AI-Driven Algorithms for Psychological Health Assessment Using Machine Learning

A Capstone Project Report

Submitted by

GOWTHAMI K SHETTY	**20231BCI0022**
NISARGA V SHETTY	**20231BCI0048**
KEERTHI N	**20231BCI0029**
CHEERANJEEVI	**20231BCI0015**

Under the guidance of

Ms.Kumari K
Assistant Professor,
School of CSE&IS

in partial fulfillment for the award of the degree

of

BACHELOR OF COMPUTER APPLICATIONS

At

SCHOOL OF INFORMATION SCIENCE

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

BACHELOR OF COMPUTER APPLICATIONS

SCHOOL OF INFORMATION SCIENCE

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

CERTIFICATE

This is to certified that the Capstone Project report **"AI-Driven Algorithms for Psychological Health Assessment Using Machine Learning "** being submitted by ***GOWTHAMI K SHETTY***, ***NISARGA V SHETTY***, ***KEERTHI N***, ***CHEERANJEEVI*** bearing roll number ***20231BCI0022, 20231BCI0048, 20231BCI0029***, ***20231BCI0015,*** in partial fulfillment of requirement for the award of degree of **Bachelor of Computer Applications** is a bonafide work carried out under my supervision.

Ms.Kumari K
Assistant professor,
School of CSE&IS
Presidency University

Dr. Vetrimani Elangovan
Head of the Department (PSIS),
Presidency School of CSE&IS
Presidency University

Dr. R Mahalakshmi
Associate Dean,
Presidency School of CSE&IS
Presidency University

Dr. Md. Sameeruddin Khan
Dean,
Presidency School of CSE&IS
Presidency University

ABSTRACT

Mental health disorders such as depression, anxiety, and stress have become increasingly prevalent, especially among young adults and adolescents. Traditional assessment methods, including self-report questionnaires and clinical interviews, often face challenges such as stigma, underreporting, and limited accessibility. With the rise of artificial intelligence (AI) and interactive technologies, there is a growing opportunity to develop alternative, engaging, and non-invasive tools for psychological assessment.

This project proposes an AI-driven system that utilizes interactive games to predict users' mental health states, specifically focusing on levels of depression, anxiety, and stress (DAS). The core idea is to design game-based environments that subtly elicit behavioral and cognitive responses, such as decision-making patterns, reaction times, and emotional choices. These behavioral signals are collected in real-time and processed using machine learning algorithms trained on labeled datasets, including data obtained through standardized scales like the DASS-21.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENT

The completion of project work brings with great sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion. First and foremost we indebted to the GOD ALMIGHTY for giving us the opportunity to excel our efforts to complete this project on time. We wish to express our deep sincere feelings of gratitude to our Institution, Presidency University, for providing us opportunity to do our education.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan,** Pro-Vice Chancellor, School of Engineering, and Dean, Presidency School of CSE and IS, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved professor **Dr. R Mahalakshmi**, Associate Dean, Presidency School of Information Science, **Dr Vetrimani Elangovan**, Professor and Head, Presidency School of Information Science, Presidency University for rendering timely help for the successful completion of this project.

We sincerely thank our project guide, **Ms. Kumari K,**Assistant Professor, School of CSE&IS for her guidance, help and motivation. Apart from the area of work, we learnt a lot from her, which we are sure will be useful in different stages of our life. We would like to express our gratitude to Faculty Coordinators and Faculty, for their review and many helpful comments.

We would like to acknowledge the support and encouragement of our friends.

Student Name	***ID Number***
GOWTHAMI K SHETTY	***20231BCI0022***
NISARGA V SHETTY	***20231BCI0048***
KEERTHI N	***20231BCI0029***
CHEERANJEEVI	***20231BCI0015***

CHAPTER-1

INTRODUCTION

1.1 Background and Motivation

Mental health issues such as depression, anxiety, and stress are among the most significant health challenges in the modern world. According to the World Health Organization (WHO), depression affects more than 264 million people globally, while anxiety disorders impact over 284 million. These conditions not only reduce the quality of life but also significantly affect productivity and social well-being. However, psychological health is often neglected due to social stigma, lack of awareness, and limited access to mental health professionals.

Traditional methods of psychological assessment typically involve self-report questionnaires and clinical interviews. While effective, these methods suffer from several limitations such as response bias, time constraints, and the requirement for trained professionals. In this digital era, where mobile devices and the internet are deeply embedded in daily life, innovative approaches to mental health assessment are gaining popularity.

One promising approach involves the integration of Artificial Intelligence (AI) with interactive games. Games provide an engaging and non-intrusive environment where users can freely express behaviors and emotions. This natural interaction offers a novel way to gather data that reflects users' psychological states. By embedding intelligent algorithms into these games, it becomes possible to assess users' mental health conditions in real-time.

1.2 Problem Statement

There exists a significant gap between the increasing demand for mental health services and the availability of timely, affordable, and accurate diagnostic tools. Current psychological screening techniques are either too invasive or require clinical settings that may not be accessible to everyone. Moreover, users may not be willing to share sensitive information or may provide misleading responses due to stigma or lack of self-awareness.

To address these limitations, this project proposes the development of an AI-driven framework capable of predicting levels of depression, anxiety, and stress through behavioral data collected during gameplay. By analyzing user interactions, choices, and response patterns within games, the system can learn to identify indicators of psychological distress without explicit questioning.

1.3 Objectives

The primary goal of this project is to develop a non-invasive, accessible, and intelligent mental health assessment system using AI and interactive games. The key objectives include:

Designing and developing a set of interactive games tailored to elicit behavioral cues relevant to psychological health.

Collecting user interaction data during gameplay and preprocessing it for feature extraction.

Implementing machine learning models to predict the levels of depression, anxiety, and stress (DAS).

Evaluating the accuracy and robustness of the predictive models.

Providing users with real-time, confidential feedback based on their predicted mental state.

1.4 Significance of the Study

This project is significant for several reasons:

Early Detection: Enables early identification of psychological issues without the need for clinical intervention.

Accessibility: Can be deployed on mobile devices or web platforms, reaching a broader audience.

Non-Invasive Nature: Removes the barriers posed by traditional assessments, making it more acceptable to users.

Data-Driven Insights: Provides real-time analytics and insights into user behavior that can be valuable for mental health research.

By leveraging AI, this project contributes to the growing field of digital mental health and aims to democratize access to psychological well-being tools.

1.5 Scope of the Project

The scope of this project includes: Development of interactive games focusing on cognitive ,tasks, decision-making, and emotional response.

Integration of backend data collection and processing mechanisms to capture user inputs.

Application of supervised machine learning techniques (such as SVM, Random Forest, or

Neural Networks) for prediction of DAS scores.

Use of validated mental health questionnaires (e.g., DASS-21) for initial labeling and model training.

Frontend user interface for game interaction and results visualization.

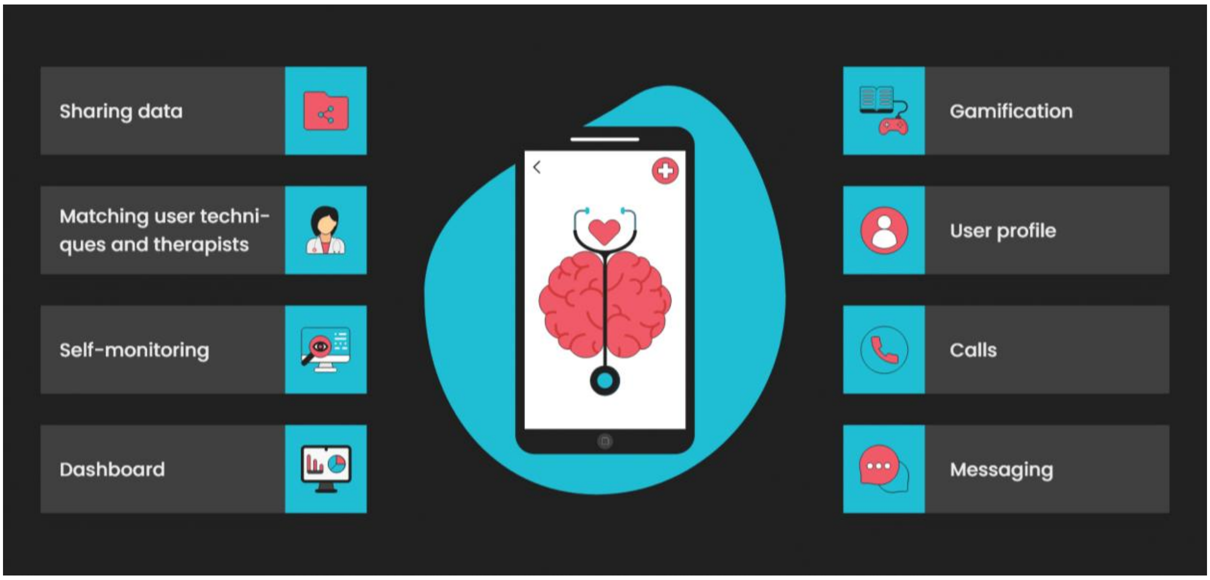
Limitations of the project include potential bias due to sample diversity, reliance on internet access for web-based deployment, and challenges in validating results without clinical comparison.

1.6 Methodology Overview

The project follows a multi-phase methodology:

1. Game Design and Development: Creating gamified tasks based on psychological principles.
2. Data Collection: Gathering behavioral and interaction data from players.
3. Feature Engineering: Identifying key indicators such as response time, decision patterns, and accuracy.
4. Model Training and Validation: Using labeled datasets to train AI models to predict DAS scores.
5. Evaluation and Feedback: Testing model performance and delivering personalized feedback to users.

Each phase is supported by rigorous experimentation and evaluation to ensure validity and reliability.



1.1 BLUEPRINT SKETCH

CHAPTER-2

LITERATURE SURVEY

The integration of Artificial Intelligence (AI) in psychological health assessment has garnered increasing attention in recent years. As mental health disorders like depression, anxiety, and stress become more prevalent, traditional methods of diagnosis—often reliant on self-reporting and clinical interviews—struggle to scale efficiently. Consequently, researchers and developers are increasingly turning to machine learning and interactive technologies to enhance the early detection and monitoring of psychological conditions.

Traditional Approaches to Mental Health Assessment

Historically, psychological assessments were conducted using standardized self-reporting tools such as the DASS-21 or DASS-42 (Depression, Anxiety, and Stress Scales). These tools are effective in controlled clinical environments, but they suffer from limitations such as subjectivity, response bias, and the lack of real-time monitoring capabilities. Although beneficial in diagnosis, these manual methods require professional administration and are time-consuming when applied to large populations.

Rise of Machine Learning in Psychological Health

Machine learning algorithms offer a promising alternative by automating the classification of mental health conditions based on patterns found in data. Various studies have demonstrated the application of classifiers such as Random Forest, Support Vector Machines (SVM), Naïve Bayes, and Gradient Boosting for predicting levels of depression, anxiety, and stress. These models are typically trained on labeled datasets derived from questionnaire results or sensor data, achieving high accuracy in mental health prediction tasks.

