MENTAL HEALTH PREDICTION USING MACHINE LEARNING

MINI PROJECT REPORT

Submitted by

ARTHY M

REENA S

SHREE HARINI S

SOUNDARYA AL

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND BUSINESS SYSTEMS

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

SAMAYAPURAM, TRICHY

ANNA UNIVERSITY

CHENNAI 600 025

DECEMBER 2024

MENTAL HEALTH PREDICTION USING MACHINE LEARNING

UCB1512 MINI PROJECT REPORT

Submitted by

ARTHY M (8115U22CB007)

REENA S (8115U22CB044)

SHREE HARINI S (8115U22CB050)

SOUNDARYA AL (8115U22CB052)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND BUSINESS SYSTEMS

Under the Guidance of

Mrs. M.R.NITHYA, M.E.,

Department of Computer Science and Business Systems

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

COMPUTER SCIENCE AND BUSINESS SYSTEMS

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certificate that this mini project report **"MENTAL HEALTH PREDICTION USING MACHINE LEARNING"** is the bonafide work of **"ARTHY M (8115U22CB007), REENA S (8115U22CB044), SHREE HARINI S (8115U22CB050), SOUNDARYA AL (8115U22CB052)"** who carried out the mini project work under my supervision.

Dr. J. SASIDEVI, M.E., Ph.D.,	**Mrs. M.R.NITHYA, M.E.,**
Associate Professor	Assistant Professor
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Computer Science and Business Systems	Department of Computer Science and Business Systems
K.Ramakrishnan College of Engineering, (Autonomous)	K.Ramakrishnan College of Engineering, (Autonomous)
Samayapuram, Trichy-621112.	Samayapuram, Trichy-621112.

SIGNATURE OF INTERNAL EXAMINER	**SIGNATURE OF EXTERNAL EXAMINER**
NAME:	**NAME:**
DATE:	**DATE:**

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

ACKNOWLEDGMENT

We thank the almighty GOD, without whom it would not have been possible for us to complete our mini project.

We wish to address our profound gratitude to **Dr. K. RAMAKRISHNAN,** Chairman, K.Ramakrishnan College of Engineering (Autonomous), who encouraged and gave us all help throughout the course.

We express our hearty gratitude and thanks to our honorable and grateful Executive Director **Dr. S. KUPPUSAMY, B.Sc., MBA., Ph.D.,** K.Ramakrishnan College of Engineering (Autonomous).

We are glad to thank our Principal **Dr.D. SRINIVASAN, M.E., Ph.D., FIE., MIIW., MISTE., MIS, AE., C.Eng.,** K.Ramakrishnan College of Engineering (Autonomous), for giving us permission to carry out this project.

We wish to convey our sincere thanks to **Dr. J. SASIDEVI, M.E.,Ph.D.,** Head of the Department of Computer Science and Business Systems, K. Ramakrishnan College of Engineering (Autonomous), for giving us constant encouragement and advice throughout the course.

We are grateful to **Mrs. M.R.NITHYA, M.E.,** Assistant Professor in Computer Science and Business Systems, K. Ramakrishnan College of Engineering (Autonomous), for her guidance and valuable suggestions during the course of study.

Finally, we sincerely acknowledge in no less terms all our staff members, colleagues, parents and friends for their cooperation and help at various stages of this project work.

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

DECLARATION BY THE CANDIDATE

I declare that to the best of my knowledge the work reported here in has been composed solely by myself and that it has not been in whole or in part in any previous application for a degree.

Submitted for the mini project Viva-Voce held at K.Ramakrishnan College of Engineering on _________

SIGNATURE OF THE CANDIDATE

(ARTHY M)

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

DECLARATION BY THE CANDIDATE

I declare that to the best of my knowledge the work reported here in has been composed solely by myself and that it has not been in whole or in part in any previous application for a degree.

Submitted for the mini project Viva-Voce held at K.Ramakrishnan College of Engineering on _________

SIGNATURE OF THE CANDIDATE

(REENA S)

.

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

DECLARATION BY THE CANDIDATE

I declare that to the best of my knowledge the work reported here in has been composed solely by myself and that it has not been in whole or in part in any previous application for a degree.

Submitted for the mini project Viva-Voce held at K.Ramakrishnan College of Engineering on _________

SIGNATURE OF THE CANDIDATE

(SHREE HARINI S)

K.RAMAKRISHNAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Under

ANNA UNIVERSITY, CHENNAI

DECLARATION BY THE CANDIDATE

I declare that to the best of my knowledge the work reported here in has been composed solely by myself and that it has not been in whole or in part in any previous application for a degree.

Submitted for the mini project Viva-Voce held at K.Ramakrishnan College of Engineering on _________

SIGNATURE OF THE CANDIDATE

(SOUNDARYA AL)

ABSTRACT

The early detection of mental health issues enables specialists to provide more effective treatment, significantly enhancing student quality of life. Mental health encompasses psychological, emotional, and social well-being, profoundly affecting how individuals think, feel, and behave. It is vital at every life stage, from childhood and adolescence through adulthood, influencing overall functioning and well-being. This project focused on two machine learning techniques—decision tree classifiers and random forest algorithms—and evaluated their accuracy in detecting mental health problems. By employing various accuracy metrics, we aimed to identify the most effective method for early detection, ultimately supporting targeted interventions and improving mental health outcomes across diverse populations.

The prediction of mental health issues using machine learning (ML) involves applying advanced algorithms to analyze patterns and predict the likelihood of mental health conditions. This approach leverages large datasets to identify key indicators and correlations that may not be easily recognizable through traditional methods. These models split data into branches based on specific criteria, forming a tree-like structure. At each node, the model decides the best feature to split the data on, aiming to maximize the distinction between classes Decision trees are intuitive and simple to understand but can overfit on complex datasets, reducing accuracy. Random forests are an ensemble method that builds multiple decision trees and aggregates their predictions. Random forests are well-suited for handling large, noisy datasets and capturing subtle patterns in the data. The ultimate goal was to support targeted interventions and improve mental health outcomes across diverse populations, addressing the global need for scalable and precise mental health solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
ML	Machine Learning
DT	Decision Tree
RF	Random Forest
SVM	Support Vector Machines
BD	Bipolar Disorder
MDD	Major Depressive Disorder
IDE	Interactive Development Environment
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
XG	Extreme Gradient

CHAPTER 1
INTRODUCTION

Mental well-being is an important part of a person's overall health. It includes how they feel mentally and the environment around them. Problems with brain chemistry can lead to mental illnesses, which affect how people think and behave. A person's mental health can help us understand how well they can deal with different health issues. To spot potential problems early, it's important to track the mental health of different groups, like working professionals, college students, and high school students. By understanding mental health in these groups, we can identify risks and create better support and treatments to improve mental health in the community.

1.1 PROBLEM STATEMENT

Mental health issues often begin in childhood and can persist into adulthood, highlighting the importance of early intervention, focuses on using Decision Tree classifiers and Random Forest algorithms to predict mental health challenges across students. By examining these algorithms, we aim to analyse their effectiveness in identifying mental health issues such as depression, panic attack, anxiety, etc.

We will collect data from existing mental health datasets that include students records. After pre-processing the data to handle missing values and ensure consistency, we will identify key features influencing mental health outcomes. We will implement Decision Tree classifiers for their interpretability and Random Forest algorithms for their accuracy and robustness. Model evaluation will involve using metrics such as accuracy, precision, recall, and F1-score, along with cross-validation to ensure reliability. We will analyse the results to identify critical predictors of mental health issues and utilize techniques like feature importance to interpret the findings. This aims to enhance early detection and treatment strategies in mental health care, ultimately fostering better outcomes across the lifespan.

1.2 PROJECT OBJECTIVE

This project aims to develop an accurate, machine learning-based model to predict mental health conditions, such as anxiety, depression, and panic attack, among students. Using student mental health data, this focuses on using machine learning algorithms, specifically Decision Tree and Random Forest classifiers, to identify individuals in need of mental health support.

Specific Objectives

1. Data Analysis and Feature Selection

Analyse student mental health data to identify key factors related to mental health disorders. Select the most relevant features that impact conditions like anxiety, depression, and panic attack.

2. Model Development

Develop and train a machine learning model using Decision Tree Classifier (DT) and Random Forest algorithms (RF). Focus on creating a model that can accurately classify individuals based on mental health risk factors identified in the training data.

3. Performance Evaluation

Evaluate the model's accuracy, precision, and recall to measure predictive effectiveness. Compare the new model’s performance with existing approaches to ensure improvement in reliability and accuracy.

4. Visualization and Reporting

Visualize the feature importance for both Decision Tree and Random Forest models. Provide comparative performance charts for the two algorithms to justify the final model selection.

1.3 AIM AND OBJECTIVE

Develop a predictive system that leverages machine learning algorithms, specifically Random Forest and Decision Tree, to accurately assess and predict mental health conditions based on student data of mental issues such as depression, anxiety, and panic attack.

Objectives

1. Data Collection and Pre-processing

Gather and pre-process diverse data sources, including questionnaires, clinical assessments, and behavioural data, while ensuring data privacy and ethical handling Clean, normalize, and balance data to improve model robustness and reduce bias in predictions.

2. Feature Engineering

Identify relevant features such as mental health history, and environmental triggers. Engineer additional features as needed to capture complex relationships, trends, and correlations relevant to mental health predictions.

3. Model Training and Optimization

Train Random Forest and Decision Tree algorithms on labeled datasets to classify or predict mental health conditions, ensuring high accuracy and generalizability. Optimize hyper parameters using cross-validation to balance accuracy, precision, and recall across mental health categories.

4. System Evaluation and Comparison

Compare the performance of Random Forest and Decision Tree algorithms on various metrics, including accuracy, F1 score Evaluate model performance on datasets to assess reliability and applicability.

5. System Improvement and Adaptation :

Continuously refine the model based on user feedback and identified limitations. Adapt the system to accommodate new data types or improve predictive accuracy, thereby enhancing its applicability in mental health diagnostics and support.

1.4 SCOPE OF THE PROJECT

Aims to address the pressing need for accurate and accessible mental health prediction, providing a tool that supports early detection of conditions like anxiety, depression, and panic attack among students. By harnessing the power of machine learning algorithms, specifically Decision Tree and Random Forest classifiers, we seek to create a reliable predictive model that can be readily integrated into healthcare settings.

Our focus isn’t on diagnosing every mental health disorder but rather on identifying prevalent conditions that impact daily life. This model will offer predictions based on key mental health indicators and aim to support healthcare professionals in assessing and managing mental health more effectively.

Initially, the model will translate these predictions into actionable insights, highlighting the need for potential follow-up or further intervention.

Envision this system as a bridge to timely support, providing clear, easy-to-interpret results that can guide conversations between healthcare providers and patients. The model will prioritize low latency to deliver fast predictions, helping professionals make decisions more efficiently in clinical environments. We will also explore user-friendly features, such as customizable reporting and visual outputs, to suit diverse healthcare needs.

