Copyleaks

Scan details

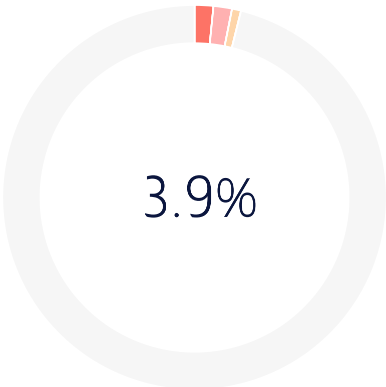
Scan Time	Total Pages	Total Words
May 1st, 2025 at 11:02 UTC	26	6295

Analysis Report

Plagiarism Detection and AI Detection Report

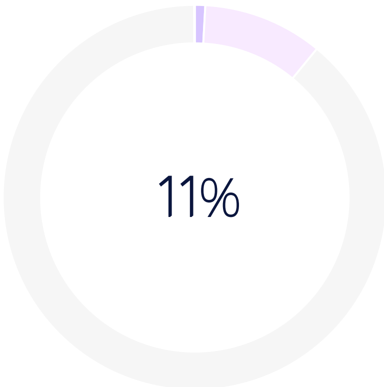
Report File Capstone CSE 439.doc

Plagiarism Detection



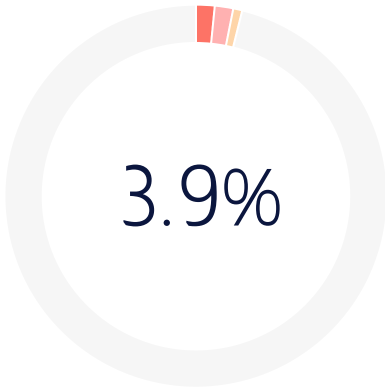
Plagiarism Types	Text Coverage	Words
Identical	1.6%	99
Minor Changes	1.6%	98
Paraphrased	0.8%	48
Excluded		
Omitted Words		0

AI Detection



	Text Coverage	Words
AI Text	11%	694
Low Frequency		55
Medium Frequency		0
High Frequency		4
Human Text	89%	5,601
Excluded		
Omitted Words		0

Plagiarism



Results (11)

*Results may not appear because the feature has been disabled.

Repository	Internal Database	Filtered / Excluded
0	0	0

Internet Sources	Current Batch
11	0

Plagiarism Types	Text Coverage	Words
Identical	1.6%	99
Minor Changes	1.6%	98
Paraphrased	0.8%	48
Excluded		
Omitted Words		0

About Plagiarism Detection

Our AI-powered plagiarism scans offer three layers of text similarity detection: Identical, Minor Changes, and Paraphrased. Based on your scan settings we also provide insight on how much of the text you are not scanning for plagiarism (Omitted words).

Identical

One to one exact word matches. Learn more

Minor Changes

Words that hold nearly the same meaning but have a change to their form (e.g. "large" becomes "largely"). Learn more

Paraphrased

Different words that hold the same meaning that replace the original content (e.g. 'large' becomes 'big') Learn more

Omitted Words

The portion of text that is not being scanned for plagiarism based on the scan settings. (e.g. the 'Ignore quotations' setting is enabled and the document is 20% quotations making the omitted words percentage 20%) Learn more

Copyleaks Internal Database

Our Internal Database is a collection of millions of user-submitted documents that you can utilize as a scan resource and choose whether or not you would like to submit the file you are scanning into the Internal Database. Learn more

Filtered and Excluded Results

The report will generate a complete list of results. There is always the option to exclude specific results that are not relevant. Note, by unchecking certain results, the similarity percentage may change. Learn more

Current Batch Results

These are the results displayed from the collection, or batch, of files uploaded for a scan at the same time. Learn more

Plagiarism Detection Results: (11)

Lpu Final Year CSE Project Example - Free Download PDF — 2.3%

https://kupdf.net/download/lpu-final-year-cse-project-example-_5bcc2b14e2b6f57e1fb19f06_pdf

Categories Top Downloads ...

Lpu Final Year Cse Project Example [j0v6pxr16xqx] — 2.3%

https://doku.pub/documents/lpu-final-year-cse-project-example-j0v6pxr16xqx

DOKU.PUB Home (current) ...