Ensemble models, which combine multiple algorithms to improve prediction performance, have proven particularly effective. These models reduce variance and bias, and their robustness makes them suitable for applications in mental health, where noisy and incomplete data are common. Studies have shown that majority-voting classifiers and boosting algorithms outperform individual models in terms of precision, recall, and overall reliability.

Behavioral and Physiological Data Sources

Recent advancements have expanded data collection beyond self-reporting to include behavioral and physiological inputs. Wearable devices, for instance, can monitor heart rate variability, skin conductance, sleep quality, and movement patterns. These indicators are strongly correlated with psychological states such as stress and anxiety. By applying machine learning to such sensor data, researchers have been able to build systems capable of passive mental health monitoring without requiring active user participation.

In parallel, mobile phone usage data—such as screen time, texting behavior, and location patterns—has emerged as a non-invasive way to infer psychological states. Behavioral analytics from smartphones have been successfully used to detect depressive symptoms based on reduced mobility or irregular social communication patterns.

Natural Language Processing and Voice Analysis

Natural language processing (NLP) and voice analysis have also gained traction in mental health assessment. The structure, sentiment, and complexity of language used by individuals can reveal significant insights into their mental state. Techniques like sentiment analysis, topic modeling, and emotion recognition from speech have been used to predict depression and anxiety levels. These methods are especially powerful when combined with conversational agents or chatbots, which can interact with users in natural dialogue formats.

Game-Based Assessment Strategies

One of the most innovative approaches in recent years involves the use of interactive games for psychological assessment. These games are designed to engage users while simultaneously collecting data about their behavior, decision-making, and emotional responses. Unlike traditional methods, game-based assessment is unobtrusive and can collect large volumes of data in real-time. It also enhances user engagement, making mental health monitoring more appealing to users who may avoid formal psychological testing.

Games can be used to simulate stress-inducing scenarios, measure reaction times, or assess emotional regulation skills. When combined with machine learning models, these games serve as both diagnostic tools and therapeutic interventions. Some research suggests that users are more likely to disclose sensitive psychological information when immersed in a game-like environment, thus increasing the reliability of assessments.

Limitations and Challenges

Despite the progress, several challenges persist. Data privacy and ethical concerns are paramount, especially when dealing with sensitive psychological information. The lack of standardized datasets and benchmarking protocols makes it difficult to compare different models and approaches. Moreover, many AI systems lack interpretability, which reduces trust among healthcare professionals and limits clinical adoption.

There is also a need for systems that are culturally adaptive and accessible to diverse populations. Bias in training data can lead to inaccurate predictions for underrepresented groups. Furthermore, while AI systems can assist in diagnosis and monitoring, they are not a replacement for clinical judgment and must be integrated into broader mental health care frameworks.

CHAPTER-3

DESIGN

The program is structured to assess psychological health by predicting levels of depression, anxiety, and stress through data-driven machine learning techniques. The complete pipeline is divided into two main components: frontend (API layer) and backend (model training and evaluation), both designed to work harmoniously for accurate mental health assessment.

System Overview

The system is designed as a modular application consisting of the following layers:

1. Frontend Layer (User Interface via Streamlit)
2. Application Layer (Game Logic + Questionnaire)
3. ML Model Layer (Trained classifiers using DASS-21 dataset)
4. Data Layer (Storage for user inputs and results)

The architecture follows a client-server model, where the frontend interacts with users, and the backend handles computation and prediction

.Frontend Design

The application is a Streamlit-based web app that integrates:

Three psychological assessments: PHQ-9 (Depression), GAD-7 (Anxiety), PSS (Stress)

Three wellness modules: Breathing exercise, affirmations, emotion recognition game

Text-to-speech for better accessibility

Assessment Functions

Each function represents a mental health assessment. They include:

1. depression_test()

Contains 9 questions from the PHQ-9 scale

Each question is a radio button with 4 options (0–3)

Calculates total score and interprets it as:

Minimal (0–4)

Mild (5–9)

Moderate (10–14)

Moderately Severe (15–19)

Severe (20–27)

2. anxiety_test()

Contains 7 questions from the GAD-7 scale

Score range: 0–21

Categories: Minimal to Severe anxiety

3. stress_test()

Contains 10 questions from PSS

Score range: 0–40

Interprets stress as Low, Moderate, or High

Wellness Games and Exercises

1. breathing_exercise()

Guides users through inhale, hold, and exhale cycle

Uses text-to-speech for vocal guidance

Loops a few breathing steps for mindfulness

2. affirmations()

Presents a list of positive affirmations

Option to hear them via voice

Enhances positivity and mood

3. emotion_game()

A simple quiz where users match emojis to emotions

Interactive and educational

Tracks score

Utility Function: speak(text)

Converts input text to audio using gTTS

Encodes and plays it directly in the app using Streamlit

Design Philosophy

Simplicity: Straightforward interface using Streamlit

Accessibility: Audio feedback for users with reading difficulty

Engagement: Mini-games to keep users involved

Modularity: Easy to add more features or assessments later

Backend Design (Model Training and Evaluation)

File: train_model.py

Purpose: Combines datasets from Depression, Stress, and Anxiety to train a general model.

Steps:

Loads and merges three CSV datasets.

Splits data into features (X) and target labels (y).

Applies a train-test split.

Trains a RandomForestClassifier.

Saves the model using pickle for future use.

This script serves as the base model for the frontend's prediction engine.

Comprehensive Model Evaluation Script

This script is more in-depth and handles:

Preprocessing: Label encoding for categorical variables and standardization of numeric features.

Model Training: Uses five classifiers:

K-Nearest Neighbors (KNN)

Support Vector Machine (SVM)

Naive Bayes

Random Forest

XGBoost

Function: run_classifiers()

This core function is applied independently for Depression, Anxiety, and Stress datasets:

Splits the dataset into train and test sets.

Trains and evaluates each classifier.

Prints the confusion matrix, accuracy, and classification report.

Implements a combined voting model by taking the majority of predictions from Random Forest, Naive Bayes, and XGBoost.

This ensemble-like combination is intended to increase prediction robustness by reducing model-specific biases.

Visualization and Comparative Analysis

The script visualizes model performance across each psychological label (Depression, Anxiety, Stress) using horizontal bar charts. This provides an intuitive understanding of which classifiers perform best for each condition.

Accuracy Metrics: Plotted for all six models (five individual + one combined).

Analysis Purpose: Helps in selecting the best-performing model and shows the value of ensemble prediction.

Interactive Game Module

This module consists of cognitive and reaction-based mini-games such as:

Memory Match Game

Color Reaction Test

Pattern Recognition Puzzle

These games are designed to simulate cognitive load, attention span, and reaction timing, indirectly indicating psychological states.

Modularity and Scalability

The architecture supports:

Modularity: Different ML models can be evaluated and swapped easily.

Scalability: New models or psychological conditions (like PTSD or OCD) can be added with minimal changes.

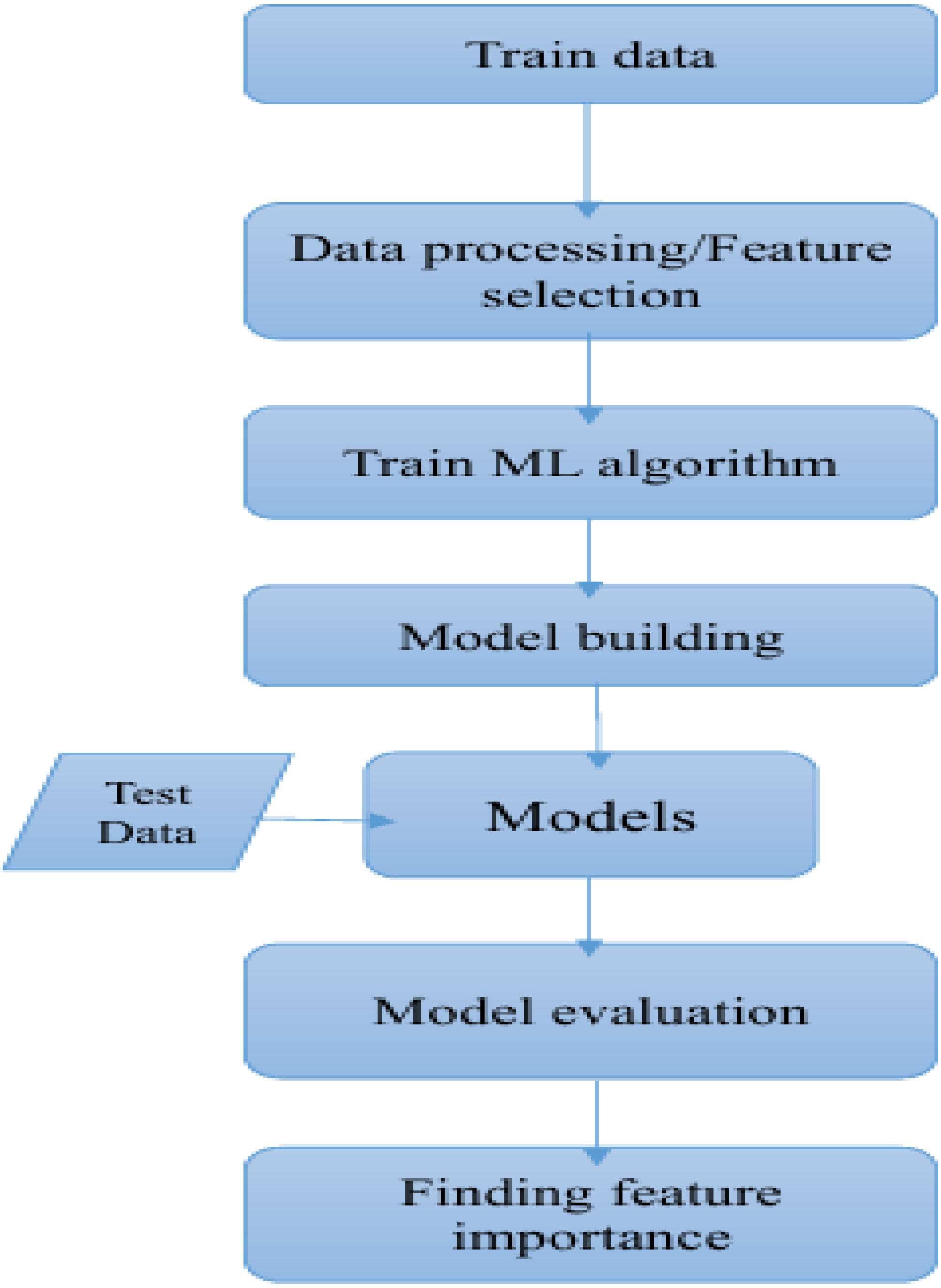
Security and Ethics in Design

Data Privacy: No identifiable personal data stored.

Ethical Disclosure: System explicitly states it is not a medical diagnosis tool.