Achieving accuracy and speed in real-world conditions will be challenging. This project must overcome factors like noise in data, varied patient backgrounds, and the complex nuances of mental health. Using adaptive machine learning models, we will address these challenges to ensure high performance across different patient profiles and environments.

Our model's evaluation will extend beyond lab settings to real-world applications, involving collaborations with mental health clinics and feedback from healthcare professionals. This feedback will help us refine the model, adapt it to real-life demands, and prioritize improvements.

Ultimately, this project isn't just about developing an algorithm; it's about empowering healthcare providers with a powerful tool to better support their patients and fostering an inclusive approach to mental health care.

1.5 FLOWCHART

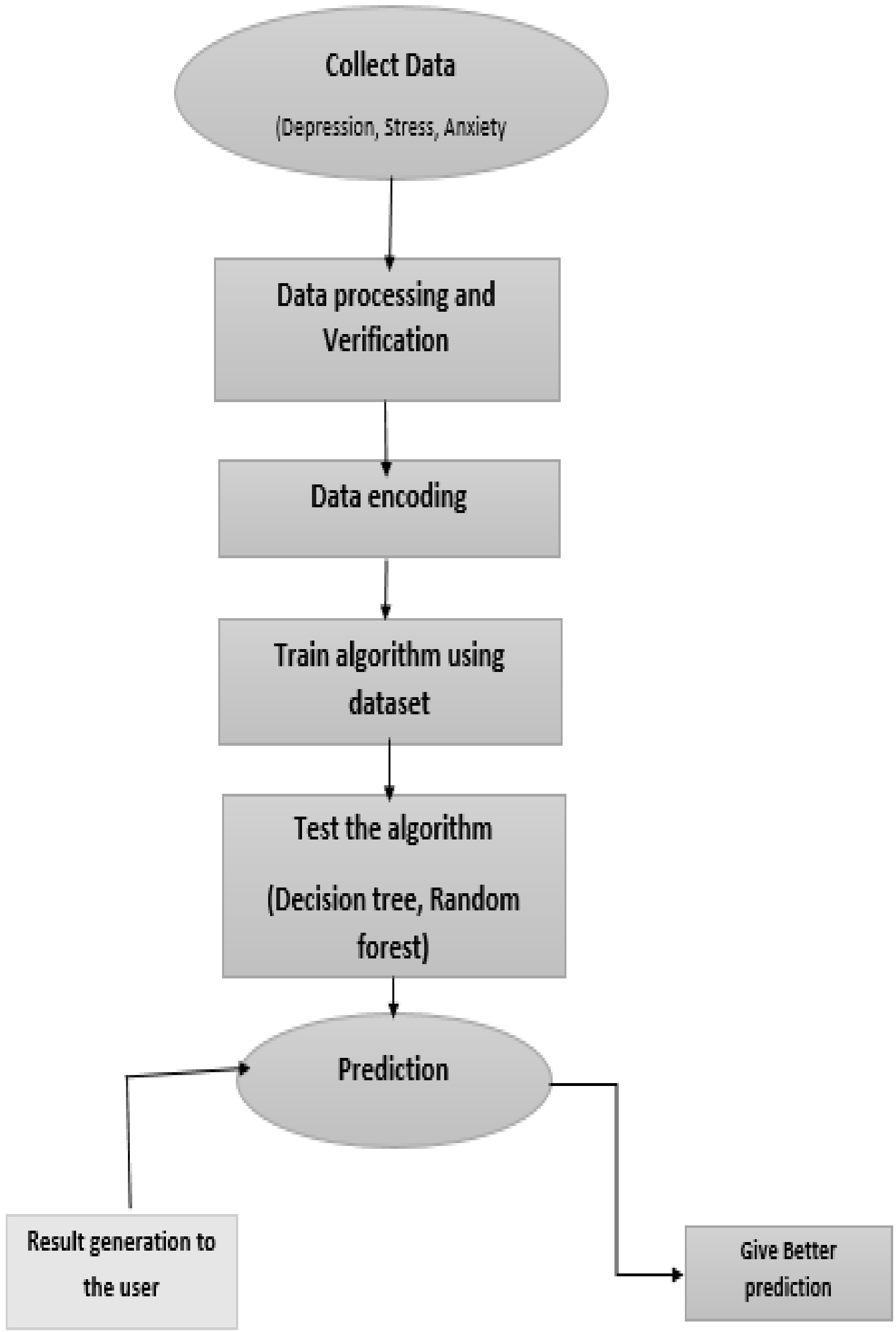

Fig 1.5. Flow chart

CHAPTER 2

LITERATURE SURVEY

2.1 JUDGING MENTAL HEALTH DISORDERS USING THE DECISION TREE MODELS

AUTHOR: Sandip Roy , P.S.Aithal , Rajesh Bose

YEAR OF PUBLICATION: 2017

"Judging Mental Health Disorders Using the Decision Tree Models" explores the application of Decision Tree models in diagnosing mental health disorders. This research emphasizes the growing need for automated systems in mental health care due to increasing cases and limited mental health professionals. The authors demonstrate the effectiveness of Decision Trees in classifying and predicting mental health conditions based on various factors such as symptoms, behavioral patterns, and demographic data.

2.2 PREDICTING MENTAL HEALTH ILLNESS USING MACHINE LEARNING

AUTHOR : Konda Vaishnavi , U Nikhitha Kamath , B Ashwath Rao and N V Subba Reddy

YEAR OF PUBLICATION: 2021

The paper "Predicting Mental Health Illness Using Machine Learning" focuses on leveraging machine learning techniques to predict mental health disorders. The authors underline the rising global concern of mental health issues and the necessity of early diagnosis to prevent severe consequences. Including Support Vector Machines (SVM), Decision Trees, and Random Forests, to analyze patient data and predict mental health conditions effectively.

2.3 PREDICTION OF MENTAL HEALTH PROBLEMS AMONG CHILDREN USING MACHINE LEARNING TECHNIQUES

AUTHOR : Ms .Sumathi And Dr .B. Poorna

YEAR OF PUBLICATION: 2016

The paper ″Prediction of Mental Health Problems Among Children Using Machine Learning Techniques″ explores the application of machine learning in identifying and predicting mental health issues among children. It highlights the increasing prevalence of mental health challenges in children and the limitations of traditional diagnostic methods in addressing these issues effectively. underscores the importance of early detection of mental health problems to enable timely intervention. The authors conclude that integrating machine learning models with educational and healthcare systems can provide a cost-effective and efficient tool for monitoring and managing children's mental health.

2.4 A MACHINE LEARNING ALGORITHM TO DIFFERENTIATE BIPOLAR DISORDER FROM MAJOR DEPRESSIVE DISORDER USING AN ONLINE MENTAL HEALTH QUESTIONNAIRE AND BLOOD BIOMARKER DATA

AUTHOR : Jakub Tomasik, Sung Yeon Sarah Han, Jason D. Cooper.,

YEAR OF PUBLICATION: 2021

"A Machine Learning Algorithm to Differentiate Bipolar Disorder from Major Depressive Disorder Using an Online Mental Health Questionnaire and Blood Biomarker Data" presents a novel approach to diagnosing complex mental health conditions. The study focuses on distinguishing between Bipolar Disorder (BD) and Major Depressive Disorder (MDD), which often exhibit overlapping symptoms, making traditional diagnostic methods challenging.

The authors utilize a combination of machine learning algorithms, mental health questionnaires, and blood biomarker data to enhance diagnostic accuracy. The integration of biomarkers provides a biological basis to complement subjective questionnaire responses, leading to more precise differentiation. Algorithms such as Random Forest and Support Vector Machines (SVM) are employed, showcasing their ability to handle multidimensional data and identify subtle differences between BD and MDD. The study highlights the importance of combining clinical assessments with advanced computational techniques to improve diagnostic outcomes. The authors conclude that such hybrid models could revolutionize mental health diagnostics, enabling personalized treatment plans and reducing misdiagnoses in psychiatric care.

2.5 APPLICATION OF MACHINE LEARNING METHODS IN MENTAL HEALTH DETEDTION

AUTHOR : ROHIZAH ABD RAHMAN,KHAIRUDDIN OMAR.,

YEAR OF PUBLICATION: 2020

"Application of Machine Learning Methods in Mental Health Detection" explores the potential of machine learning techniques in diagnosing and predicting mental health conditions. The study addresses the increasing mental health challenges globally and the limitations of traditional diagnostic tools, such as subjectivity and time constraints. The authors recommend the integration of these machine learning-based tools into mental health care systems to facilitate early diagnosis and intervention. This approach ensures scalability, accessibility, and cost-effectiveness in managing mental health challenges.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The current landscape of mental health prediction systems demonstrates the growing integration of technology and healthcare to improve early diagnosis and treatment outcomes. Existing systems employ various methodologies to predict mental health conditions, ranging from traditional statistical models to advanced machine learning approaches.

3.1.1 Traditional Approaches

Traditional mental health assessment methods often rely on standardized questionnaires, clinical interviews, and manual evaluations conducted by healthcare professionals. These methods, while effective in some cases, are time-consuming and subjective, as they depend heavily on the experience and judgment of the practitioner. Additionally, these approaches may not always capture the nuanced patterns and correlations in patient data, leading to potential delays in diagnosis.

3.1.2 Machine Learning-Based Systems

With the advancement of technology, there has been a significant shift towards machine learning-based systems for mental health prediction. These systems leverage large datasets and powerful algorithms to identify patterns and trends in patient behaviour, clinical history, and other relevant factors.

3.2 PROPOSED SYSTEM

The proposed method is to build a ML model to predict mental health (depression).The dataset is first pre-processed and the columns are analyzed and then different machine learning algorithms would be compared to obtain the predictive model with maximum accuracy.

Data is loaded, checked for cleanliness, and then trimmed and cleaned for analysis.The data set collected for predicting given data is split into Training set and Test set. The Data Model which was created using machine learning algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

ML algorithms prediction model is effective because of the following reasons: It provides better results in classification problem. It is strong in pre-processing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables. It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

These reports are to the investigation of applicability of machine learning techniques for Mental Health prediction in operational conditions. Finally, it highlights some observations on future research issues, challenges, and needs.

The system will employ machine learning algorithms, particularly:

Decision Tree: For interpretable and straightforward classification.

Random Forest: To improve prediction accuracy by aggregating the outputs of multiple decision trees.
This proposed system seeks to bridge the gap between traditional mental health assessment methods and the growing need for efficient, data-driven solutions in mental healthcare.

3.3 METHODOLOGY

It employs a systematic approach to examine the effectiveness of Decision Tree and Random Forest algorithms in predicting mental health conditions like anxiety, depression, and panic attack. The methodology involves several stages: data collection and pre-processing, feature extraction, model training, and performance evaluation.