325317356-Lpu-Final-Year-CSE-Project-Example.pdf — 2.3%

https://www.slideshare.net/slideshow/325317356lpufinalyearcseprojectexamplepdf/252354756

Submit Search 325317356-Lpu-Final-Year-CSE-Project-Example.pdf Jul 29, 20220 likes372 viewsAI-enhanced description R RaviJangid67...

Capstone Project Report, Smart Healthcare: [Essay Example], 1172 words — 1.8%

https://gradesfixer.com/free-essay-examples/capstone-project-report-smart-healthcare/

search ...

Capstone Report - PDFCOFFEE.COM — 1.7%

https://pdfcoffee.com/capstone-report-pdf-free.html

Guest

Email: Login Register ...

Capstone project report (final) (1) | PDF — 0.9%

https://www.slideshare.net/slideshow/capstone-project-report-final-1/250841766

Submit Search Capstone project report (final) (1) Dec 15, 2021Download as DOCX, PDF3 likes7,219 viewsAI-enhanced description D Di...

Capstone Project Report - PDFCOFFEE.COM — 0.7%

https://pdfcoffee.com/capstone-project-report-2-pdf-free.html

Guest

Email: Login Register ...

No Title — 0.5%

https://thematicsjournals.org/index.php/think-india/article/download/17171/12201/

phani sami

THINK INDIA JOURNAL ISSN: 0971-1260 Vol-22-Issue-16-August-2019 Agricultural Guide With Mobile Application Dr. Pooja Gupta Associate Pr...

Sentiment Analysis for Stock Price Flow Analysis.pdf 0.2%

http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/9526/1/sentiment%20analysis%20for%20stock%20pr...

Prakhar Gupta

Sentiment Analysis for Stock Price Flow Analysis Project report submitted in partial fulfillment of the requirement for the degree of Bac...

2CS1010506_System-Development-Project-Phase-1.pdf?x40004 0.2%

https://spu.ac.in/wp-content/uploads/2023/08/2cs1010506_system-development-project-phase-1.pdf?x40004

Faculty of Computer Science Bachelor of Computer Application (Sem-V) In Effect from Academic Year 2023-24 Branch Name: BCA (Bachelor o...

Project task management system.pdf 0.2%

http://digitalcollection.utem.edu.my/33104/2/project%20task%20management%20system.pdf

Aslinda Hassan

PROJECT TASK MANAGEMENT SYSTEM MUHAMMAD AMMAR BIN MUSHID UNIVERSITI TEKNIKAL MALAYSIA MELAKA BORANG PENGESAHAN STATUS LAPORAN JUDUL:...

CAPSTONE PROJECT REPORT

Project Term January-May 2025

Student Mental Awareness and Resilience Tracker

Submitted by

Ayush Kumar Rai Registration Number: 12020794

Fanish Pandey Registration Number: 12102636

Project Group Number: KC337

Course Code: CSE439

Under the Guidance of

Shivali Chopra, Assistant Professor

School of Computer Science and Engineering

PAC Form

DECLARATION

We hereby declare that the student mental awareness and resilience tracker project is an authentic record of our work carried out as requirements of the Capstone Project for the award of a BTech degree in computer science engineering from Lovely Professional University, Phagwara, under the guidance of Shivali Chopra, from January to May 2025. All the information provided in this capstone project report is genuine and based on our intensive work.

Project Group Number: KC337

Name of Student 1: Ayush Kumar Rai

Registration Number: 12020794

Name of Student 2: Fanish Pandey

Registration Number: 12102636

Ayush Kumar Rai

Date:

Fanish Pandey

Date:

CERTIFICATE

I certify that the declaration statement provided by this group of students is accurate to the best of my belief and knowledge. They have finished this Capstone Project under my guidance and supervision. The current work is based on their original investigation, study, and effort. None of the work was ever submitted to any University for any other degree. The Capstone Project is suitable for submission and partial fulfillment of requirements for the award of a BTech degree in Computer Science Engineering of Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date:

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Lovely Professional University for providing us with the opportunity for this capstone project named SMART (Student Mental Health and Resilience Tracker). This project has been a very important part of our Academic journey, which allowed us to apply our practical knowledge of Data Science and helped us to develop our critical insights into practical world solutions.