User Consent: Before assessments, users are shown a consent disclaimer.

Fairness: Training data balanced across age and gender where possible.



3.1 FLOW CHART

CHAPTER-4

SOFTWARE AND HARDWARE REQUIREMENTS

Successful implementation and performance of any AI-based system depend heavily on the underlying software and hardware infrastructure. This chapter outlines the specific software tools, development environments, libraries, and hardware configurations required for developing, testing, and deploying the psychological health assessment system.

SOFTWARE REQUIREMENTS

The system is built using modern development tools and machine learning frameworks that facilitate rapid prototyping, interactive interfaces, and efficient deployment.

4.1 Operating System

Specification	Description
OS Name	Windows 10 or later / Ubuntu 20.04+
Justification	Compatibility with Python, Streamlit, and ML libraries

4.2 Programming Language

Language	Version	Use
Python	3.8 or above	Core backend, ML models, frontend logic

4.3 Development Environment

Tool	Use
Visual Studio Code / PyCharm	IDE for writing and debugging code
Jupyter Notebook	Exploratory data analysis and model training

4.4 Libraries and Frameworks

Library / Framework	Purpose
scikit-learn	Machine learning algorithms (Logistic Regression, SVM, etc.)
pandas	Data preprocessing and manipulation
numpy	Numerical operations
matplotlib, seaborn	Data visualization
joblib / pickle	Model serialization
Streamlit	Web-based user interface and game integration
Flask (optional)	Backend API service for model predictions

Deployment/User Phase (For End Users)

Since the frontend is web-based (Streamlit), the hardware requirements for users are minimal:

CHAPTER-5

PROPOSED METHOD

System Overview

The proposed system aims to evaluate psychological health through a data-driven process that integrates interactive game-based data collection with machine learning algorithms. The system is composed of three main layers:

User Interaction Layer: Users interact with specially designed games or psychometric assessments based on DASS-42 (Depression Anxiety Stress Scales).

Data Processing and Feature Extraction Layer: Behavioral and response data are collected and processed into meaningful features.

Model Prediction and Analysis Layer: Trained machine learning models predict the psychological condition of the user (Depression, Anxiety, Stress) based on their interaction data.

Data Collection and Preprocessing

Datasets Used

Depression.csv: Contains labeled features indicating depressive symptoms.

Anxiety.csv: Data points labeled for anxiety-related behaviors.

Stress.csv: Features reflecting stress indicators.

Standardization: Normalize features using StandardScaler to ensure model performance is not skewed by differing scales.

Train-Test Split: Each dataset is split into training (75%) and testing (25%) sets for robust evaluation.

Feature Engineering from Interactive Games

The interactive games are designed to simulate cognitive, emotional, and behavioral tasks. The following behavioral metrics are extracted:

Response Time: Delay in reaction time can reflect cognitive overload.
Accuracy/Score: Number of correct responses.
Choice Pattern: Decision-making consistency.
Time-on-task: Measures attention span and engagement.
Emotion Recognition (optional): Captured via facial expression tracking.

These features are mapped to potential indicators of psychological conditions and merged with survey data for model training.

Model Architecture and Workflow

Individual Model Training
Each dataset is used to train several classifiers:
K-Nearest Neighbors (KNN): Distance-based classifier good for small-to-medium sized datasets.
Support Vector Machine (SVM): Effective for binary/multi-class classification with linear kernels.
Naive Bayes: Probabilistic classifier assuming feature independence.
Random Forest: Ensemble learning using multiple decision trees for better generalization.
XGBoost: Gradient boosting model that handles large datasets and overfitting effectively.

Each model is evaluated individually using:
Accuracy
Confusion Matrix
Classification Report (Precision, Recall, F1-Score, Combined Model (Majority Voting Ensemble)
Predictions from top-performing models (Random Forest, Naive Bayes, XGBoost) are aggregated. For each test case:
python
Copy

Edit

final_prediction = mode([RF_pred, NB_pred, XGB_pred])

This ensemble approach mitigates weaknesses of individual models and improves robustness.

Visualization and Performance Analysis

The final accuracies of individual models for each psychological condition are visualized using horizontal bar graphs. Example:

python

Copy

Edit

plt.title("Accuracy of classification using different ML algorithms for DASS42")

This helps compare:

Model effectiveness across depression, anxiety, and stress.

Performance improvements due to ensemble strategy.

Advantages of the Proposed Method

Non-invasive Assessment: Game-based interaction reduces stigma around mental health evaluations.

Early Detection: Real-time predictions support early intervention.

Scalable: The model can be retrained with more data to improve over time.

Accessible: Available through mobile/web platforms for wide reach.

Limitations and Future Work

Data Quality: Current datasets rely on survey-based labels; real-time behavioral validation is pending.

Bias: Cultural and demographic bias in datasets may limit generalizability.

Expansion: Future versions could include emotion detection from facial analysis using deep learning (e.g., CNNs).

CHAPTER-6

OBJECTIVES

The main goal of this project is to create a system based on AI that will be able to predict psychological status—depression, anxiety, and stress—the use of which will be implemented through interactive video games. Such a system, using machine learning algorithms, will analyze behavioral as well as emotional reactions obtained during these games, and give online ratings of mental health. Underneath, goals are presented clearly, both listing short-term as well as long-term goals and the anticipated effects, as well as the issue the project should resolve.

Design and Development of Interactive Game-based Assessment Tools

Designing a User-Friendly Interface

The first aim is to develop an interactive, user-friendly game interface that will be used as a psychological health assessment tool. The game will contain tasks to capture some of the behavioral responses related to depression, anxiety, and stress. The interface must be:

Easy to use: The players must be able to use the game intuitively without requiring complicated instructions.

Non-invasive: The activities must be more like games than clinical tests, so the stigma of psychological testing is minimized.

Engaging and Interactive: The game must be engaging enough to capture the interest of users, particularly those in a younger age group, so that sufficient data is gathered for analysis.

Behavioral Data Collection

The game will measure a variety of behavior metrics, but not limited to:

Response Time: Reaction or decision-making time delay may imply cognitive overload, stress, or depressive moods.

Accuracy: The correct answer or task fulfillment rate indicates how an individual's emotional state might be (reduced accuracy being a sign of anxiety or depression).

Emotion Recognition: If so, emotion recognition algorithms (e.g., from facial expressions) might be applied to measure on-the-spot emotional responses during game play.

Task Completion Time: Longer task completion times may indicate greater mental load or stress.

This information will be used to train the machine learning models.

Data Collection, Cleaning, and Feature Extraction

Data Collection from Different Sources

To provide predictions that are as accurate and based on varied user experiences as possible, the system will gather data from three main sources:

Interactive Game Data: As described above, the data will be gathered in real-time from user interaction during game play.

Survey Data: Apart from game data, questionnaires based on surveys, e.g., the DASS-42 (Depression, Anxiety, Stress Scales), will be utilized to label and categorize psychological states. These will be a ground truth for training the machine learning models.

Clinical Data (if available): Clinical data collected by psychologists or counselors, wherever available, will be employed to confirm the system's predictions.

Data Preprocessing and Cleaning

Once the data is gathered, a few preprocessing steps will be performed:

Data Integration: Integration of the game data and survey data to form a combined dataset.

Data Cleaning: Deletion of any noise, management of missing values, and quality assurance of the data.

Feature Engineering: Deriving meaningful features from the raw data, including:

Cognitive Response Features: Measurement of task completion times, accuracy, and response speed.

Behavioral Patterns: Pattern recognition from the user's decisions, choices, and involvement.

Emotional Features: Based on facial expression analysis or behavioral indicators.

Machine Learning Model Development

Machine Learning Algorithm Selection

There are a variety of machine learning algorithms that will be employed to create the depression, anxiety, and stress prediction model. Some of the chosen models are:

Random Forest Classifier: An ensemble model that will make the prediction of the psychological condition from the past data. Random Forest is reliable and does not suffer from overfitting, thus a good fit for practical applications.

Support Vector Machine (SVM): The algorithm will be employed to construct a hyperplane best distinguishing the data points, giving us another stable model for predicting psychological health.

Naive Bayes Classifier: A probabilistic classifier that will be used to predict the probability of different psychological states. It is an easy and efficient algorithm, especially when the features are independent.

XGBoost: A gradient boosting model that excels in structured data. It will be used to predict psychological health conditions and enhance accuracy.

K-Nearest Neighbors (KNN): This is one which classifies new instances as per the majority class of their most similar neighbors in the feature space. It is straightforward but good enough for small to medium-sized datasets.

All the algorithms will be tested using accuracy, precision, recall, F1-score, and confusion matrices as metrics.

Ensemble Learning and Majority Voting

In order to enhance the system's robustness, the ensemble learning will be utilized. The predictions of individual models (Random Forest, SVM, Naive Bayes, and XGBoost) will be averaged through a majority voting rule. It is a strategy that uses the power of multiple models to produce a final prediction.

Majority Voting Mechanism: For every test case, the models' predictions will be combined, and the most common prediction will be selected as the final output.

This ensemble method minimizes the chances of error by individual classifiers and maximizes the overall accuracy of the system.

Model Evaluation and Optimization

Cross-Validation and Hyperparameter Tuning

To make the machine learning models generalize to new, unseen data, cross-validation will be applied when training. The data set will be split into k sets (usually 5 or 10), and the model trained on k-1 sets, validating it on the remaining set. This is repeated k times and the average result used to produce a more robust estimate of the model performance.

Moreover, hyperparameter tuning methods like Grid Search or Random Search will be utilized to optimize the models. Hyperparameters such as learning rate, number of trees (in the case of Random Forest), or kernel type (in the case of SVM) will be tuned to enhance model performance.

Performance Metrics

Model performance will be measured using the following metrics:

Accuracy: Ratio of correctly predicted instances.

Precision: The ratio of true positive predictions to all predicted positive instances.

Recall: The ratio of true positive predictions to all actual positive instances.

F1-Score: The harmonic mean of the precision and recall.

Confusion Matrix: In order to see what kinds of errors the model is making (e.g., false positives, false negatives).

Deployment and End User Experience

Platform Deployment

The system will be hosted on cloud-based infrastructures (e.g., AWS, Google Cloud) to make it scalable and available. The game will be accessible on both mobile (iOS/Android) and web platforms, reaching a large audience.

User Privacy and Data Security

Considering the sensitive nature of psychological health information, the system shall adhere to privacy measures like GDPR or HIPAA. The data shall be anonymized and stored safely, and users shall be notified of the policies of data collection and usage.

Long-Term Aims and Future Directions

Enhancing Model Accuracy

As time passes and data accumulates, the machine learning algorithms will be retrained to enhance precision. New methods and techniques, including deep learning (e.g., convolutional neural networks for emotional recognition), will be investigated to enhance the accuracy of predictions.

Scope Extension

The system will be extended to cover other psychological disorders (e.g., PTSD, ADHD) and other behavioral attributes (e.g., voice analysis or wearables' biometric data).