The initial stage focuses on data collection from mental health records, and publicly available datasets relevant to mental health disorders. The data is curated to include a diverse set of variables, such as students details. Data pre-processing steps involve cleaning missing values, normalizing numerical values, and encoding categorical data to prepare for algorithm training. The datasets are split into training and testing sets to enable reliable model validation.

For feature extraction and selection, the Random Forest's built-in feature importance function is utilized to identify the most predictive variables. Factors with high relevance to mental health, such as age, history of mental illness, and environmental stress levels, are prioritized.

In the model training phase, Decision Tree and Random Forest algorithms are implemented. The Decision Tree classifier builds a simple, interpretable model by splitting data based on feature values at each node, producing a clear classification path. The Random Forest model is trained as an ensemble of multiple decision trees, which reduces individual tree biases and enhances predictive accuracy. Evaluation metrics—including accuracy, precision, recall, and F1-score—are employed to measure model effectiveness in mental health prediction. Cross-validation techniques are used to validate the models on various data splits, ensuring robustness and generalizability across different subsets of the dataset. Additionally, confusion matrices are analysed to understand classification errors, which are particularly critical for accurately identifying individuals at high risk for mental health issues.

Finally, the methodology includes a comparative analysis of Decision Tree and Random Forest models against baseline models used in prior research. This assessment allows for a comprehensive evaluation of the models' strengths and limitations in predicting mental health conditions, providing insights for potential integration into clinical practice.

3.4 SYSTEM ARCHITECTURE

We propose a recognition system architecture taking into account the functioning of mental health prediction using machine learning algorithms.

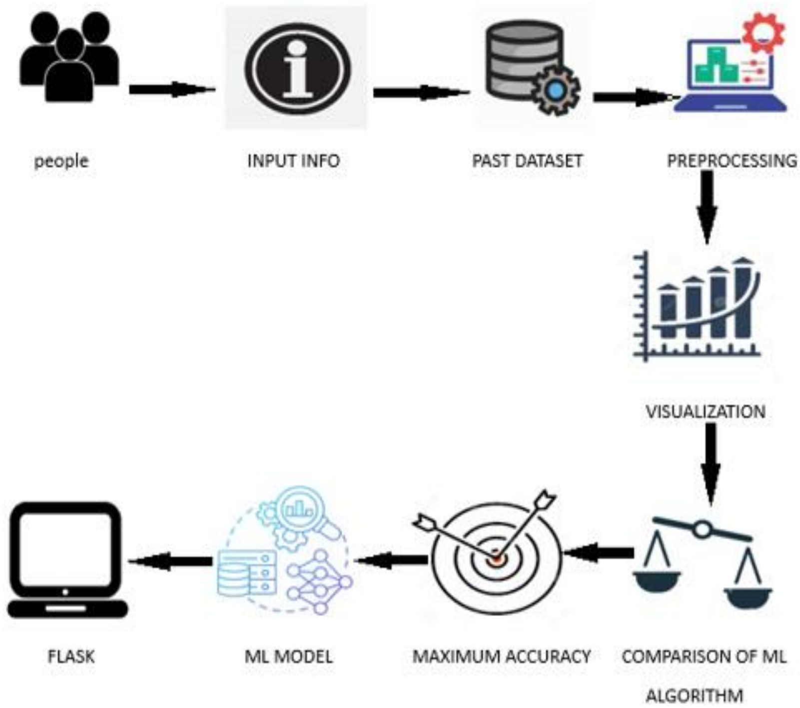

Fig 3.4. System Architecture diagram

CHAPTER 4
MODULE IMPLEMENTATION

4.1 MODULE LIST

- **Training and Validation**
- **Testing**

4.2 MODULE IMPLEMENTATION

4.2.1 Training and Validation

In this module, the dataset is split into training and validation sets to build and fine-tune the models. The Decision Tree and Random Forest algorithms are trained on the training set to classify mental health conditions based on input features such as demographic details, behavioral patterns, and responses to mental health questionnaires. During training, hyperparameters like tree depth and the number of estimators in the Random Forest are optimized to enhance the model’s accuracy and reduce overfitting.

Validation is performed by testing the models on a separate validation set to measure their generalization performance. Metrics such as accuracy, precision, recall, and F1-score are calculated to evaluate the models. Cross-validation techniques like k-fold cross-validation are used to ensure the stability and reliability of the models across multiple splits of the dataset. The final trained models are saved for use in the testing phase.

4.2.2 Testing

The testing module uses the models trained in the previous module to evaluate their performance on a separate test dataset, which contains data not seen during training. The models classify mental health conditions based on the input features and generate predictions for each test instance. Performance metrics such as accuracy and confusion matrix are calculated to assess the effectiveness of the models.

The predictions are further processed to display the likelihood of mental health issues in percentage form, providing users with clear and interpretable results. The best-performing model, based on testing metrics, is finalized for deployment in the web interface, ensuring reliable real-time predictions for users.

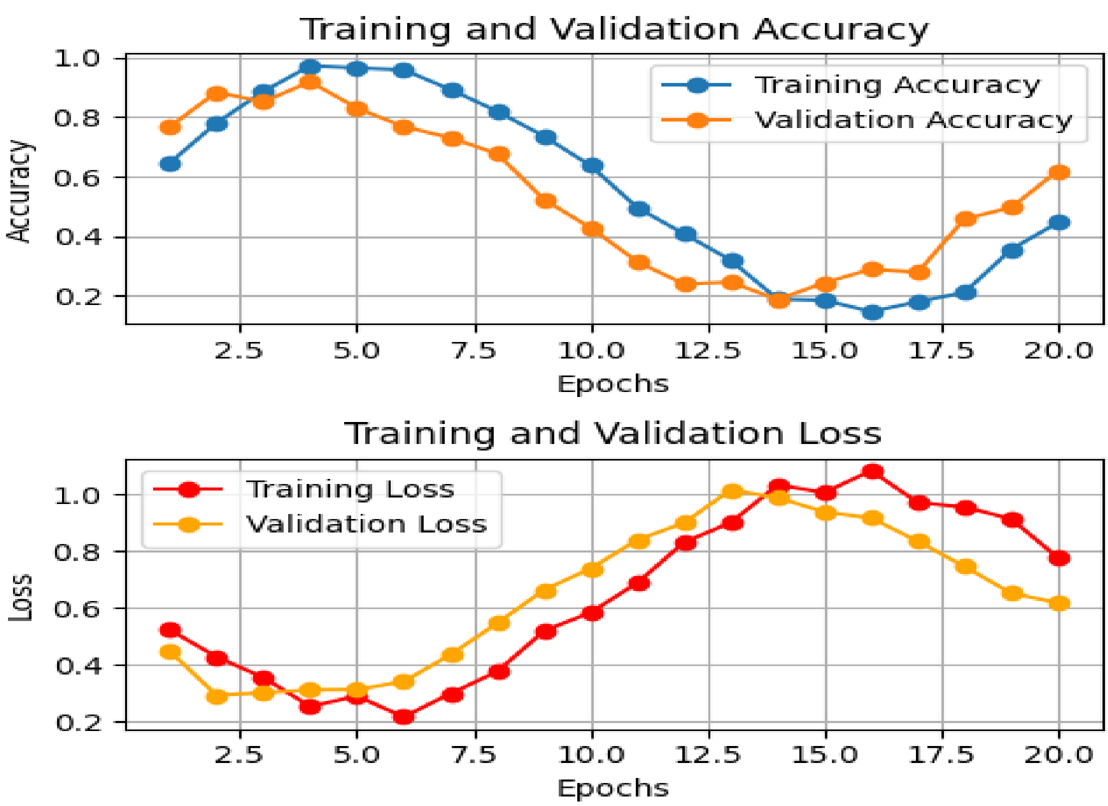

Fig 4.2.2.Training Model Chart

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE SYSTEM REQUIREMENTS

- **Processor** **:** Intel Core i3
- **RAM** **:** 4 GB
- **Hard Disk** **:** 10 GB
- **Compact Disk** **:** 650 MB
- **Keyboard** **:** Standard keyboard
- **Monitor** **:** 15 inch color monitor
- **Internet** **:** Required for Google Co-lab access

5.2 SOFTWARE SYSTEM REQUIREMENTS

- **Operating System :** Platform-independent (accessible via any OS with internet, e.g., Windows, macOS, or Linux)
- **Development Environment:** Google Co-lab (cloud-based Jupyter Notebook)
- **Programming Language:** Python
- **Libraries and Frameworks:** NumPy, Pandas , Matplotlib, Seaborn

5.3 SOFTWARE ENVIRONMENT

5.3.1 SOFTWARE USED

Python is a high-level, versatile, and widely used programming language, particularly in the field of data science and machine learning. It is known for its simplicity and readability, making it an ideal choice for implementing machine learning projects, including your Mental Health Prediction Using ML Algorithms (Decision Tree and Random Forest*)* project.

5.3.2 PYTHON TECHNOLOGY

1. **Rich Libraries and Frameworks:** Python has a vast ecosystem of libraries that facilitate data analysis, machine learning, and visualization, which are essential for this project. For example:
    a. **Pandas**: Used for data manipulation and preprocessing, handling datasets efficiently.
    b. **NumPy**: Provides support for large, multi-dimensional arrays and matrices, which are often used in machine learning algorithms.
    c. **Matplotlib and Seaborn**: Used for data visualization, helping you create plots and charts to understand and communicate results effectively.
2. **Ease of Learning and Use:** Python's simple and clean syntax makes it easy for developers to learn and use. Even for complex algorithms like Decision Trees and Random Forests, Python offers straightforward implementations and excellent documentation.
3. **Extensive Support for Machine Learning:** Python is one of the most popular languages in the machine learning community. It supports a wide range of machine learning frameworks and algorithms, making it ideal for implementing the Decision Tree and Random Forest models used in this project.

4. **Integration with Google Co-lab:** Python seamlessly integrates with cloud-based platforms like Google Co-lab, which allows you to run code without worrying about local hardware or software configurations. Google Colab provides a free, easy-to-use environment that supports Python and includes pre-installed libraries for machine learning and data analysis.
5. **Community and Resources:** Python has a large community of developers and researchers, which means a wealth of resources, tutorials, and forums are available to help solve any issues you encounter. This makes it easier to find support during development.

How Python is Used in the Project

In your Mental Health Prediction project, Python is used to:

- Load and preprocess datasets using **Pandas**.
- Apply machine learning algorithms (Decision Tree and Random Forest) using **Scikit-learn**.
- Train models on the dataset and evaluate their performance.
- Visualize the results and insights using **Matplotlib** and **Seaborn**.
- Save and load models for further use or deployment with tools like

.

5.3.3 GOOGLE CO-LAB

Google Colab (short for Colaboratory) is a cloud-based interactive development environment (IDE) that is primarily used for running Python code. It provides an easy way to work on projects involving data analysis, machine learning, and artificial intelligence. Google Colab offers several features that make it ideal for your machine learning-based project, including Mental Health Prediction Using ML Algorithms (Decision Tree and Random Forest).