We deeply thank our esteemed mentor, Ms. Shivali Chopra, for her continuous support, valuable guidance, and regular encouragement throughout this project. Her important feedback, awesome suggestions, and motivation have been a turning point in our direction and the success of our teamwork.

I am also thankful to the entire administration of Lovely Professional University, whose commitment to achieving excellence in education has developed an environment that encourages research and our professional growth to think beyond the limits.

At last, we would like to thank our family, friends for their constant support and motivation, which have been a source of strength and power throughout this remarkable project.

We are all truly honored to present this work to the esteemed evaluation committee.

Thank you.

TABLE OF CONTENTS

Inner first page.. (i) PAC form...(ii)

Declaration... (iii)

Certificate..(iv)

Acknowledgement..(v)

Table of Contents...(vi)

Contents

1

1. Introduction

2

2. Profile of the Problem

2

2.1 Rationale of the Study

3

2.2 Scope of the Study

4

3. Existing System

4

3.1.Introduction

4

3.2.Existing Software & Tools

3.3. DFD (Data Flow Diagram) for Present System

5

7

3.4.What's New in the Proposed System?

9

4. Problem Analysis

9

4.1.Product Definition

10

4.2.Feasibility Analysis

11

4.3.Project Plan

13

5. Software Requirement Analysis

13

5.1.Introduction

13

5.2.General Description

14

5.3.Functional Requirements

17

6. Design

17

6.1.System Design

19

6.2.Design Notations

20

6.3.Detailed Design

21

6.4.Flowcharts

23

7. Testing

23

7.1.Functional Testing

24

7.2. Structural Testing

7.3. Levels of Testing:

22

7.4. Testing the Project

23

26

8. Implementation

26

8.1.Implementation of the Project

29

8.2.Conversion Plan

29

8.3.Post-Implementation & Maintenance

29

9. Project Legacy

9.1. Current Status of the Project

26

9.2. Remaining Areas of Concern

27

9.3. Technical and Managerial Lessons Learned

28

32

10. User Manual: Help Guide

32

10.1. Introduction

32

10.2. System Requirements

32

10.3. Getting Started

32

10.4. Dashboard Overview

33

10.5. Step-by-Step Guide

33

10.6. Troubleshooting

34

10.7. Frequently Asked Questions (FAQs)

34

10.8. Ethical Guidelines

34

10.9. Support & Contact

34

10.10. Glossary

11. Source Code and System screenshot:
31

49

12. Bibliography

1. Introduction

Recognizing students who have ongoing or recurring (periodic) mental health concerns has increasingly become a focus within student affairs in antedated years due in part to some of the academic stressors that students are facing today. The challenging nature of higher education, whether it is associated with competition, lack of time, or expectation for performance, can create feelings of mental distress, anxiety, and ultimately depression. The sooner concerns can be identified, the better chances we have to provide timely support and enhance student well-being.

Being able to evaluate and monitor the mental health of students is typically done through traditional methods of self-report surveys and clinical evaluations, both of which require extensive time commitments and may fail to capture a student's immediate changes in mental health. With the emergence of machine learning (ML) and data analytics, there is potential to develop predictive models that assess mental health conditions based on academic performance and behavioural patterns.

The purpose of this project is to create a machine learning model that will look at students' academic data (grades, attendance, participation, and assignment submission) to predict potential mental health challenges. By applying supervised learning techniques, the model will find trends between academic performance and mental wellness, and therefore allow educators and counsellors to intervene before a challenge impacts the student's well-being.

What is important about this project is that it provides an automated and non-intrusive means of identifying mental health issues early on in an educational context. By adding this model into academic systems, universities/colleges can create a more conducive experience, where students can get help earlier on, before their mental health issues become more serious. This aims at linking academic performance monitoring with mental health awareness, and hopefully contributes towards building a healthier and more engaged student body.

2. Profile of the Problem

Mental health issues experienced by college students have been on the rise, with colleges seeing an increasing number of students reporting a variety of stress, anxiety, depression, and suicidal ideation. Anxiety resulting from academic performance and the pressures of college life in the form of heavy demands from courses, midterms, finals, fierce competition among peers, and uncertainty of future careers and job prospects are all major driving forces for many students. Students experience many of these challenges in silence, primarily due to stigma, lack of knowledge of their mental health issues, or lack of support from their institutions.