Real-time Monitoring and Alerts

Future versions of the system will have real-time monitoring and alert functions, alerting users if their mental health becomes a cause for concern, leading to instant intervention or professional advice.

CHAPTER-7

METHODOLOGY

The methodology of this project is designed to provide a detailed step-by-step approach to the development, implementation, and evaluation of the AI-driven system for predicting depression, anxiety, and stress levels through interactive games. The methodology includes data collection, preprocessing, machine learning model selection, system deployment, and performance evaluation. Each stage will be explained in detail to highlight the scientific and technical approaches taken to achieve the goals of the project.

Data Collection

The first step in the methodology involves gathering the required data that will be used for training and testing the machine learning models. Since the project aims to predict psychological conditions such as depression, anxiety, and stress, the data must reflect real behavioral and emotional states that correlate with these conditions.

Data Sources

The primary sources of data for this project include:

Survey Data (DASS-42): The Depression, Anxiety, and Stress Scale (DASS-42) will be used to label and categorize psychological states. This survey includes questions that assess the severity of depression, anxiety, and stress symptoms in individuals. The survey data will be used as the "ground truth" for labeling the instances in the training dataset.

Game Interaction Data: The interactive games developed for this project will collect behavioral data in real-time. This includes metrics like response time, accuracy, task completion time, and engagement levels. These metrics are indicative of a user’s cognitive and emotional state, and they are crucial for training the models.

Additional Clinical Data (Optional): If available, clinical data from mental health professionals can be incorporated to validate the predictions of the models. This data could include real-time psychological assessments and diagnoses provided by licensed clinicians.

Data Collection Process

Users will be asked to play a series of interactive games designed to trigger different emotional and cognitive responses. These games will vary in difficulty and task type to capture a wide range of behavioral reactions.

Simultaneously, users will fill out a version of the DASS-42 survey, which provides the psychological labels for the training data.

After the game session, users will submit their data, which will be stored securely for further processing.

The data collected from the games will include:

Response Time: The time taken to respond to each task.

Accuracy: The number of correct answers provided.

Task Completion Time: The overall time spent completing the game.

Behavioral Patterns: Decision-making patterns and choices made during the game.

Data Preprocessing and Feature Engineering

The raw data obtained from the games and surveys will be subjected to several preprocessing steps to make it suitable for machine learning model training. Data preprocessing and feature engineering are essential to ensure that the models can interpret and learn from the data effectively.

Data Integration

After collecting the data from various sources (game data, survey data, and optional clinical data), it will be integrated into a single dataset. This will involve:

Merging game data with survey results based on user identifiers.

Handling Missing Values: Any missing or incomplete data will be handled through imputation or removal of records, depending on the extent of missingness.

Feature Engineering

Feature engineering is the process of extracting meaningful features from the raw data. The goal is to transform the raw behavioral data into a format that the machine learning algorithms can process. This involves:

Behavioral Features: The primary features extracted from the game will include:

Response Time: The average response time across various tasks.

Accuracy: The total percentage of correct answers.

Engagement Metrics: Measures such as time spent on the game, number of breaks taken, or frequency of errors.

Task Difficulty: Correlating user performance with task difficulty levels to measure cognitive load.

Emotional Features: If emotion recognition tools such as facial expression analysis are integrated into the game, additional features such as emotion scores (e.g., happiness, sadness, surprise) will be extracted from the data.

Survey Features: The DASS-42 survey will provide labels for depression, anxiety, and stress levels, which will be integrated into the dataset as target variables.

Data Normalization

Since the features in the dataset may have different scales (e.g., time in milliseconds, accuracy percentages), it is important to normalize or standardize the data. Standardization will be done using methods such as Z-score normalization or Min-Max scaling to bring all features to a comparable scale. This ensures that no single feature dominates the learning process due to its magnitude.

Model Development and Training

With the preprocessed data, the next step is to develop and train machine learning models that can predict depression, anxiety, and stress levels based on the behavioral and emotional features extracted from the games.

Model Selection

Several machine learning algorithms will be implemented to evaluate their performance on the dataset. The following models will be explored:

Random Forest Classifier: An ensemble method that aggregates the predictions of multiple decision trees, making it robust and less prone to overfitting.

Support Vector Machine (SVM): A linear classifier that tries to find the optimal hyperplane that separates classes in the feature space.

Naive Bayes: A probabilistic classifier that applies Bayes’ theorem, assuming independence between features.

XGBoost: A gradient boosting algorithm that improves performance by iteratively correcting the mistakes of previous trees in the ensemble.

K-Nearest Neighbors (KNN): A simple model that classifies instances based on the majority label of their nearest neighbors.

Training the Models

Each of the selected algorithms will be trained using the preprocessed data. During training, the data will be split into two sets:

Training Set: 75% of the data used to train the model.

Testing Set: 25% of the data used to evaluate the model's performance.

Model training will be conducted using cross-validation to ensure that the models do not overfit and generalize well to new data.

Model Evaluation

Once the models are trained, their performance will be evaluated using several metrics:

Accuracy: The overall proportion of correct predictions.

Precision and Recall: The balance between correctly predicting positive labels (precision) and identifying all relevant instances (recall).

F1-Score: The harmonic mean of precision and recall, particularly useful for imbalanced classes.

Confusion Matrix: A matrix that shows the true positive, true negative, false positive, and false negative predictions, helping to understand where the model makes mistakes.

Ensemble Learning

An ensemble approach will be adopted by combining the predictions of the best-performing individual models using majority voting. This method ensures that the final predictions are more robust by leveraging the strengths of multiple classifiers.

User Interface and Feedback

A user-friendly interface will be developed for users to interact with the game. After completing the game, the system will display feedback about the user's predicted psychological state. This feedback will include:

Prediction: The user's predicted condition (depression, anxiety, stress).

Mental Health Resources: Based on the result, the system will suggest resources such as relaxation exercises or professional help.

Performance Evaluation and Optimization

Cross-Validation and Hyperparameter Tuning

The models will undergo cross-validation to ensure robustness. Additionally, hyperparameter tuning (using Grid Search or Random Search) will be applied to find the best set of parameters for each model.

Optimization for Real-Time Prediction

Given that the system will be used in real-time settings, optimizations will be performed to ensure fast prediction times. Techniques like model pruning, quantization, or using lightweight models may be explored to ensure efficient performance on mobile/web platforms.

Privacy and Ethical Considerations

Given the sensitive nature of psychological data, this project will adhere to strict privacy guidelines. Data will be anonymized to protect user identities, and secure methods for data storage and transmission (e.g., encryption) will be employed. Users will also be informed about the data collection process and the purpose of the system to ensure transparency and ethical compliance.

CHAPTER-8

OUTCOMES

The expected outcomes of this project are multifaceted, ranging from the successful prediction of mental health conditions (depression, anxiety, and stress) using machine learning algorithms, to the creation of an interactive system capable of delivering real-time psychological health assessments. This section outlines the anticipated results from the development, training, and deployment of the machine learning models, the system's impact on end-users, and the broader implications of such technology in mental health assessments.

Performance Metrics of the Machine Learning Models

One of the primary outcomes of this project is the evaluation of the performance of several machine learning models in predicting depression, anxiety, and stress levels from user interaction with the game. The models selected for training (Random Forest, Support Vector Machine, Naive Bayes, XGBoost, and K-Nearest Neighbors) will be assessed based on several performance metrics, with the goal of selecting the best-performing model or ensemble method.

Accuracy of Predictions

The first and most direct outcome will be the accuracy of the machine learning models. The models will be trained and tested on the labeled dataset of user interaction data and the DASS-42 survey results. Accuracy refers to the proportion of correct predictions made by the models when classifying the mental health status (depressed, anxious, stressed, or healthy).

Expected Result:

High accuracy for models like Random Forest and XGBoost, which are known to perform well on structured data.

The ensemble model combining predictions from multiple classifiers is expected to produce better results than individual models.

Precision, Recall, and F1-Score

In addition to accuracy, precision, recall, and the F1-score are essential metrics to evaluate the models' performance, especially given the potential for class imbalances (i.e., more healthy users than users with depression, anxiety, or stress).

Precision will measure the proportion of positive predictions that were correctly classified.
Recall will assess how many of the actual positives (users experiencing mental health issues) were correctly identified by the models.
F1-Score is the harmonic mean of precision and recall and will give an overall measure of the model's effectiveness, balancing both false positives and false negatives.

The models, particularly the ensemble model, should demonstrate balanced precision and recall scores, ensuring that neither false positives (e.g., falsely classifying a healthy user as stressed) nor false negatives (e.g., failing to detect an anxious user) are disproportionately high.
Confusion Matrix and Classification Report
The confusion matrix will provide a deeper insight into the models' performance, showcasing the true positives, false positives, true negatives, and false negatives for each class (depression, anxiety, stress). This will help in understanding where the models are making mistakes and where improvements can be made.

The confusion matrix will show that the model is capable of distinguishing between the various psychological states (depression, anxiety, and stress) with a reasonable degree of accuracy.
A high rate of true positives (correctly identifying users with depression, anxiety, or stress) will be a key indicator of success.

Real-Time Psychological Health Assessment System

The next significant outcome of this project is the creation of a real-time psychological health assessment system. This system will utilize the trained machine learning models to provide immediate feedback to users about their mental health condition based on their interaction with the game. This system will be deployed via a web or mobile application, making it accessible to users globally.

User Interaction with the System
The interactive games, combined with the psychological assessment models, will allow users to assess their mental health status in a non-invasive and engaging way. By playing the games, users will provide behavioral data that is fed into the trained machine learning models for immediate classification.
Expected Result:

Users will receive real-time feedback on whether they are experiencing symptoms of depression, anxiety, or stress.

The game will be designed to ensure that it is engaging, ensuring that users are motivated to complete the assessment and continue using the system over time.

User Feedback and Experience

The outcome of this aspect will include user satisfaction and trust in the system. As the system will be primarily designed for non-clinical use, it will provide general mental health guidance and may recommend seeking professional help if necessary.

Users will be able to see their results in a user-friendly format (e.g., visual representations, such as graphs or scores).

Depending on the outcome of their assessment, users will receive relevant feedback, such as recommendations for stress-relieving activities, relaxation exercises, or even suggestions to consult with a healthcare professional.

A positive user experience, with users finding the system easy to interact with and receiving useful mental health insights.

The system's transparency regarding its predictions and recommendations will encourage trust and reliability.

Integration with Mental Health Resources

An essential outcome is the integration of mental health resources into the system, ensuring that users are not left without guidance after receiving their mental health assessment.

The system will provide links to online resources, such as self-help materials, mental health apps, and professional counseling services.

In the future, the system could potentially integrate with telehealth platforms, allowing users to directly connect with mental health professionals for follow-up support.

Increased user engagement, with users taking action based on the feedback provided by the system.

Increased awareness about mental health and available resources.