Key Features of Google Colab

1. **Cloud-Based Environment:** Google Colab is hosted in the cloud, meaning you don't need to install any software or worry about local hardware specifications. You can access your project from anywhere, on any device, as long as you have an internet connection.
2. **Pre-installed Libraries and Tools:** Google Colab comes with most of the popular Python libraries pre-installed, such as **Pandas**, **NumPy**, **Matplotlib**, and **Seaborn**, which are essential for your machine learning project. This eliminates the need to install and configure libraries manually, saving time and effort.
3. **Jupyter Notebook Interface:** Google Colab is based on the Jupyter notebook interface, which allows you to write and execute code in a step-by-step manner. You can run Python code blocks individually, visualize data, and display results directly within the notebook. This makes it easier to experiment with different machine learning models, such as Decision Tree and Random Forest, and immediately see the results.

4. **Collaborative Environment:** One of the most significant advantages of Google Colab is its collaborative features. You can share your Colab notebooks with others, allowing them to view or edit your work. This is helpful if you're working on a team or need feedback from mentors or colleagues. It also integrates well with Google Drive for easy file sharing and saving.
5. **Integration with Google Drive:** Google Colab seamlessly integrates with Google Drive, enabling you to save your datasets, models, and notebooks directly in the cloud. This integration also makes it easy to load data into Colab from your Drive and store trained models for future use.
6. **Support for Python Code and Machine Learning:** Since your project is focused on implementing machine learning models (Decision Tree and Random Forest), Colab is an ideal platform. It supports Python's extensive machine learning libraries and offers a smooth experience for implementing, training, and evaluating models.
7. **Easy Visualization:** Google Colab supports rich text formatting, including charts, graphs, and plots generated by libraries such as **Matplotlib** and **Seaborn**. Visualizing results is an essential part of understanding the performance of your models and presenting your findings clearly.
8. **No Setup Required:** Unlike local environments, Google Colab requires no setup on your machine. You can begin working on your project immediately by simply signing in with your Google account. This is particularly useful when experimenting with various models or datasets without worrying about local configuration issues.

How Google Colab Is Used in This Project

For the Mental Health Prediction Using ML Algorithms project, Google Colab is used in the following ways:

- **Data Loading and Preprocessing:** You can upload datasets to Google Drive or directly into Colab, then use **Pandas** to clean and preprocess the data.
- **Model Implementation:** Google Colab allows you to easily implement machine learning algorithms like Decision Trees and Random Forest .You can train and evaluate these models directly within the notebook.
- **Model Training and Evaluation:** With access to GPUs and TPUs, Colab accelerates model training, making it quicker to test different configurations and improve performance.
- **Data Visualization:** Use **Matplotlib** and **Seaborn** to create graphs and visualizations that help interpret the results of the models, such as accuracy, confusion matrices, or feature importance.
- **Collaborative Sharing:** You can share the notebook with team members, mentors, or stakeholders to review or collaborate on the project.

Advantages of Using Google Colab

- **Cost-Effective:** Google Colab is free and offers cloud resources without requiring any local hardware setup.
- **Ease of Use:** The notebook format makes it intuitive and easy to document code, results, and explanations all in one place.
- **Scalability:** Google Colab scales well to larger datasets and complex machine learning models due to its access to high-performance hardware.

5.3.4 PYTHON LIBRARIES

USES

Pandas is used to load, clean, and preprocess the dataset. It provides efficient data structures like DataFrames for handling tabular data (rows and columns). You can read data from CSV, Excel, or SQL databases, clean missing values, and filter or transform the data before applying machine learning models.

NumPy is used to handle numerical operations, such as array manipulation and mathematical computations. Many machine learning algorithms, including Decision Trees and Random Forests, require mathematical operations that NumPy performs efficiently.

Matplotlib is used for creating static, animated, and interactive visualizations. It helps in visualizing the model's performance and the data.

Seaborn builds on Matplotlib and provides a higher-level interface for creating more attractive and informative statistical graphics. It's useful for visualizing relationships between features, distributions of data, and model evaluation results.

Together, these libraries provide the necessary tools for data preprocessing, implementing machine learning models, evaluating performance, and visualizing results, making Python the ideal choice for your mental health prediction project using machine learning

CHAPTER 6
SYSTEM DESIGN

6.1 DATAFLOW DIAGRAM

This also known as bubble chart. It is a simple graphical Formalism that can be used to represent a system in terms of the input data to the system, various Processing carried out on these data, and the output data is generated by the system. It maps out the flow of information for any process or system, how data is processed in terms of inputs and outputs.

It uses defined symbols like rectangles, circles and arrows to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyse an existing system or model of a new one. It can often visually "say" things that would be hard to explain in words and they work for both technical and non- technical.

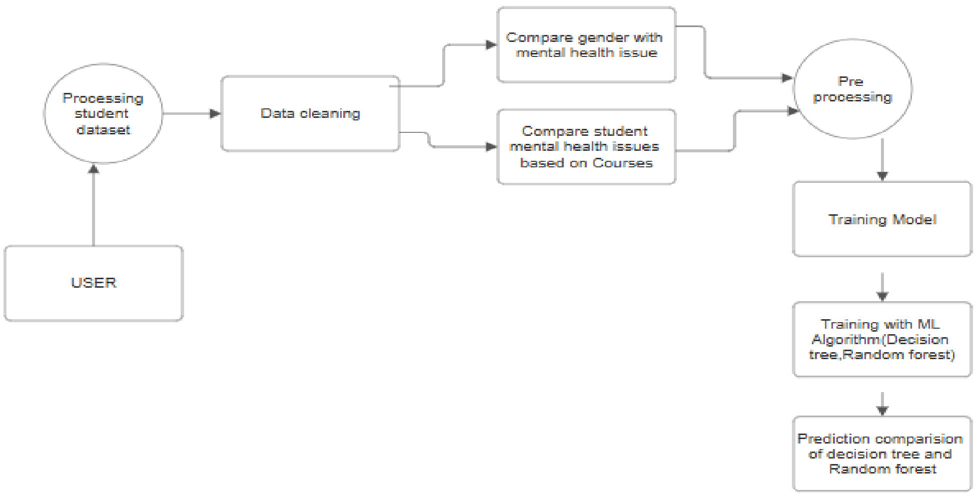

Fig 6.1. Dataflow diagram

6.2 USECASE DIAGRAM

Use Case during requirement elicitation and analysis to represent the functionality of the system. Use case describes a function by the system that yields a visible result for an actor. The identification of actors and use cases result in the definitions of the boundary of the system i.e., differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors are on the outside of the system's border, whilst the use cases are on the inside. The behaviour of the system as viewed through the eyes of the actor is described in a use case.

It explains the system's role as a series of events that result in a visible consequence for the actor. Use Case Diagrams: What Are They Good For? The objective of a use case diagram is to capture a system's dynamic nature. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and(State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

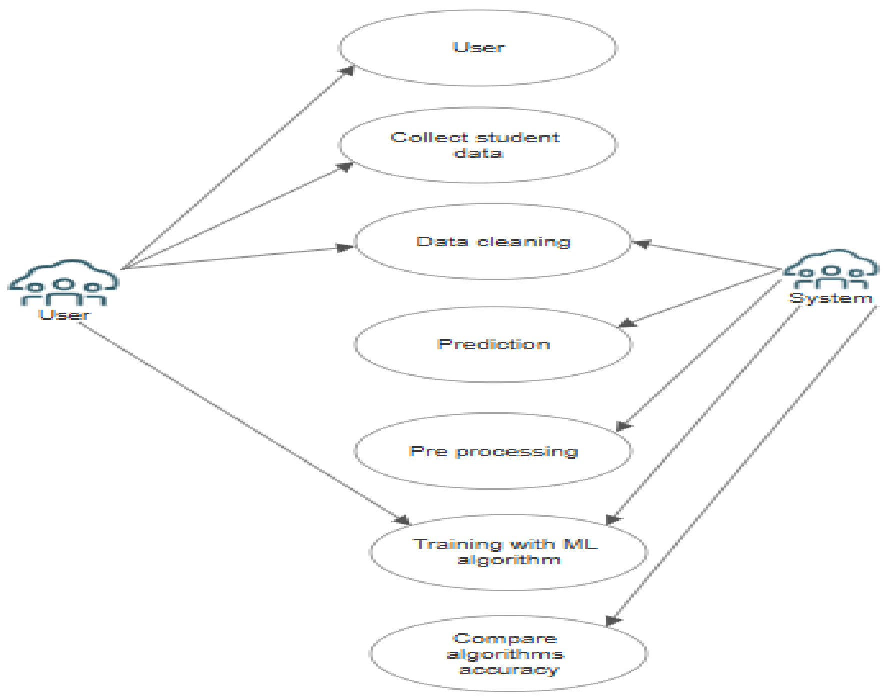

Fig 6.2. Usecase diagram

6.3 CLASS DIAGRAM

Model class structure and contents using design elements such as classes, packages and objects. Class diagram describe the different perspective when designing a system-conceptual, specification and implementation.

Classes are composed of three things: name, attributes, and operations. Class diagram also display relationships such as containment, inheritance, association etc. The association relationship is most common relationship in a class diagram. The association shows the relationship between instances of classes

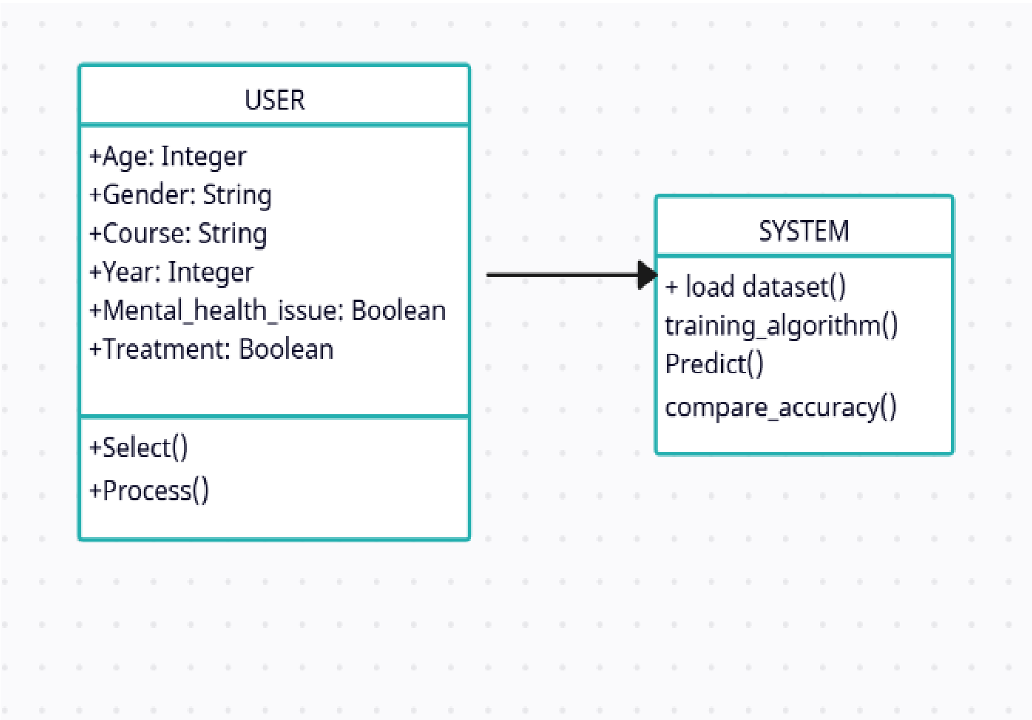

Fig 6.3. Class diagram

6.4 SEQUENCE DIAGRAM

Displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects). Objects: An object can be thought of as an entity that exists at a specified time and has a definite value, as well as a holder of identity. A sequence diagram depicts item interactions in chronological order. It illustrates the scenario's objects and classes, as well as the sequence of messages sent between them in order to carry out the scenario's functionality. In the Logical View of the system under development, sequence diagrams are often related with use case realisations.