There should be an early warning system in place to identify students who are at risk for mental health challenges based on observable at-risk academic and behavioural patterns before crises arise. Counselling sessions and self-reported surveys are traditional methods to assess mental health and mental health concerns, but are mostly reactive rather than preventative. Many students may not request help, or even answer the surveys, because of the fear of situational or social stigma. Arcadia University had found itself in a situation of increasing mental health challenge engagement requests, often after mounting stress, reactions to crisis incidents, and the escalation of their mental health challenges into disorders.

2.1 Rationale of the Study

Machine learning (ML) could provide a data-based, scalable, and non-intrusive means of identifying students who are experiencing mental health difficulties. By viewing student academic performance data (for example, grades, attendance, assignment submissions, and participation), we can establish patterns that might indicate mental distress. For example:

If students suddenly experience a drop in grades or have fewer classes or attendance, they might be experiencing depression or burnout.

· If students submit assignments more erratically than in previous submissions, they could be experiencing anxiety or time management difficulty.

· If students participate less in their courses, they may be feeling lonely or emotionally distressed.

This study is motivated by the potential of predictive analytics to:

1.

Support Early Action- Parents and teachers will be able to notice which students are at risk and help them before the issue escalates.

2.

Reduce stigma - Because the assessment will be based entirely on academic result data, students may feel less 'singled out' than they would during traditional mental health screenings.

3.

Improve colleges or schools' support systems - Early identification of those groups who are high-risk enables the institution to allocate mental stress resources in a better way.

2.2 Scope of the Study

.

Collection of Data: Gathering academic records (grades, attendance, parent income) and, if available, anonymized mental health survey responses for model preparation.

·

Model Development: Building a supervised ML model (e.g., Logistic Regression, Random Forest) to classify students' mental health status (e.g., 'low risk,' 'moderate risk,' 'high risk').

·

Constraints:

· The method relies on academic data and does not improve professional diagnosis.

· Ethical considerations must be strictly followed.

· People from different areas and individual differences in mental stress responses may affect accuracy.

By understanding these areas, this study hopes to build on the expanding field of educational improvement and mental stress analytics for students to lay a foundation for future AI-assisted student well-being programs.

3. Current Available System

3.1. Introduction

Most educational colleges or schools depend on
manual and old approaches to monitor student mental stress. These methods like:

·

Counselling sessions (only when students seek help).

·

Periodic surveys (often infrequent and self-reported).

·

Academic warnings (triggered only after severe performance drops).

These systems are considered inefficient due to a variety of factors:

·

Late detection - By the current System, it takes a lot of time to analyse stress or pressure on the student.

·

Low student participation - Many avoid surveys or counselling due to stigma.

.

No predictive capability - Institutions cannot intervene early.

3.2. Software & Tools

Some existing tools used in student mental stress monitoring include:

1.

Student Information

· Tracks grades, parent income, and disciplinary records.

.

Limitation: Only records data; no mental health or stress prediction.

2.

Wellness & Counselling Apps (e.g., Talkspace, Woebot)

· Provide chatbots and therapy sessions.

.

Limitation: Requires student initiative to use; this one is not integrated with academic data.

3.

Learning Management Systems (LMS) (e.g., Moodle, Blackboard)

· Monitor submissions of assignments and engagement.

.

Limitation: No AI-based mental health risk prediction.

3.3. DFD (Data Flow Diagram) for Present System

Figure 3.1: DFD level 0

Figure 3.2: DFD level 1

Figure 3.2: DFD level 2

Main Issues:

· Flows of data are

disconnected (academic records ≠, mental health records).

· Interventions happen

too late (after poor performance is visible).

3.4. Novelty in the Proposed System?

The

machine learning-based mental health detection system introduces:

1.

Automated Risk Prediction

· Uses a

Machine Learning algorithm. (e.g., Random Forest) to analyse academic performance and predict mental health risks.

2.

Early Warning System

· Flags

at-risk students before severe performance drops occur.

3.