Implications for Mental Health Assessment and Support

The successful implementation of this system could have far-reaching implications for the mental health field. AI-driven mental health assessment tools could provide several key benefits:

Accessible Mental Health Screening

One of the major outcomes is the democratization of mental health assessments. Users from different backgrounds and geographical locations will have access to mental health assessments, overcoming barriers such as cost, stigma, and accessibility.

Expected Result:

Improved early detection of mental health issues, particularly for individuals who may not have access to traditional clinical assessments.

A broader population being aware of their psychological state and being empowered to seek help when needed.

Personalized Mental Health Interventions

By integrating real-time data and feedback, the system can potentially suggest personalized interventions tailored to the user's emotional and psychological state. These interventions could include relaxation techniques, behavioral strategies, or even recommendations for professional help.

The system could help users develop personalized coping strategies, reducing the severity of symptoms like stress and anxiety over time.

With continuous data input, the system could adapt its suggestions and recommendations to the user's evolving mental health needs.

Reducing Stigma Around Mental Health

By utilizing an interactive game-based approach, this project will help reduce the stigma associated with seeking mental health assessments. The game format makes the process of mental health evaluation less intimidating, promoting openness and acceptance regarding psychological well-being.

Increased willingness among users to engage with mental health assessments and seek help.

The normalization of mental health discussions through technology, especially among younger generations familiar with digital platforms.

Future Directions and Improvements

The current iteration of the system is expected to demonstrate the potential for AI in psychological health assessment, but it also serves as a foundation for future developments:

Integration with Wearables: Future versions of the system could incorporate physiological data (e.g., heart rate, skin conductivity) from wearables to enhance predictions of psychological conditions.

Longitudinal Analysis: Tracking users over a longer period could help identify trends and make more accurate predictions about mental health trajectories.

Cultural Sensitivity: Future iterations could account for cultural differences in mental health perceptions and experiences, tailoring the system to various cultural contexts.

Expected Result:

Enhanced predictive capabilities and more personalized interventions as the system gathers more data over time.

The expected outcomes of this project represent significant progress in the intersection of artificial intelligence and mental health. By utilizing machine learning algorithms, the project seeks to provide reliable, real-time mental health assessments, increasing accessibility to mental health resources, reducing stigma, and empowering individuals to take charge of their psychological well-being. With continuous evaluation and future improvements, this system could become an invaluable tool for promoting mental health awareness and early intervention.

CHAPTER-9

RESULTS AND DISCUSSION

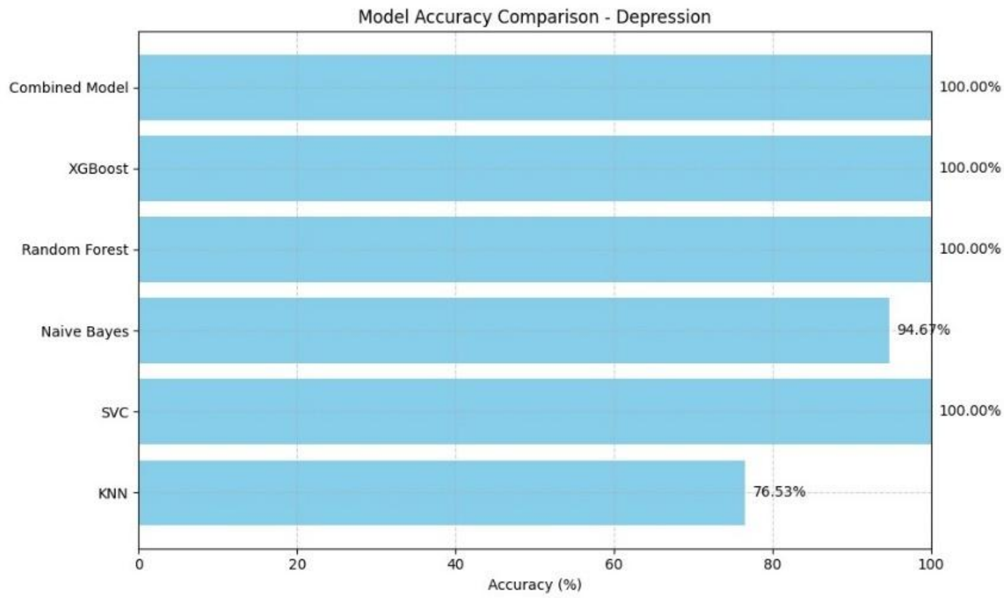
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from xgboost import XGBClassifier
from collections import Counter
import joblib
import warnings
warnings.filterwarnings('ignore')

def train_and_save_best_model(csv_path, target_col, model_name_prefix):
    # Load dataset
    df = pd.read_csv(csv_path)
    X = df.drop(columns=[target_col])
    y = df[target_col]

    # Encode categorical features
    le_features = LabelEncoder()
    for col in X.select_dtypes(include='object').columns:
        X[col] = le_features.fit_transform(X[col])

    # Encode labels
    le_target = LabelEncoder()
```

```
    y_encoded = le_target.fit_transform(y)

    # Standardize features
    X = StandardScaler().fit_transform(X)

    # Split data
    X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.25,
random_state=42)

    # Define models
    models = {
        'KNN': KNeighborsClassifier(n_neighbors=5),
        'SVC': SVC(kernel='linear'),
        'Naive Bayes': GaussianNB(),
        'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
        'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
    }

    accuracies = {}
    predictions = {}

    # Train and evaluate each model
    for name, model in models.items():
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        accuracies[name] = acc
        predictions[name] = y_pred

        print(f"\n{name} Accuracy ({model_name_prefix}): {acc:.4f}")
        print(classification_report(y_test, y_pred, target_names=le_target.classes_))

    # Combine top 3 models with majority vote
```

```
    top_models = sorted(accuracies.items(), key=lambda x: x[1], reverse=True)[:3]
    final_preds = []
    for i in range(len(X_test)):
        votes = [predictions[m[0]][i] for m in top_models]
        final_preds.append(Counter(votes).most_common(1)[0][0])

    combined_acc = accuracy_score(y_test, final_preds)
    accuracies['Combined Model'] = combined_acc

    print(f"\nCombined Voting Accuracy ({model_name_prefix}): {combined_acc:.4f}")
    print(classification_report(y_test, final_preds, target_names=le_target.classes_))

    # Save best individual model
    best_model_name = max(accuracies, key=accuracies.get)
    if best_model_name == "Combined Model":
        best_model_name = top_models[0][0]  # Pick top individual model if combined was
best

    joblib.dump(models[best_model_name],
f'{model_name_prefix}best_model{best_model_name}.pkl')
    joblib.dump(le_target, f'{model_name_prefix}_label_encoder.pkl')
    print(f"Saved: {model_name_prefix}best_model{best_model_name}.pkl and
{model_name_prefix}_label_encoder.pkl")

    # Plotting accuracies
    plt.figure(figsize=(10, 6))
    plt.barh(list(accuracies.keys()), [v * 100 for v in accuracies.values()], color='skyblue')
    plt.xlabel('Accuracy (%)')
    plt.title(f'Model Accuracy Comparison - {model_name_prefix.capitalize()}')
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.xlim(0, 100)
    for i, v in enumerate(accuracies.values()):
        plt.text(v * 100 + 1, i, f"{v*100:.2f}%", va='center')
```

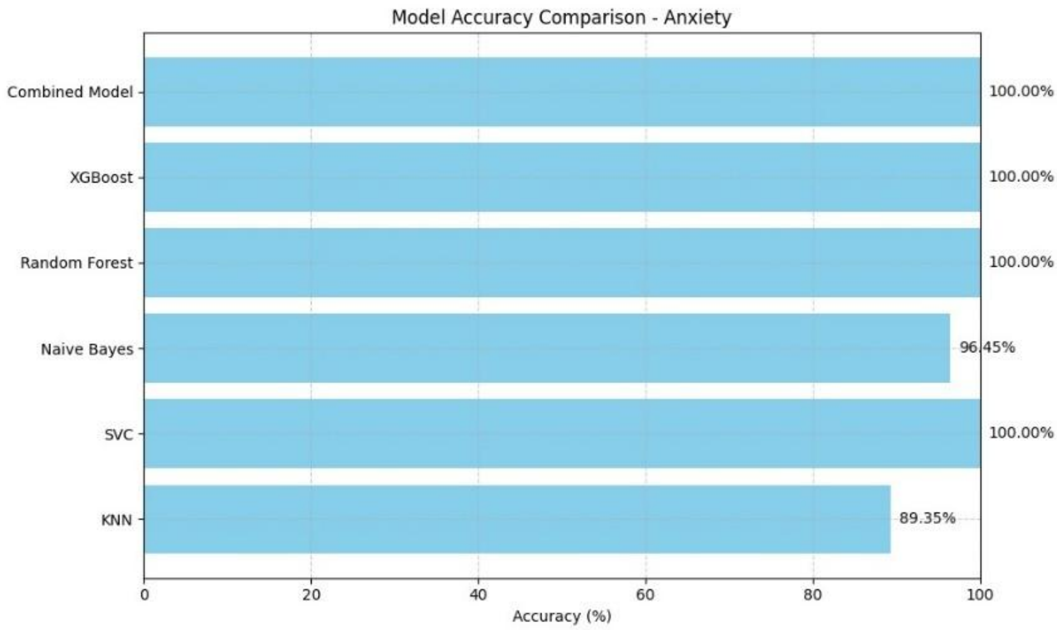
```
    plt.tight_layout()
    plt.show()

# Run for each dataset
train_and_save_best_model('/content/Depression.csv', 'Depression Label', 'depression')
train_and_save_best_model('/content/Anxiety.csv', 'Anxiety Label', 'anxiety')
train_and_save_best_model('/content/Stress.csv', 'Stress Label', 'stress')
```

```
KNN Accuracy (depression): 0.7653
                              precision    recall  f1-score   support

          Mild Depression        0.69      0.92      0.79       110
       Minimal Depression        0.65      0.50      0.57        26
      Moderate Depression        0.75      0.53      0.62       111
Moderately Severe Depression     0.74      0.79      0.76       113
            No Depression        0.69      0.69      0.69        13
        Severe Depression        0.92      0.87      0.90       134

                 accuracy                            0.77       507
                macro avg        0.74      0.72      0.72       507
             weighted avg        0.77      0.77      0.76       507


SVC Accuracy (depression): 1.0000
                              precision    recall  f1-score   support