Event diagrams and event scenarios are other names for sequence diagrams. A sequence diagram depicts multiple processes or things that exist simultaneously as parallel vertical lines (lifelines), and the messages passed between them as horizontal arrows, in the order in which they occur. This enables for the graphical specification of simple runtime scenarios.

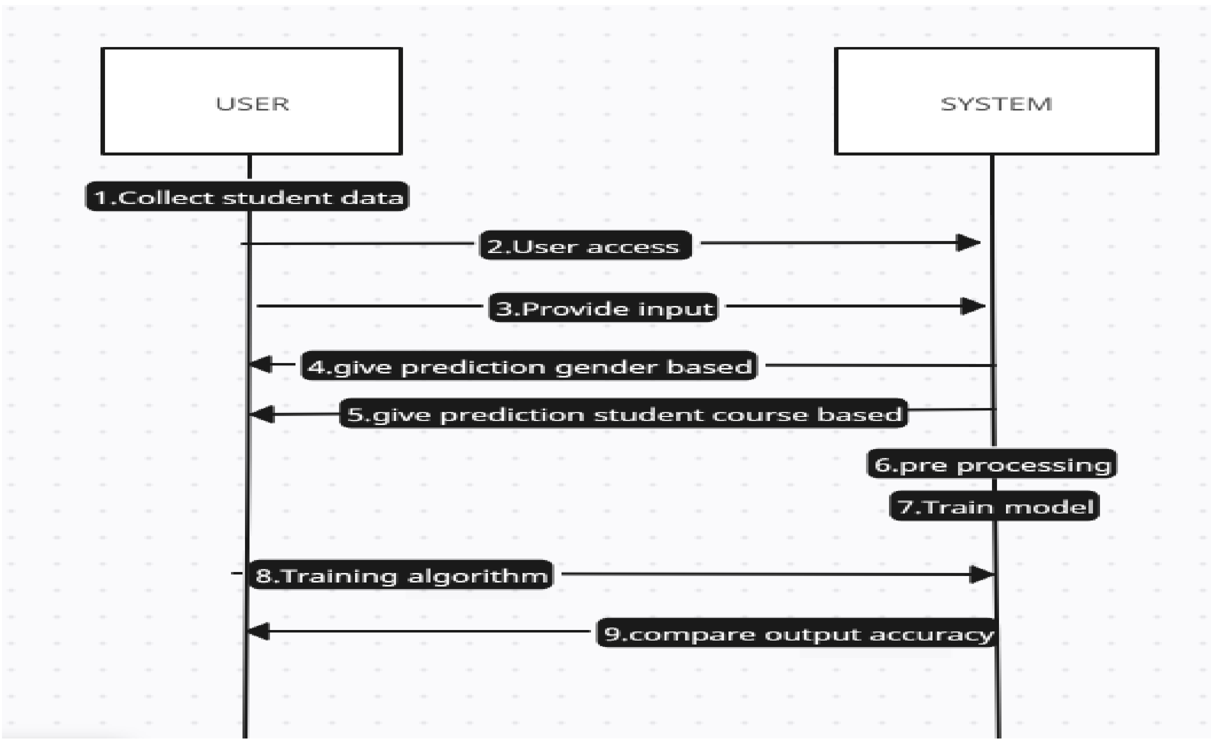

Fig 6.4. Sequence diagram

6.5 STATE CHART DIAGRAM

A state chart diagram describes a state machine which shows the behaviour of classes. It shows the actual changes in state no processes commands that create those changes and is the dynamic behaviour of objects over time by model1ing the life cycle of objects of each class. It describes how an object is changing from one state to another state. There are mainly two states in State Chart Diagram: 1. Initial State 2. Final-State. Some of the components of State Chart Diagram are:

State: It is a condition or situation in life cycle of an object during which it’s satisfies same condition or performs some activity or waits for some event. Transition: It is a relationship between two states indicating that object in first state performs some actions and enters into the next state or event. Event: An event is specification of significant occurrence that has a location in time and space

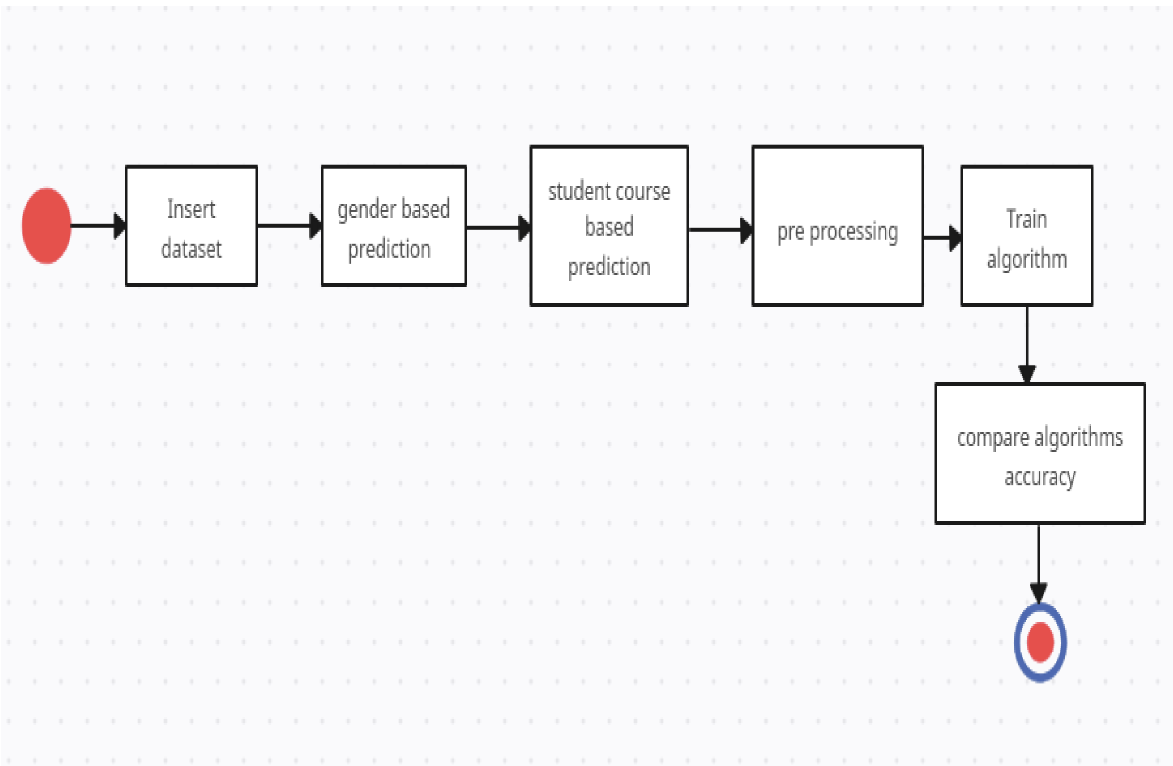

Fig 6.5. State chart diagram

CHAPTER 7
SYSTEM TESTING

System testing is essential in verifying the accuracy, robustness, and reliability of a machine learning model designed for mental health prediction. The following testing phases aim to ensure that each part of the system performs effectively, from data processing to prediction and user interface interaction.

7.1 UNIT TESTING

Data Preprocessing: Test each step in the preprocessing pipeline (e.g., handling missing values, data normalization, and encoding categorical variables) to ensure that data is correctly prepared for model training.

Feature Selection: Validate that the most relevant features, such as symptoms of anxiety, depression, and stress, are accurately identified and used in the models, ensuring optimized model performance.

Model Algorithms: Evaluate the Decision Tree and Random Forest algorithms separately, testing their ability to correctly classify mental health conditions using sample datasets. Ensure each algorithm performs as expected and generates appropriate metrics.

Performance Metrics: Assess individual model accuracy, precision, recall, and F1 score to confirm each model meets the desired performance standards. Compare results with established benchmarks to gauge the system's effectiveness.

Error Handling: Test the system’s responses to potential errors, such as missing input data or out-of-range values, ensuring appropriate error messages and recovery mechanisms are in place.

7.2 USER ACCEPTANCE TESTING

End-to-End Functionality: Conduct comprehensive testing of the entire system, from data input to mental health prediction output, using real-world datasets. Evaluate the system's overall functionality, ensuring predictions align with clinical insights.

Model Accuracy and Responsiveness: Test the speed and accuracy of predictions across various testing conditions. Assess model performance under diverse data inputs, ensuring predictions remain accurate and timely.

User Interface: Ensure the user interface is intuitive and accessible, allowing healthcare professionals and users to interact effectively with the system.

Cross-Platform Compatibility: Verify that the application works across different devices and operating systems to ensure wide usability.

User Accessibility: Test accessibility features to make the system usable for individuals with disabilities, such as screen readers or high-contrast modes.

7.3 INTEGRATION TESTING

Model Integration with Data Processing Pipelines: Check the smooth integration of machine learning models with preprocessing components to ensure that data flows correctly from preprocessing to prediction.

Cross-Module Communication: Test communication between different modules, such as from data preprocessing to prediction and from prediction to result display, ensuring accurate data transfer at each stage.

Performance Under Stress: Simulate heavy loads to observe the system's ability to handle high volumes of data without compromising accuracy speed.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

This project demonstrates the potential of machine learning algorithms, specifically Decision Tree and Random Forest classifiers, to predict mental health conditions such as anxiety, depression, and stress with accuracy and reliability. By leveraging real-world survey data from diverse sources, this approach highlights how predictive models can aid in identifying individuals at risk and supporting early intervention. Using Google Co-lab as the development platform, we implemented a workflow that included data pre-processing, feature engineering, model training, and evaluation. Google Co-lab's robust environment allowed us to efficiently utilize some resources for faster computation, integrate interactive visualizations, and streamline the collaboration and deployment process.

