm)")
        st.markdown("- [Suicide and Suicidal  
Thoughts](https://www.mayoclinic.org/diseases-conditions/suicide/symptoms-causes/syc-20378048)")
        st.markdown("- [Suicide  
Prevention](https://www.ted.com/talks/ashleigh_husbands_suicide_prevention?hasSummary=true)")
        st.markdown("- [National Suicide Prevention  
Lifeline](https://suicidepreventionlifeline.org/)")
        st.markdown("- [Crisis Text  
Line](https://www.crisistextline.org/)")
        st.markdown("- [International Association for Suicide  
Prevention](https://www.iasp.info/resources/Crisis_Centres/)")

    else:
        st.write("Great job! Keep up the positivity and stay clear of  
negative thinking.")

def login(json_file_path="data.json"):
    st.title("Login Page")
    username = st.text_input("Email:")
    password = st.text_input("Password:", type="password")

    login_button = st.button("Login")

    if login_button:
        user = check_login(username, password, json_file_path)
        if user is not None:
            session_state["logged_in"] = True

```

```

        session_state["user_info"] = user
    else:
        st.error("Invalid credentials. Please try again.")

def get_user_info(email, json_file_path="data.json"):
    try:
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)
            for user in data["users"]:
                if user["email"] == email:
                    return user
        return None
    except Exception as e:
        st.error(f"Error getting user information: {e}")
        return None

def render_dashboard(user_info, json_file_path="data.json"):
    try:
        st.title(f"Welcome to the Dashboard, {user_info['name']}!")
        st.subheader("User Information:")
        st.write(f"Name: {user_info['name']}")
        st.write(f"Sex: {user_info['sex']}")
        st.write(f"Age: {user_info['age']}")

    except Exception as e:
        st.error(f"Error rendering dashboard: {e}")

def preprocess(text):
    def remove_url(text):
        return re.sub(r"http\S+", "", text)
    exclude = string.punctuation
    def remove_punctuation(text):
        return text.translate(str.maketrans("", "", exclude))

    def remove_stopwords(text):
        stopword = stopwords.words('english')
        new_text = []
        for word in text.split():
            if word in stopword:
                new_text.append('')
            else:
                new_text.append(word)
    x = new_text[:]
    new_text.clear()

```

```

        return " ".join(x)
    def lemmatize_text(text):
        words = word_tokenize(text)
        lemmatizer = WordNetLemmatizer()
        lemmatized_words = [lemmatizer.lemmatize(word) for word in
words]
        return " ".join(lemmatized_words)
    text=remove_url(text)
    text=remove_punctuation(text)
    text=remove_stopwords(text)
    text=lemmatize_text(text)
    return text

def transcribe_audio_from_data(file_data):
    with open("temp.mp3", "wb") as f:
        f.write(file_data)
    model = whisper.load_model("base")
    result = model.transcribe("temp.mp3")
    os.remove("temp.mp3")
    return result['text']

def main(json_file_path="data.json"):
    st.sidebar.title("Tendency Text")
    page = st.sidebar.radio(
        "Go to",
        ("Signup/Login", "Dashboard", "Text Classification"),
        key="comment_classifying",
    )

    if page == "Signup/Login":
        st.title("Signup/Login Page")
        login_or_signup = st.radio(
            "Select an option", ("Login", "Signup"),
key="login_signup"
        )
        if login_or_signup == "Login":
            login(json_file_path)
        else:
            signup(json_file_path)

    elif page == "Dashboard":
        if session_state.get("logged_in"):
            render_dashboard(session_state["user_info"])
        else:
            st.warning("Please login/signup to view the dashboard.")

```

```

elif page == "Text Classification":
    if session_state.get("logged_in"):
        st.title("Text Classification")

        # Upload audio file
        selection = st.radio(
            "Choose an option:",
            ("Audio", "Text")
        )
        if selection == "Text":
            user_input = st.text_area("Enter your Comment here:",
height=200)
            if st.button('Submit'):
                process_input_and_provide_resources(user_input)
            else:
                options = ["Record", "Upload"]
                choice = st.radio("Choose an option", options)
                # Check the choice and display the corresponding
widget
                if choice == "Record":
                    st.write("Click the button below to start
recording:")

                    audio = st_audiorec()
                    if audio is not None and st.button('Submit'):
                        user_input = transcribe_audio_from_data(audio)
                        process_input_and_provide_resources(user_input)

                elif choice == "Upload":
                    audio = st.file_uploader("Upload an audio file",
type=["mp3", "wav", "ogg"])
                    if audio is not None and st.button('Submit'):
                        st.audio(audio, format="audio/wav")
                        user_input=
transcribe_audio_from_data(audio.read())
                        process_input_and_provide_resources(user_input
)

            else:
                st.warning("Please login/signup to use the app")

if __name__ == "__main__":
    initialize_database()
    main()

```

B. SCREENSHOTS