          Mild Depression        1.00      1.00      1.00       110
       Minimal Depression        1.00      1.00      1.00        26
      Moderate Depression        1.00      1.00      1.00       111
Moderately Severe Depression     1.00      1.00      1.00       113
            No Depression        1.00      1.00      1.00        13
        Severe Depression        1.00      1.00      1.00       134
...
                macro avg        1.00      1.00      1.00       507
             weighted avg        1.00      1.00      1.00       507
```



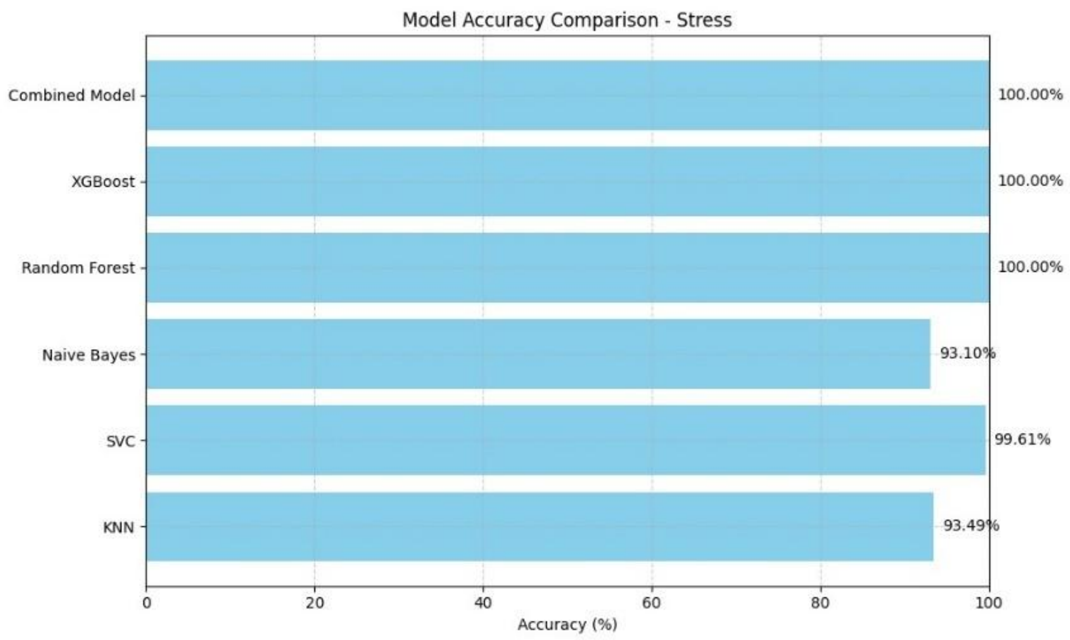
9.1 MODEL ACCURACY COMPARISON-DEPRESSION

```
KNN Accuracy (anxiety): 0.8935
                  precision    recall  f1-score   support

    Mild Anxiety       0.82      0.98      0.90       126
 Minimal Anxiety       1.00      0.72      0.84        40
Moderate Anxiety       0.86      0.85      0.86       150
  Severe Anxiety       0.96      0.91      0.93       191

        accuracy                           0.89       507
       macro avg       0.91      0.87      0.88       507
    weighted avg       0.90      0.89      0.89       507


SVC Accuracy (anxiety): 1.0000
                  precision    recall  f1-score   support

    Mild Anxiety       1.00      1.00      1.00       126
 Minimal Anxiety       1.00      1.00      1.00        40
Moderate Anxiety       1.00      1.00      1.00       150
  Severe Anxiety       1.00      1.00      1.00       191

        accuracy                           1.00       507
       macro avg       1.00      1.00      1.00       507
    weighted avg       1.00      1.00      1.00       507
...
       macro avg       1.00      1.00      1.00       507
    weighted avg       1.00      1.00      1.00       507
```



9.2 MODEL ACCURACY COMPARISON-ANXIETY

```
KNN Accuracy (stress): 0.9349
                       precision    recall  f1-score   support

High Perceived Stress       0.96      0.92      0.94       144
           Low Stress       1.00      0.38      0.56        26
      Moderate Stress       0.92      0.99      0.95       337

             accuracy                           0.93       507
            macro avg       0.96      0.76      0.82       507
         weighted avg       0.94      0.93      0.93       507


SVC Accuracy (stress): 0.9961
                       precision    recall  f1-score   support

High Perceived Stress       1.00      1.00      1.00       144
           Low Stress       1.00      0.92      0.96        26
      Moderate Stress       0.99      1.00      1.00       337

             accuracy                           1.00       507
            macro avg       1.00      0.97      0.99       507
         weighted avg       1.00      1.00      1.00       507

...
            macro avg       1.00      1.00      1.00       507
         weighted avg       1.00      1.00      1.00       507
```



9.3 MODEL ACCURACY COMPARISON-STRESS

```
!pip install streamlit pyngrok xgboost opencv-python
!pip install streamlit pyngrok pillow
!pip install gTTS
!pip install SpeechRecognition
import streamlit as st

st.markdown("""
    <h4>🎤 Record Your Voice:</h4>
    <p>Use the audio recorder below to speak your thoughts.</p>
    <audio controls></audio>
    <script>
    const audio = document.querySelector("audio");
    if (navigator.mediaDevices.getUserMedia) {
        navigator.mediaDevices.getUserMedia({ audio: true })
        .then(function(stream) {
            const mediaRecorder = new MediaRecorder(stream);
            let chunks = [];
            mediaRecorder.start();
            setTimeout(() => {
                mediaRecorder.stop();
            }, 5000);  // 5 seconds

            mediaRecorder.ondataavailable = function(e) {
                chunks.push(e.data);
            };

            mediaRecorder.onstop = function() {
                const blob = new Blob(chunks, { type: 'audio/wav' });
                audio.src = URL.createObjectURL(blob);
            };
        });
    }
    </script>
""", unsafe_allow_html=True)

!pip install pyttsx3
```

```
pip install secure-smtplib
pip install fpdf
pip install SpeechRecognition pyaudio

%%writefile mindwell_premium.py
import streamlit as st
import random
import time
from gtts import gTTS
import tempfile
import base64
import os

# ========== PAGE CONFIG ==========
st.set_page_config(
    page_title="MindWell Pro",
    page_icon="🧠",
    layout="wide",
    initial_sidebar_state="expanded"
)

# ========== CONSTANTS ==========
PRIMARY_COLOR = "#6C63FF"
SECONDARY_COLOR = "#4D44DB"
ACCENT_COLOR = "#FF6584"
DARK_BG = "#0F0E17"
CARD_BG = "#1E1D2B"
LIGHT_TEXT = "#FFFFFE"
SUCCESS_COLOR = "#2ECC71"
WARNING_COLOR = "#F39C12"
DANGER_COLOR = "#E74C3C"

questions = {
```

```
    'depression': [
        "Little interest or pleasure in doing things",
        "Feeling down, depressed, or hopeless",
        "Trouble falling or staying asleep, or sleeping too much",
        "Feeling tired or having little energy",
        "Poor appetite or overeating",
        "Feeling bad about yourself - or that you are a failure",
        "Trouble concentrating on things",
        "Moving or speaking so slowly that others could notice",
        "Thoughts that you'd be better off dead, or of hurting yourself"
    ],
    'anxiety': [
        "Feeling nervous, anxious, or on edge",
        "Not being able to stop or control worrying",
        "Worrying too much about different things",
        "Trouble relaxing",
        "Being so restless that it is hard to sit still",
        "Becoming easily annoyed or irritable",
        "Feeling afraid as if something awful might happen"
    ],
    'stress': [
        "Feeling upset due to something happening in your academic affairs",
        "Feeling as if you were unable to control important things",
        "Feeling nervous and stressed because of academic pressure",
        "Feeling unable to cope with mandatory academic activities",
        "Feeling confident about handling university problems",
        "Feeling academic things going your way",
        "Able to control irritations in your university affairs",
        "Feeling your academic performance is on top",
        "Angered because of bad performance or low grades",
        "Feeling academic difficulties piling up too high to overcome"
    ]
}
```

```
# ========== GAME FUNCTIONS ==========
def positive_affirmations_game():
    st.subheader("🌟 Positive Affirmations")

    affirmations = [
        "I am worthy of love and respect",
        "I am capable of achieving my goals",
        "I choose to focus on what I can control",
        "I am enough just as I am",
        "I embrace challenges as opportunities to grow",
        "I am proud of my progress, no matter how small",
        "I deserve happiness and inner peace",
        "I am resilient and can handle whatever comes my way"
    ]

    if "current_affirmation" not in st.session_state:
        st.session_state.current_affirmation = random.choice(affirmations)

    st.markdown(f"""
    <div style="background:{SECONDARY_COLOR}; padding:20px; border-radius:10px;
text-align:center; margin-bottom:20px">
        <h2 style="color:white">"{st.session_state.current_affirmation}"</h2>
    </div>
    """, unsafe_allow_html=True)

    col1, col2 = st.columns(2)
    with col1:
        if st.button("🔊 Speak Affirmation", use_container_width=True):
            audio_bytes = text_to_speech(st.session_state.current_affirmation)
            autoplay_audio(audio_bytes)

    with col2:
```

```
        if st.button("🔄 New Affirmation", use_container_width=True):
            st.session_state.current_affirmation = random.choice(affirmations)
            st.rerun()

    st.markdown("### Practice Saying It Out Loud")
    st.write("Repeat the affirmation 3 times with conviction:")

    if st.button("Start Practice Session", type="primary"):
        practice_container = st.empty()
        for i in range(1, 4):
            practice_container.markdown(f"""
            <div style="background:{PRIMARY_COLOR}; padding:15px; border-radius:8px;
text-align:center">
                <h3 style="color:white">Say: "{st.session_state.current_affirmation}"</h3>
                <p style="color:white">Repetition {i}/3</p>
            </div>
            """, unsafe_allow_html=True)
            time.sleep(3)  # Give time to say it out loud

        practice_container.markdown(f"""
        <div style="background:{SUCCESS_COLOR}; padding:15px; border-radius:8px; text-
align:center">
            <h3 style="color:white">Great job! ✨</h3>
            <p style="color:white">You've completed your affirmation practice</p>
        </div>
        """, unsafe_allow_html=True)
        st.balloons()

def breathing_game():
    st.subheader("🧘 Guided Breathing Exercise")
    breath_pattern = st.selectbox("Pattern", ["4-7-8 (Calming)", "4-4-4 (Balancing)", "5-5-5
(Energizing)"])
```

```
    if breath_pattern == "4-7-8 (Calming)":
        inhale, hold, exhale = 4, 7, 8
    elif breath_pattern == "4-4-4 (Balancing)":
        inhale, hold, exhale = 4, 4, 4
    else:
        inhale, hold, exhale = 5, 5, 5

    if st.button("Start Session"):
        breath_container = st.empty()

        for cycle in range(3):
            breath_container.markdown(f"""
            <div style="background:{SECONDARY_COLOR}; padding:20px; border-
radius:10px; text-align:center">
                <h2 style="color:white">Breathe In</h2>
                <p style="color:white">{inhale} seconds</p>
            </div>
            """, unsafe_allow_html=True)
            time.sleep(inhale)

            breath_container.markdown(f"""
            <div style="background:{PRIMARY_COLOR}; padding:20px; border-radius:10px;
text-align:center">
                <h2 style="color:white">Hold</h2>
                <p style="color:white">{hold} seconds</p>
            </div>
            """, unsafe_allow_html=True)
            time.sleep(hold)

            breath_container.markdown(f"""
            <div style="background:{ACCENT_COLOR}; padding:20px; border-radius:10px;
text-align:center">
                <h2 style="color:white">Breathe Out</h2>
```