The analysis underscored the importance of selecting key features, such as demographic details, lifestyle factors, and psychosocial indicators, which significantly influence mental health outcomes. The Random Forest algorithm emerged as particularly effective due to its robustness and ability to handle complex feature interactions, while the Decision Tree model provided interpretability, making it valuable for understanding the impact of individual factors. Random forest has better accuracy.

8.2 FUTURE ENHANCEMENTS

The "Mental Health Prediction using Machine Learning Algorithms" project can be enhanced by integrating additional machine learning algorithms to improve prediction accuracy and robustness. Algorithms like Support Vector Machines (SVM), Gradient Boosting, or XG Boost could be explored alongside Decision Trees and Random Forests to compare and find the most efficient model. Additionally, expanding the dataset by including more diverse data points, such as social media activity, sleep patterns, or lifestyle habits, can provide a more comprehensive understanding of mental health. Incorporating deep learning techniques, particularly Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) networks, can help capture complex temporal patterns in mental health data, further enhancing prediction capabilities. The project can also be extended by creating a more user-interactive website that offers personalized recommendations based on the predictions and includes real-time analysis with visualization tools to make the results more actionable for users. These advancements will ensure the system is more accurate, scalable, and user-friendly, providing deeper insights into mental health prediction.

CHAPTER 9

APPENDICES

9.1 APPENDIX A (Sample Coding)

Initialization

```
!pip install seaborn –U

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.__version__

from google.colab import files

uploaded = files.upload()

data = pd.read_csv('/content/Student_Mental_health.csv')

df_clean = data.copy()
df_clean.head(5)
```

Data Cleaning

```
# Check dtypes

print('Col types:\n',df_clean.dtypes,'\n','='*25,sep='')

# Check for NA values

print('Number of NA per Col:')

df_clean.isna().sum()

# Since only age has NA we can replace it with the mean as an int

df_clean['Age'] = df_clean['Age'].fillna(df_clean['Age'].mean()).astype('int64')

df_clean.isna().sum()

# Rename columns for clarity

df_clean.rename(columns={

'Choose your gender':'Gender',

'What is your course?':'Course',

'Your current year of Study':'Year',
```

```
'What is your CGPA?':'GPA',
'Marital status':'Married',
'Do you have Depression?':'Depression',
'Do you have Anxiety?':'Anxiety',
'Do you have Panic attack?':'Panic_Attacks',
'Did you seek any specialist for a treatment?':'Treatment'}, inplace=True)
```

Gender to Responses & Conditions

```
# Compare Gender to Mental Health conditions
sns.countplot(data=df_clean, hue='Anxiety', x='Gender', hue_order=['Yes','No'])
plt.title('Anxiety by Gender')
plt.xlabel('')
plt.show()
sns.countplot(data=df_clean, hue='Depression', x='Gender', hue_order=['Yes','No'])
plt.title('Depression by Gender')
plt.xlabel('')
plt.show()
sns.countplot(data=df_clean, hue='Panic_Attacks', x='Gender', hue_order=['Yes','No'])
plt.title('Panic Attacks by Gender')
plt.xlabel('')
plt.show()
```

Course to Conditions with Gender

```
# Compare courses to Anxiety and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean, x='Anxiety', y='Course', hue='Gender',order=['Yes','No'])
plt.show()
# Compare Courses to Depression and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean, x='Depression', y='Course', hue='Gender',order=['Yes','No'])
plt.show(
# Compare Courses to Panic Attacks and Gender
plt.figure(figsize=(10, 10))
sns.swarmplot(data=df_clean, x='Panic_Attacks', y='Course',
hue='Gender',order=['Yes','No'])
```

```
plt.show()
```

Modelling

```
for col in df_clean.columns:
  print(df_clean[col].value_counts().sort_index(),'\n','='*50,sep='')
```

Data Pre-Processing

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report,accuracy_score
from sklearn.preprocessing import StandardScaler

df_model = df_clean.copy()
df_model.drop(columns='Timestamp',inplace=True)
df_model.dtypes

#Convert Binary columns into numeric
for col in bool_cols:
df_model[col] = df_model[col].replace({'Yes':1,'No':0})
```

Train models

```
# Split data
X = df_model.drop(columns=['Depression'])
y = df_model['Depression']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,random_state=89)
# Decision Tree

decision_tree = DecisionTreeClassifier(random_state=43) decision_tree.fit(X_train,y_train)
print('Decision Tree:',cross_val_score(decision_tree, X_train, y_train, cv=8).mean())

# Random Forest

random_forest = RandomForestClassifier(random_state=43)
random_forest.fit(X_train,y_train) print('Random Forest:',cross_val_score(random_forest,
X_train, y_train, cv=8).mean())
```

Decision tree and random forest accuracy

```
import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Make predictions on the test set

pred_dtree = decision_tree.predict(X_test)

pred_rforest = random_forest.predict(X_test)

# Calculate accuracy as percentage

accuracy_dtree = accuracy_score(y_test, pred_dtree) * 100

accuracy_rforest = accuracy_score(y_test, pred_rforest) * 100

# Print accuracies

print(f'Decision Tree Accuracy: {accuracy_dtree:.2f}%')

print(f'Random Forest Accuracy: {accuracy_rforest:.2f}%')

# Print classification reports

print('Decision Tree Classification Report:\n', classification_report(y_test, pred_dtree))
print('Random Forest Classification Report:\n', classification_report(y_test, pred_rforest))

 # Compute confusion matrices

cm_dtree = confusion_matrix(y_test, pred_dtree)

cm_rforest = confusion_matrix(y_test, pred_rforest)

# Plot confusion matrices

fig, axes = plt.subplots(1, 2, figsize=(10, 5))

sns.heatmap(cm_dtree, annot=True, fmt='d', cmap='Blues', ax=axes[0])
axes[0].set_title('Decision Tree Confusion Matrix')

axes[0].set_xlabel('Predicted Labels')