```
import pandas as pd
import re
import string
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, recall_score, precision_score
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.naive_bayes import MultinomialNB
import pickle
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import StackingClassifier
from xgboost import XGBClassifier
```

```
def lemmatize_text(text):
    words = word_tokenize(text)
    lemmatizer = WordNetLemmatizer()
    lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
    return " ".join(lemmatized_words)
```

```
def preprocess(train_data):
    train_data['text'] = train_data['text'].str.lower()
    train_data['text'] = train_data['text'].apply(remove_url)#remove url
    train_data['text'] = train_data['text'].apply(remove_punctuation)# remove punctuation
    train_data['text'] = train_data['text'].apply(remove_stopwords)
    train_data['text'] = train_data['text'].apply(lemmatize_text)
```

```
preprocess(X_train)
```

```
preprocess(X_test)
```

```

random_forest = RandomForestClassifier( random_state=42 , n_estimators=1000)

# Fit the model on the training data
random_forest.fit(tfidf_matrix, y_train)

y_pred_custom = random_forest.predict(test_tf)

# Calculate accuracy and print classification report
accuracy = accuracy_score(y_test, y_pred_custom)
classification_r = classification_report(y_test, y_pred_custom)
print("Validation Accuracy:", accuracy)
print("Classification Report:\n", classification_r)

cm = confusion_matrix(y_test, y_pred)
# plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=y_lab,
            yticklabels=y_lab)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()

filename = 'random_forest.pkl'
pickle.dump(random_forest, open(filename, 'wb'))

```

Accuracy: 0.7537938252223967

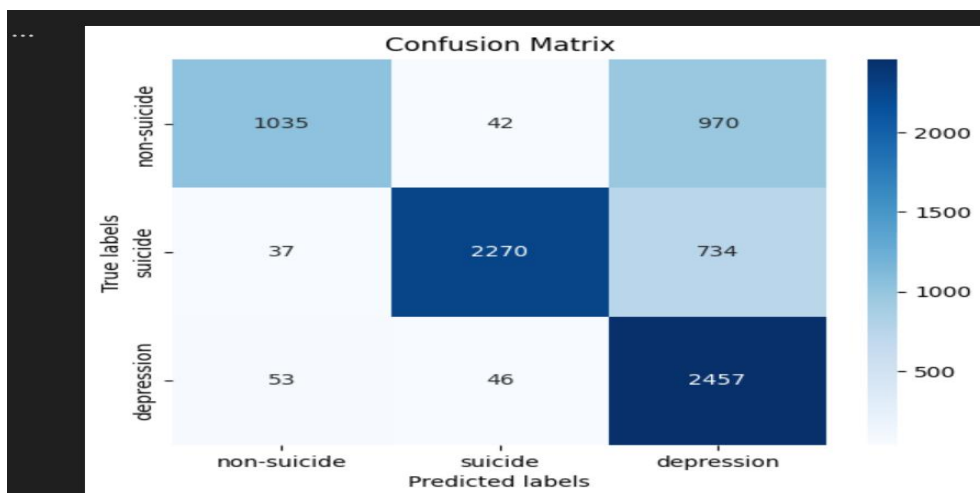
```

Classification Report:
              precision    recall  f1-score   support

 depression           0.92      0.51      0.65       2047
 non-suicide          0.96      0.75      0.84       3041
 suicide              0.59      0.96      0.73       2556

 accuracy              0.75              0.75       7644
 macro avg             0.82      0.74      0.74       7644
 weighted avg          0.83      0.75      0.75       7644

```



```

decision_tree = DecisionTreeClassifier( random_state=42)

# Fit the model on the training data
decision_tree.fit(tfidf_matrix, y_train)

y_pred_custom = decision_tree.predict(test_tf)

# Calculate accuracy and print classification report
accuracy = accuracy_score(y_test, y_pred_custom)
classification_r = classification_report(y_test, y_pred_custom)
print("Validation Accuracy:", accuracy)
print("Classification Report:\n", classification_r)

cm = confusion_matrix(y_test, y_pred)
# plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=y_lab,
            yticklabels=y_lab)
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')
plt.show()

filename = 'decision_tree.pkl'
pickle.dump(decision_tree, open(filename, 'wb'))

```

```

Validation Accuracy: 0.771978021978022
Classification Report:
              precision    recall  f1-score   support

 depression         0.78      0.79      0.79        2047
 non-suicide         0.80      0.80      0.80        3041
      suicide         0.73      0.72      0.72        2556

 accuracy              0.77              0.77              0.77        7644
 macro avg              0.77              0.77              0.77        7644
 weighted avg           0.77              0.77              0.77        7644

```