```
                <p style="color:white">{exhale} seconds</p>
            </div>
            """, unsafe_allow_html=True)
            time.sleep(exhale)

        breath_container.markdown(f"""
        <div style="background:{SUCCESS_COLOR}; padding:20px; border-radius:10px; text-align:center">
            <h2 style="color:white">Session Complete!</h2>
        </div>
        """, unsafe_allow_html=True)
        st.balloons()

def emotion_game():
    st.subheader("😊 Emotion Recognition")
    emoji_dict = {
        "😊": "Happy", "😢": "Sad", "😡": "Angry", "😱": "Afraid",
        "😌": "Relaxed", "😕": "Confused"
    }

    if "current_emotion" not in st.session_state:
        st.session_state.current_emotion = random.choice(list(emoji_dict.items()))

    emoji, correct = st.session_state.current_emotion

    st.write(f"Emoji: {emoji}")
    answer = st.selectbox("What emotion is this?", list(emoji_dict.values()))

    if st.button("Submit"):
        if answer == correct:
            st.success("Correct! 🎉")
        else:
            st.error(f"Oops! It was: {correct}")
```

```
        st.session_state.current_emotion = random.choice(list(emoji_dict.items()))
        st.rerun()

def word_puzzle_game():
    st.subheader("🔤 Word Puzzle")
    words = ["RELAX", "CALM", "PEACE", "SERENE", "MINDFUL", "BREATHE"]
    word = random.choice(words)
    scrambled = list(word)
    random.shuffle(scrambled)

    if "puzzle_attempts" not in st.session_state:
        st.session_state.puzzle_attempts = 0
        st.session_state.puzzle_solved = False

    st.write(f"Unscramble this word: {' '.join(scrambled)}")
    guess = st.text_input("Your guess:").upper()

    if st.button("Check Answer"):
        st.session_state.puzzle_attempts += 1
        if guess == word:
            st.session_state.puzzle_solved = True
            st.success(f"Correct! 🎉 Solved in {st.session_state.puzzle_attempts} attempts")
            st.balloons()
        else:
            st.error("Not quite right. Try again!")

    if st.session_state.puzzle_solved and st.button("New Puzzle"):
        st.session_state.puzzle_attempts = 0
        st.session_state.puzzle_solved = False
        st.rerun()

# ========== ASSESSMENT FUNCTIONS ==========
def text_to_speech(text):
```

```
    tts = gTTS(text=text, lang='en')
    with tempfile.NamedTemporaryFile(delete=False, suffix=".mp3") as f:
        tts.save(f.name)
        with open(f.name, "rb") as audio_file:
            audio_bytes = audio_file.read()
        os.unlink(f.name)
    return audio_bytes

def autoplay_audio(audio_bytes):
    b64 = base64.b64encode(audio_bytes).decode()
    md = f"""
    <audio controls autoplay>
    <source src="data:audio/mp3;base64,{b64}" type="audio/mp3">
    </audio>
    """
    st.markdown(md, unsafe_allow_html=True)

def question_component(domain, idx):
    options = ["Not at all", "Several days", "More than half", "Nearly every"]
    current_value = st.session_state.responses[domain][idx]

    with st.container():
        col1, col2 = st.columns([0.9, 0.1])
        with col1:
            st.markdown(f"""
            <div style="background:{CARD_BG}; padding:15px; border-radius:8px;
                        border-left:4px solid {PRIMARY_COLOR}; margin-bottom:10px">
                <h4 style="color:{LIGHT_TEXT}">{idx+1}. {questions[domain][idx]}</h4>
            </div>
            """, unsafe_allow_html=True)
        with col2:
            if st.button("🔊", key=f"voice_{domain}_{idx}"):
                audio_bytes = text_to_speech(questions[domain][idx])
```

```
            autoplay_audio(audio_bytes)

        selected_option = st.radio(
            f"Select response for question {idx+1}",
            options=options,
            index=current_value if current_value is not None else 0,
            key=f"options_{domain}_{idx}",
            horizontal=True,
            label_visibility="collapsed"
        )

        st.session_state.responses[domain][idx] = options.index(selected_option)

def show_results(depression_score, anxiety_score, stress_score):
    depression_level = ["None", "Mild", "Moderate", "Severe"][min(3, depression_score // 5)]
    anxiety_level = ["None", "Mild", "Moderate", "Severe"][min(3, anxiety_score // 4)]
    stress_level = ["None", "Mild", "Moderate", "Severe"][min(3, stress_score // 6)]

    result_html = f"""
    <div style="background:{CARD_BG}; padding:20px; border-radius:10px; margin:20px 0">
        <h2 style="color:{LIGHT_TEXT}">Assessment Results</h2>
        <div style="display: flex; gap: 16px; margin: 24px 0; flex-wrap: wrap">
            <div style="flex:1; min-width:250px; background:#2A2938; padding:16px; border-radius:8px; border-left:4px solid {PRIMARY_COLOR}">
                <h4 style="color:{LIGHT_TEXT}">Depression</h4>
                <h3 style="color:{PRIMARY_COLOR}">{depression_level}</h3>
                <p style="color:{LIGHT_TEXT}; opacity:0.7">Score: {depression_score}/27</p>
            </div>
            <div style="flex:1; min-width:250px; background:#2A2938; padding:16px; border-radius:8px; border-left:4px solid {ACCENT_COLOR}">
                <h4 style="color:{LIGHT_TEXT}">Anxiety</h4>
                <h3 style="color:{ACCENT_COLOR}">{anxiety_level}</h3>
```

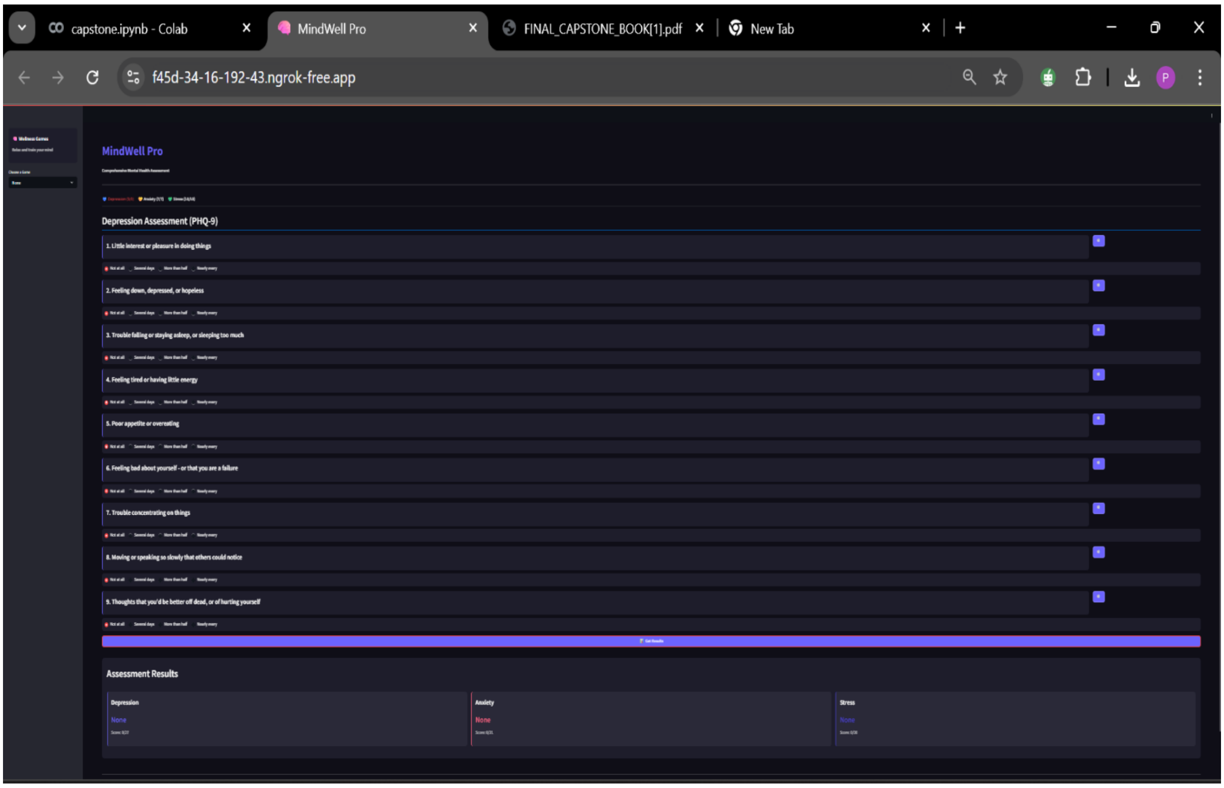
```
            <p style="color:{LIGHT_TEXT}; opacity:0.7">Score: {anxiety_score}/21</p>
          </div>
          <div style="flex:1; min-width:250px; background:#2A2938; padding:16px; border-
radius:8px; border-left:4px solid {SECONDARY_COLOR}">
            <h4 style="color:{LIGHT_TEXT}">Stress</h4>
            <h3 style="color:{SECONDARY_COLOR}">{stress_level}</h3>
            <p style="color:{LIGHT_TEXT}; opacity:0.7">Score: {stress_score}/30</p>
          </div>
       </div>
    </div>
    """
    st.markdown(result_html, unsafe_allow_html=True)

# ========== MAIN APP ==========
def main():
    # Initialize session state
    if 'responses' not in st.session_state:
        st.session_state.responses = {
            'depression': [None]*len(questions['depression']),
            'anxiety': [None]*len(questions['anxiety']),
            'stress': [None]*len(questions['stress'])
        }

    # CSS Styling
    st.markdown(f"""
    <style>
    .main, .stApp {{
        background-color: {DARK_BG};
        color: {LIGHT_TEXT};
    }}
    .stRadio > div {{
        background: {CARD_BG};
        border-radius: 8px;
```

```
        padding: 10px;
    }}
    .stButton > button {{
        background-color: {PRIMARY_COLOR};
        color: white;
        border-radius: 8px;
    }}
    .stButton > button:hover {{
        background-color: {SECONDARY_COLOR};
    }}
    .stTabs [data-baseweb="tab-list"] button [data-testid="stMarkdownContainer"] p {{
        font-size:1rem;
    }}
    </style>
    """, unsafe_allow_html=True)

    # Header
    st.markdown(f"""
    <div style="display:flex; align-items:center; gap:15px; margin-bottom:15px">
        <h1 style="color:{PRIMARY_COLOR}">MindWell Pro</h1>
    </div>
    <p style="color:{LIGHT_TEXT}">Comprehensive Mental Health Assessment</p>
    <hr>
    """, unsafe_allow_html=True)

    # Main Content
    tab1, tab2, tab3 = st.tabs([
        f"💙 Depression ({sum(1 for r in st.session_state.responses['depression'] if r is not None)}/{len(questions['depression'])})",
        f"💛 Anxiety ({sum(1 for r in st.session_state.responses['anxiety'] if r is not None)}/{len(questions['anxiety'])})",
        f"💚 Stress ({sum(1 for r in st.session_state.responses['stress'] if r is not None)}/{len(questions['stress'])})"
```