axes[0].set_ylabel('True Labels')
```

sns.heatmap(cm_rforest, annot=True, fmt='d', cmap='Greens', ax=axes[1])
axes[1].set_title('Random Forest Confusion Matrix')

axes[1].set_xlabel('Predicted Labels')

axes[1].set_ylabel('True Labels')

plt.tight_layout()

plt.show()

Comparison of algoritms

import matplotlib.pyplot as plt

Define model names and accuracies

models = ['Decision Tree', 'Random Forest']

accuracies = [accuracy_dtree, accuracy_rforest]

Plot the bar chart

plt.figure(figsize=(5, 5))

bars = plt.bar(models, accuracies, color=['skyblue', 'orange'])

Annotate the accuracy percentages on top of each bar

for bar, accuracy in zip(bars, accuracies): plt.text(bar.get_x() + bar.get_width() / 2,

bar.get_height() - 5, f'{accuracy:.2f}%', ha='center', va='bottom', color='black', fontsize=12)

Add titles and labels

plt.title('Comparison of Model Accuracies', fontsize=16) plt.ylabel('Accuracy (%)', fontsize=12) plt.xlabel('Models', fontsize=12) plt.ylim(0, 100) # Set y-axis range to 0-100

for clarity

Highlight which model performs better better_model = models[accuracies.index(max(accuracies))]

plt.text(0.5, 90, f'{better_model} has better accuracy!', ha='center', color='green', fontsize=14, bbox=dict(facecolor='white', edgecolor='green')) # Show the plot plt.tight_layout()

9.2 APPENDIX B (Screenshots)

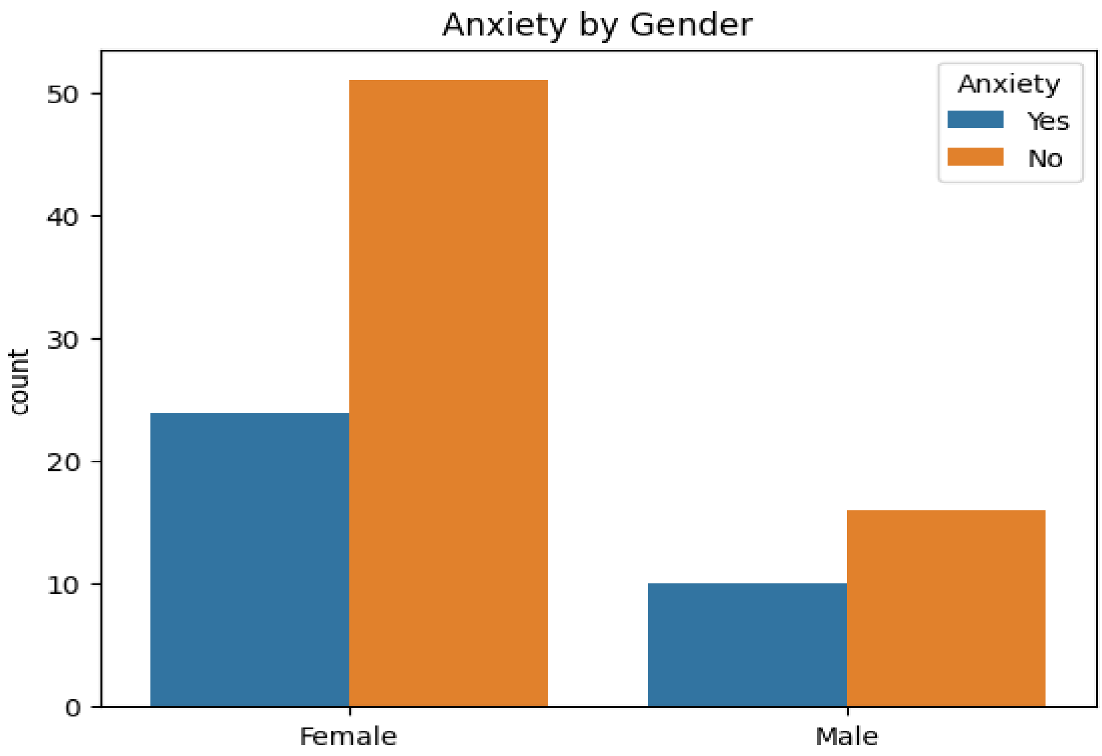

Fig 9.2.1. Anxiety by gender

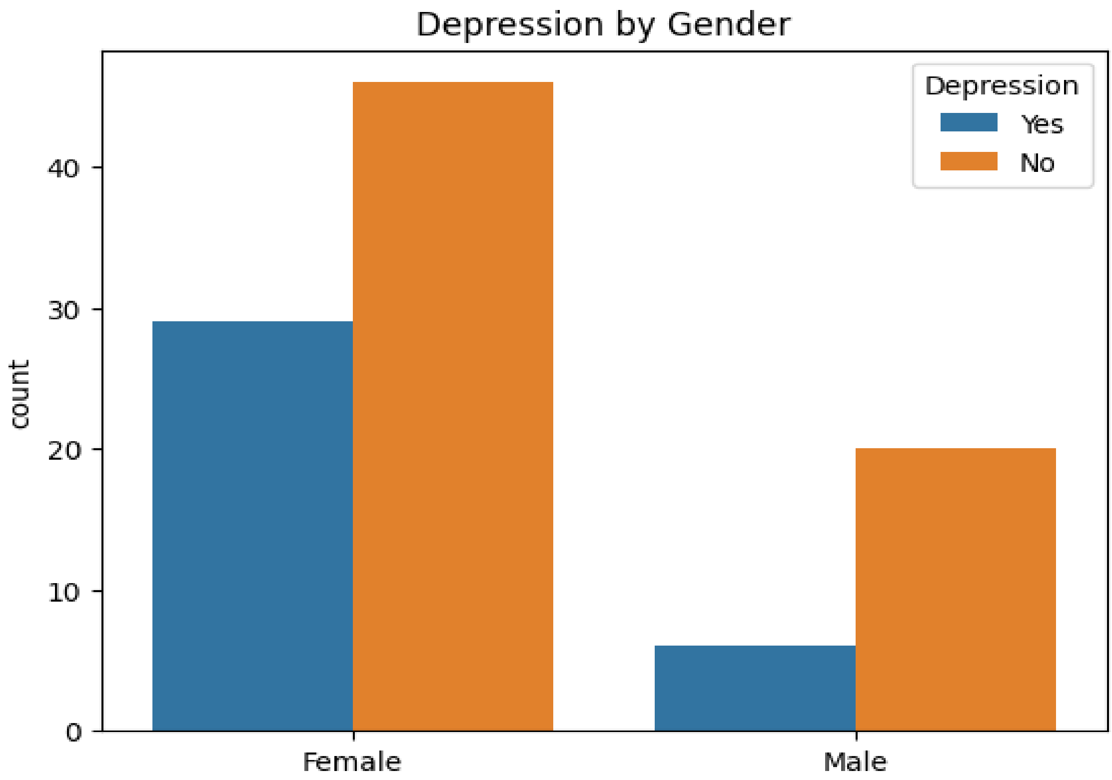

Fig 9.2.2. Depression by gender

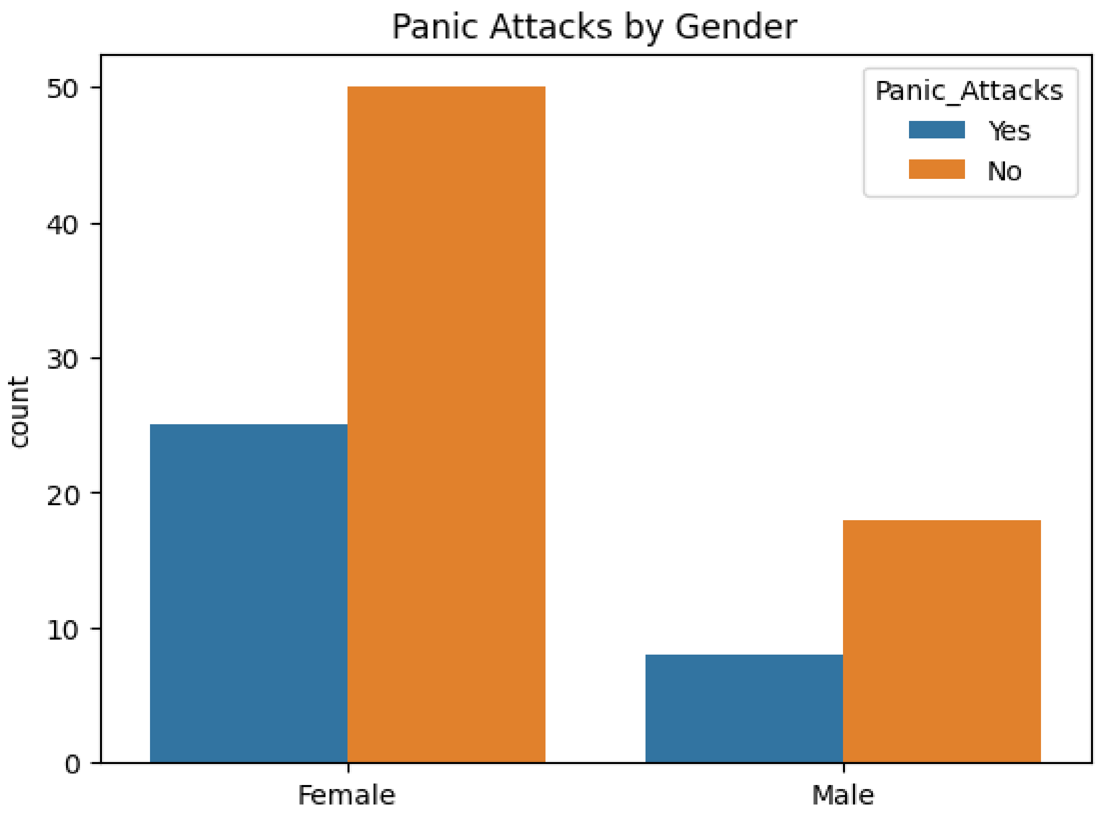

Fig 9.2.3. Panic attack by gender

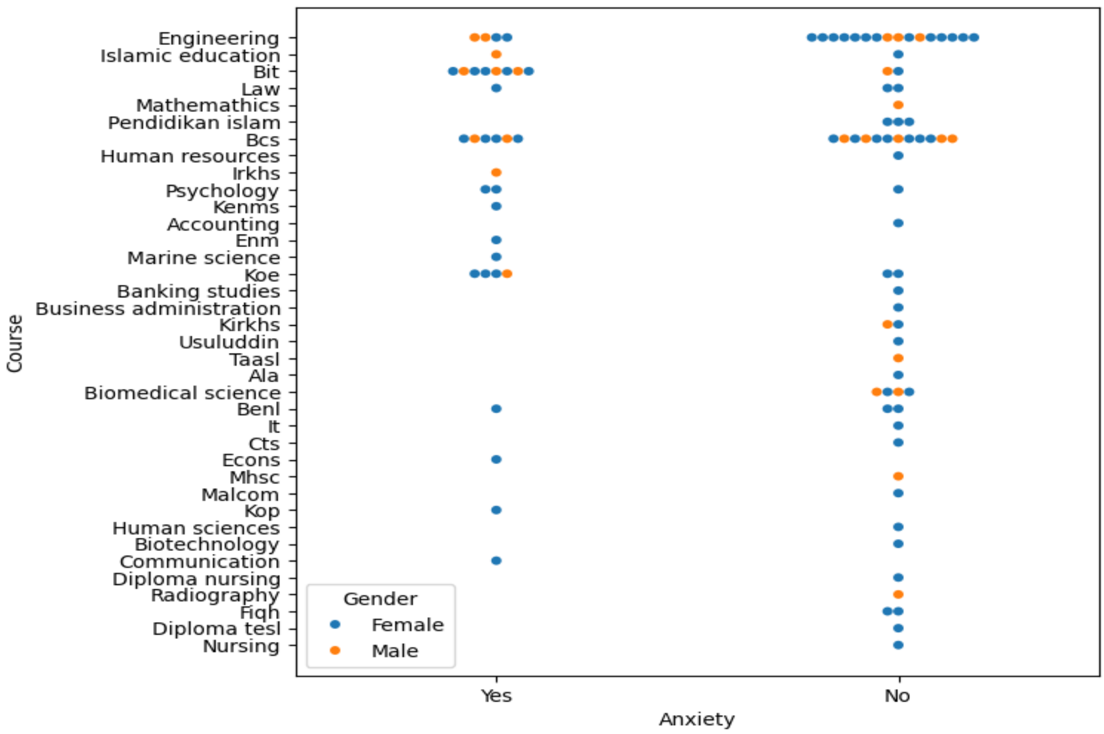

Fig 9.2.4. Anxiety by student's course

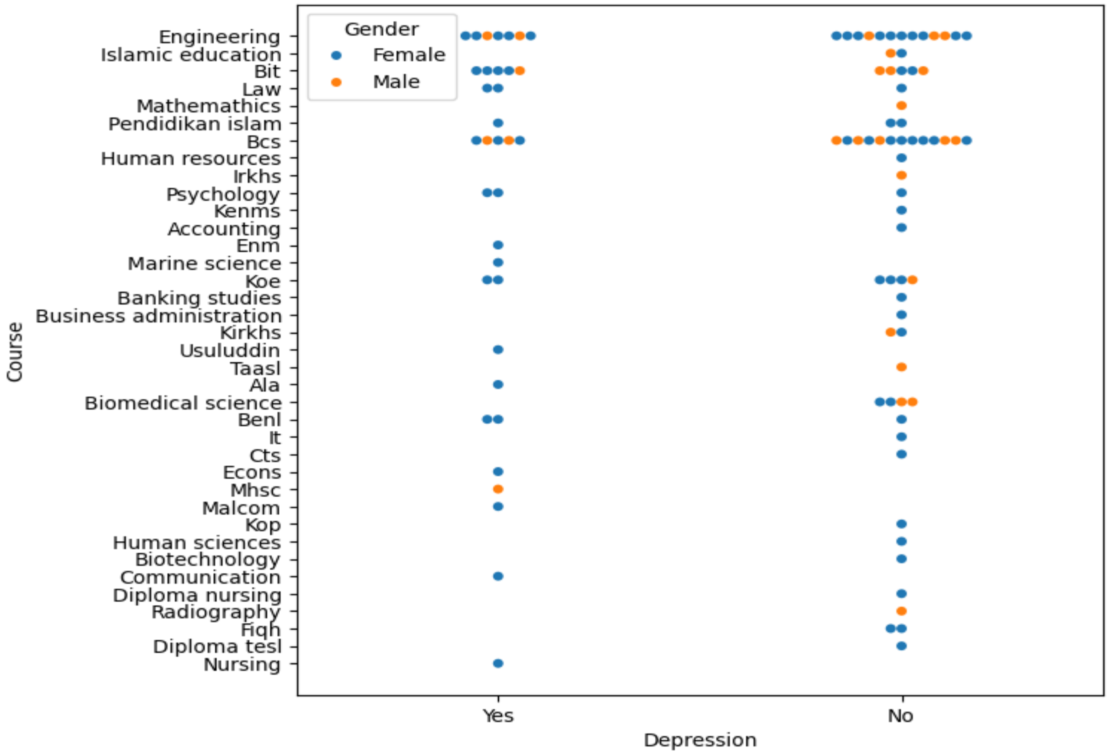

Fig 9.2.5. Depression by student's course

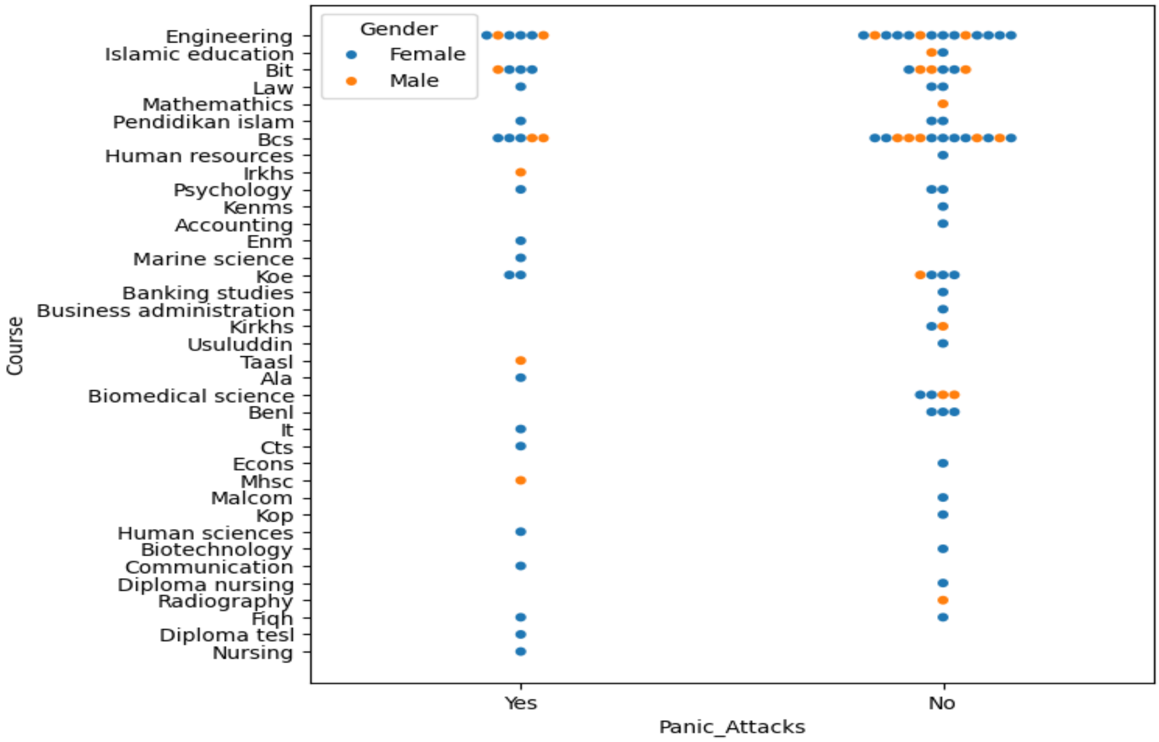

Fig 9.2.6. Panic attack by student's course

9.3 RESULT ANALYSIS

Decision Tree Accuracy: 64.52%
Random Forest Accuracy: 87.10%

Decision Tree Classification Report:

```
              precision    recall  f1-score
support

           0       0.80      0.60      0.69        20
           1       0.50      0.73      0.59        11

    accuracy                           0.65        31
   macro avg       0.65      0.66      0.64        31
weighted avg       0.69      0.65      0.65        31
```

Random Forest Classification Report:

```
              precision    recall  f1-score
support

           0       0.83      1.00      0.91        20
           1       1.00      0.64      0.78        11

    accuracy                           0.87        31
   macro avg       0.92      0.82      0.84        31
weighted avg       0.89      0.87      0.86        31
```

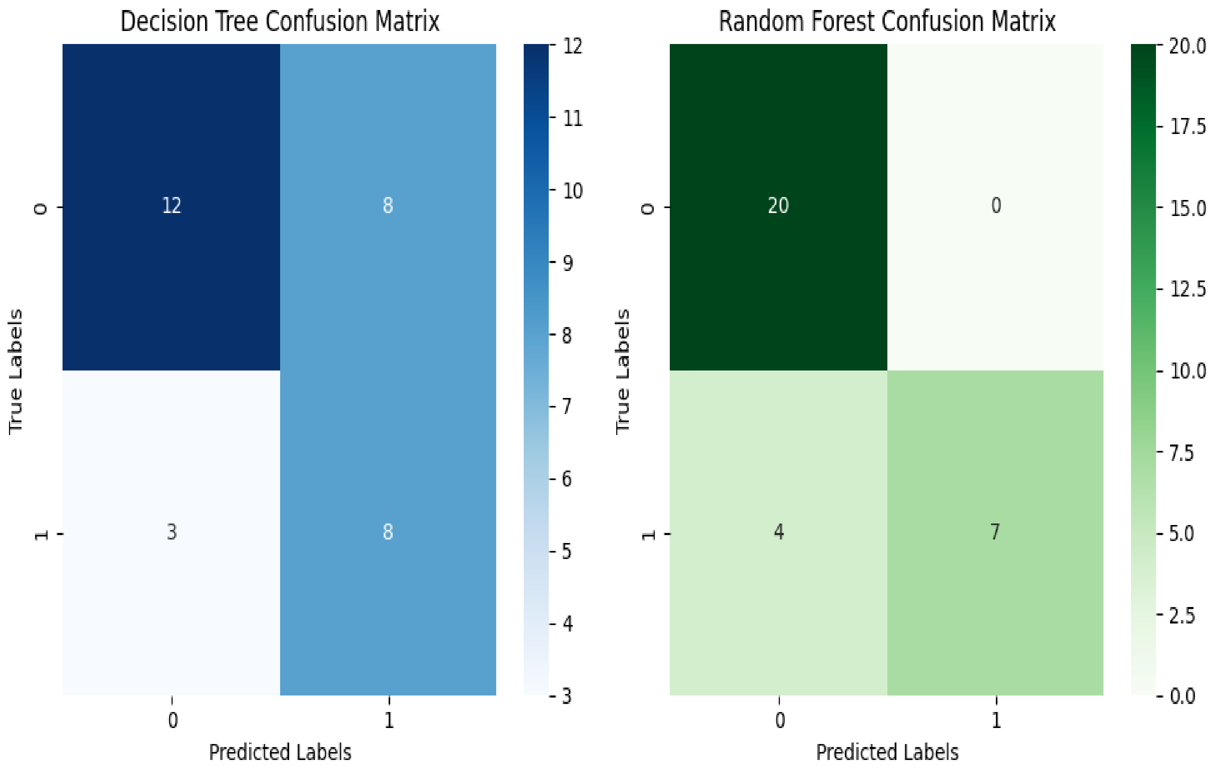

Fig 9.3.1 Confusion matrix

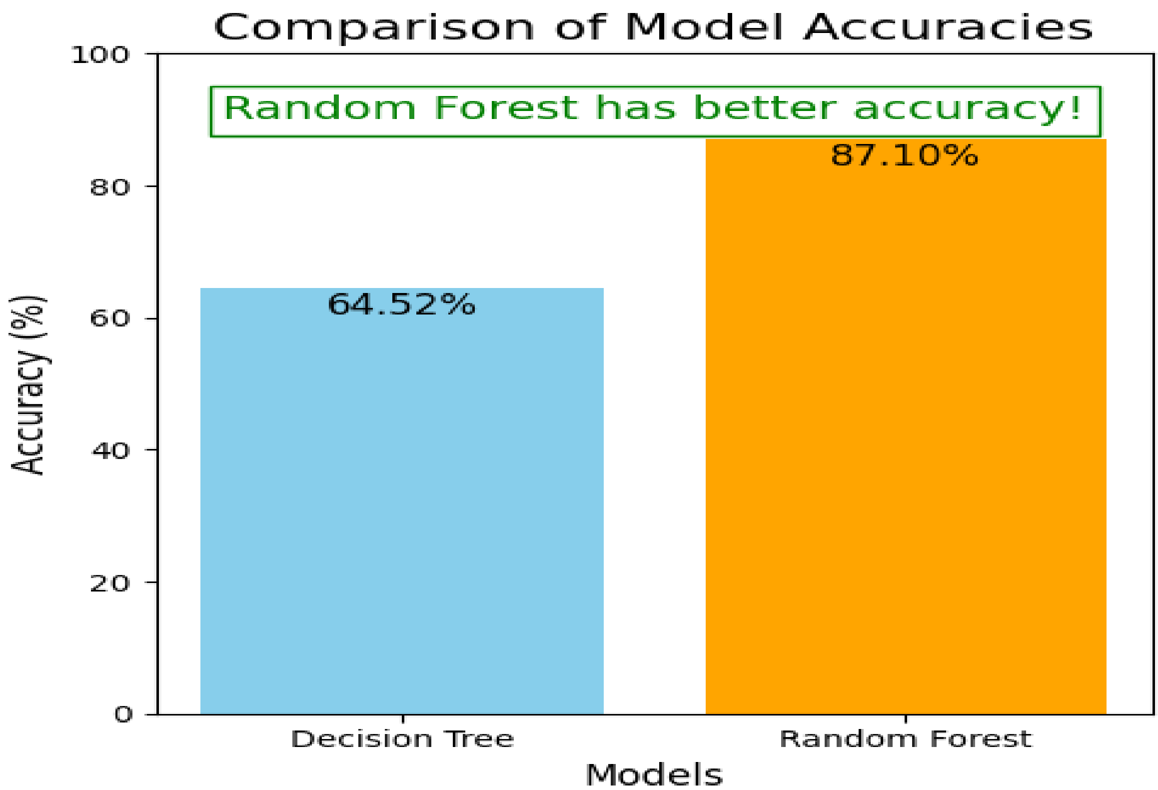

Fig 9.3.2 Algorithm Comparison chart

CHAPTER 10

REFERENCES

[1]Sandip Roy , P.S.Aithal , Rajesh Bose,"Judging mental health disorders using the decision tree models",IEEE International conference,2017.

[2]Konda Vaishnavi , U Nikhitha Kamath , B Ashwath Rao and NV Subba Reddy,"Predicting mental health illness using machine learning", IEEE International conference,2021.

[3]Ms.Sumathi and Dr.B.Poorna,"Prediction of mental health problems among children using machine learning techniques", IEEE International conference,2016.

[4]Jakub Tomasik, Sung yeon sarah han, Jason D.Cooper, "A machine learning algorithm to differentiate bipolar disorder from major depressive disorder using an online mental health questionnaire and blood biomarker data",IEEE International conference, 2021.

[5]Rohizah abd rahman, Khairuddin omar, "Application of machine learning methods in mental health detedtion", IEEE International conference,2020.

[6]Jetli Chung and Jason Teo," Mental Health Prediction Using Machine Learning: Taxonomy, Applications, and Challenges", ResearchGate Journel,2022.

[7]Satvik Gurjar, Chetna Patil, Ritesh Suryawanshi, Madhura Adadande, Ashwin Khore, Noshir Tarapore., "Mental Health Prediction Using Machine Learning",International Research Journal of Engineering and Technology (IRJET),2022.

[8]Fadhluddin Sahlan, Faris Hamidi ,Muhammad Zulhafizal ,Misrat Muhammad ,Haziq AdliSharyar ,WaniYonis., "Prediction of Mental Health Among University Students", ResearchGate Journel,2021.