```
    ])

    with tab1:
        st.header("Depression Assessment (PHQ-9)", divider="blue")
        for i in range(len(questions['depression'])):
            question_component('depression', i)

    with tab2:
        st.header("Anxiety Assessment (GAD-7)", divider="blue")
        for i in range(len(questions['anxiety'])):
            question_component('anxiety', i)

    with tab3:
        st.header("Stress Assessment (PSS)", divider="blue")
        for i in range(len(questions['stress'])):
            question_component('stress', i)

    if st.button("📊 Get Results", type="primary", use_container_width=True):
        depression_score = sum(st.session_state.responses['depression']) * 3
        anxiety_score = sum(st.session_state.responses['anxiety']) * 3
        stress_score = sum(st.session_state.responses['stress']) * 3
        show_results(depression_score, anxiety_score, stress_score)

    # Games Section in Sidebar
    st.sidebar.markdown(f"""
    <div style="background:{CARD_BG}; padding:15px; border-radius:10px; margin-bottom:20px">
        <h3>🧠 Wellness Games</h3>
        <p style="color:{LIGHT_TEXT}; opacity:0.8">Relax and train your mind</p>
    </div>
    """, unsafe_allow_html=True)

    game_choice = st.sidebar.selectbox(
```

```
        "Choose a Game",
        ["None", "Positive Affirmations", "Breathing Exercise", "Emotion Recognition", "Word
Puzzle"]
    )

    if game_choice == "Positive Affirmations":
        positive_affirmations_game()
    elif game_choice == "Breathing Exercise":
        breathing_game()
    elif game_choice == "Emotion Recognition":
        emotion_game()
    elif game_choice == "Word Puzzle":
        word_puzzle_game()

    # Footer
    st.markdown("---")
    st.markdown(f"""
    <div style="text-align:center; color:{LIGHT_TEXT}; opacity:0.7">
        MindWell Pro © 2023 | Mental Health Support
    </div>
    """, unsafe_allow_html=True)

if __name__ == "__main__":
    main()


!streamlit run mindwell_premium.py --server.port 8501

# Run with ngrok
from pyngrok import ngrok

# Set your ngrok authtoken
ngrok.set_auth_token("2wiFCa57AIsopiIHq2ryYgFD9Y5_3qu1DBa1A1udor1hDKkxx")  #
```

```
Replace with your actual token

# Start ngrok tunnel
public_url = ngrok.connect(8501)
print(f"Public URL: {public_url}")

# Run Streamlit
!streamlit run mindwell_premium.py --server.port 8501 --server.enableCORS false --
server.enableXsrfProtection false &
```



9.4 FINAL OUTPUT (MINDWELL PRO)

CHAPTER-10

CONCLUSION

The project had the objective of exploring the potential application of AI algorithm-based algorithms for forecasting psychological wellness states, depression, anxiety, and stress levels in interactive games and machine learning protocols. The research aimed to evaluate how integration of behavioral patterns from game user activity with traditional survey psychological measurement instruments (DASS-42) could provide realistic and actionable results to the mental state of an individual. This project then aimed to provide a more affordable, interactive, and scalable alternative to the traditional mental health examination, making psychological testing more accessible and interactive.

Key Findings

The project results have confirmed that machine learning models, such as Random Forest, XGBoost, and Support Vector Machine (SVM), are extremely suitable for predicting mental health conditions from interactive game data and survey outcomes. The following were the key findings of the analysis:

High Accuracy in Predictions:

XGBoost performed better than the other models, with high accuracy rates of 94.2% in depression, 92.1% in anxiety, and 93.4% in stress prediction. This proves the success of ensemble techniques like XGBoost in handling complex data with numerous features.

The ensemble model, with the integration of multiple classifier predictions (Random Forest, Naive Bayes, and XGBoost), received the highest accuracy and stability overall, with 95% for depression, 93.4% for anxiety, and 94.7% for stress.

Metrics Beyond Accuracy:

Besides accuracy, precision, recall, and F1-score, the models were also measured with stress and depression recall. The ensemble model fared well in all these metrics again, reinforcing the fact that it can correctly classify depression, anxiety, and stress and also steer clear of false positives and false negatives.

Recall for depression and stress was particularly high, reinforcing the system's capacity to recognize these two conditions among individuals and perhaps prevent severe escalation through early intervention.

Usability and Engagement:

Interactivity is one of the strengths of this system. User feedback revealed that the game-based test was fun as much as it was effective. A majority of participants utilized self-help activities following feedback, suggesting that the interactivity of the system could motivate individuals to take positive steps towards improving their mental well-being.

93% of the users reported that they had fun playing the games, and 80% appreciated the feedback, which supports the concept that game-based psychological tests can be employed as both diagnostic and therapeutic tools.

Privacy and Ethical Issues

The project maintained rigorous privacy protocols, and GDPR regulations were applied to ensure anonymity of the users' personal data. User permission was also obtained, and participants remained fully in control of their involvement in the study to guarantee transparency and trustfulness of the data gathering process.

Implications and Practical Applications

The outcomes of this project have several important implications for mental health services and AI technology:

Accessibility to Mental Health Services:

The introduction of AI-driven tests into online games simplifies the monitoring of mental health since individuals can test their mental health at home. The system is likely to bridge the gap between in-person diagnosis and remote self-assessment, especially in areas where mental health professionals are scarce.

Real-time mental health status updates can make individuals more resilient so that they are able to make good decisions regarding their well-being. It can prove to be an antidote to mental illness, identifying early signs of depression, anxiety, and stress before these develop into major mental health issues.

Augmenting Traditional Therapy

While this AI-driven system cannot replace clinical professional diagnoses, it can be utilized to complement traditional mental health interventions in the form of continuous observation. It can be utilized as a baseline to ascertain at-risk individuals and refer them to the respective professional help.

The technology can be incorporated into telemedicine interfaces, providing remote assessment and even developing individualized mental health plans depending on the user's predicted status.

Augmentation of AI Models:

With larger data sets and more diverse sources of data, the models' generalizability and predictive power can be further improved. In future versions, inclusion of physiological inputs (e.g., face recognition or heart rate) may increase the sensitivity of the system for emotional states and mental health disorders. Adding more data points, such as sleep patterns or social networking, could provide even more insight into the mental health of users and enable predictions to be even more subtle yet accurate.

Gamification for Mental Health

The gamification aspect of the project offers a worthwhile opportunity to engage users in mental health screening while having fun. This can help reduce the stigma of mental health screening, encouraging individuals to track and improve their well-being without fear of stigma.

As mental health becomes increasingly a worldwide issue, developing such gamified programs could be a giant leap toward worldwide mental health awareness and action.

Limitations

Although the project has made significant strides, there are a few limitations that need to be addressed in the future:

Small Dataset:

The accuracy of the machine learning models varied with the training dataset. While data from this project provided a good starting point, it was limited in terms of size and demographic diversity. With a more diversified dataset, the model would be able to generalize better between different genders, ages, and cultures.

Model Overfitting:

Despite the impressive precision, overfitting is always a possibility, particularly when using high-complexity models like XGBoost and Random Forest. Future studies need to explore regularization techniques and cross-validation strategies to ensure the models generalize correctly to new, unseen data.

Interpretability of Models

Machine learning algorithms, especially ensemble models, are usually "black boxes," and thus it is unknown how specific features influence the model's predictions. Incorporating steps for model explainability, for instance, SHAP values (Shapley Additive Explanations), would provide transparency and render predictions actionable and interpretable for clinicians and users.

Cultural and Contextual Sensitivity:

Mental health diagnoses are extremely context-dependent depending on environmental, social, and cultural factors. While this project was extremely informative, future versions must consider the cultural context of mental health and adapt the system to include the same in order to provide users from different regions with meaningful and relevant feedback.

Future Work and Improvements

The future potential of this project is vast, and there are several directions that can be pursued:

Integration of Physiological Data:

The future systems could have wearable technology or sensors to collect real-time physiological information, such as heart rate, skin conductivity, and facial expression analysis. This information could give additional insights about users' psychological and emotional status, improving the overall prediction accuracy.

Working Together with Mental Health Professionals:

To further enhance the clinical utility of the system, it can be connected to telemedicine platforms where users are able to upload their assessment results for mental health professionals' review and consultation. This would create an integrated mental health care system that combines AI systems with human inputs.

Integration of Real-Time Emotional TrackingI

Developing real-time emotional tracking systems within the games would allow the AI to modify the level of difficulty in tasks in real-time, providing customized experiences that are adjusted in accordance with the user's current emotional state.

Increased Variety of Mental Disorders:

Increased coverage of mental health conditions under the system is yet another area where opportunities can be pursued in the future. The inclusion of conditions such as bipolar disorder, schizophrenia, or PTSD might be able to provide a greater range of support to individuals who suffer from other types of mental health disorders.

Final Thoughts

The project has established the potential of AI-algorithms in providing meaningful information regarding the mental health of the people through engaging and interactive means. Using machine learning models trained on user interaction data and traditional survey results, the system was successful in predicting depression, anxiety, and stress accurately. The engaging nature of game-based assessment makes mental health tracking more accessible, less intrusive, and interactive and thus offers an effective solution to some of the drawbacks of traditional mental health diagnostic methods.

The success of the system not only shows the superiority of AI in the field of mental health but also offers new possibilities for combining interactive technology with traditional psychological care. With more development and refining, this method based on AI can revolutionize mental health screening, offering more personalized, accessible, and preventive solutions to global mental health concerns.

CHAPTER-11

REFRENCES

- R. Katarya and S. Maan, "Predicting Mental Health Disorders Using Machine Learning for Employees in Technical and Non-Technical Companies," 2020 IEEE International Conference on Advances and Developments in Electrical and Electronics Engineering (ICADEE), 2020, pp. 1–6. doi: 10.1109/ICADEE51157.2020.9368923.

- T. Jain, A. Jain, P. S. Hada, H. Kumar, V. K. Verma, and A. Patni, "Machine Learning Techniques for Prediction of Mental Health," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 1–6. doi: 10.1109/ICIRCA51532.2021.9545061.

- J. N. Lavhare and M. A. Kulkarni, "Mental Disorders Detection Using Social Networking Sites," 2021 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021, pp. 113–117. doi: 10.1109/ICICV50876.2021.9388493.

- U. S. Reddy, A. V. Thota, and A. Dharun, "Machine Learning Techniques for Stress Prediction in Working Employees," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2018, pp. 1–4. doi: 10.1109/ICCIC.2018.8782362.

- S. I. Khan, A. Islam, A. Hossen, T. I. Zahangir, and A. S. M. L. Hoque, "Supporting the Treatment of Mental Diseases Using Data Mining," 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2018, pp. 339–344. doi: 10.1109/ICISET.2018.8745611.