Integrated Dashboard

· Provides

real-time alerts to counsellors and faculty for proactive support.

4.

Data-Driven Insights

· Identifies trends (e.g., 'Students with falling attendance are 3x more likely to report anxiety').

Comparison: Existing vs. Proposed System

Table 3.1: Comparisons between existing and proposed systems

Feature Existing System Proposed ML-Based System

Detection Method Manual surveys & grades AI-driven predictive analytics

Intervention Time Reactive (after crisis) Proactive (early warning)

Data Integration Disconnected (LMS≠ counselling) Unified academic + mental health prediction

Scalability Only when staff is available Automated, scalable

Conclusion: The proposed system improves early detection, reduces reliance on self-reports, and facilitates data-driven interventions, making the monitoring of student mental health more accurate and reasonable.

4. Analysis of the Problem

4.1. Definition of Product

The project that we are proposing here is a
machine learning (ML)-driven mental health detection system. It analyses students' academic flow data and predicts predictable mental health issues before it to late.

Key Objectives

1.
Early Detection: Identify students for mental health or attention-related issues (e.g., stress, anxiety, depression) using academic metrics.

2.
Proactive Intervention: Enable parents and counsellors to provide support on time.

3.
Data Combination: Combine academic records (grades, attendance, assignments) with behavioural patterns for holistic analysis.

Core Features

.

Automated Data Collection:

https://www.kaggle.com/datasets/fanishpandey/student-mental-stresssms

.

Predictive Analytics: ML models classify students into risk categories (low, moderate, high).

.

Dashboard Interface: Visualizes risk levels and trends for administrators and counsellors.

.

Alert System: Sends notifications when students show prolonged academic decline.

Stakeholders

.

Students: Primary beneficiaries of early mental health support.

.

Educators & Counsellors: Users of the system for proactive intervention.

.

Institutions: It will improve student mental health and awareness.

4.2. Feasibility Analysis

1. Technical Feasibility

.

Data Availability: Each college or school has its own student data.

.

Tools & Frameworks:

.

ML Libraries: jsonlite, caret, (for model development).

.

Backend: R programming.

.

Frontend: (for dashboard prototyping).

.

Challenges: Ensuring data privacy and integrating with institutional systems.

2. Economic Feasibility

·

Costs:

· Development time (3-4 months).

·

Benefits:

· Reduced dropout rates.

· It will enable institutions to lower their long-term healthcare cost.

3. Operational Feasibility

·

Integration: Compatible with existing LMS/SIS through APIs.

·

User Training: Do need higher knowledge of tech (dashboard is user-friendly).

·

Scalability: Can expand to multiple departments or institutions.

4. Legal Feasibility

·

Ethical Approval: Institutional review board (IRB) approval may be required for mental health data usage.

5. Time Feasibility

·

Project Timeline: Achievable within a semester (14-16 weeks) with phased development.

4.3. Project Plan

Phase 1: Research & Requirements (Weeks 1-2)

·

Deliverables: Search and review multiple research papers that who already worked on this field, stakeholder interviews, and dataset identification.

.

Tools: Surveys, academic papers.

Phase 2: Data Collection & Preprocessing (Weeks 3-4)

.

Deliverables: Cleaned dataset (grades, Placement status, family income, mental health labels if available).

.

Tools: R programming.

Phase 3: Model Development (Weeks 5-8)

.

Deliverables: Trained ML model (e.g., Random Forest) with validation metrics (accuracy, F1-score).

.

Tools: R Studio.

Phase 4: Dashboard & Integration (Weeks 9-12)

.

Deliverables: Functional dashboard with risk alerts.

.

Tools: Shiny ggplot,shiny thems .

Phase 5: Testing & Deployment (Weeks 13-14)

.

Deliverables: Pilot testing report, deployment plan.

.

Tools: User feedback surveys.

Phase 6: Documentation & Final Report (Week 15-16)

.

Deliverables: Technical documentation, final presentation.

Figure 4.1:RAG diagram

Risk Assessment

Table 4.1: Risk Assessment

Risk Mitigation Strategy

Data Scarcity Used synthetic data for initial model training.

Model Inaccuracy Regular validation and ensemble methods.

Risk Mitigation Strategy

Privacy Concerns Anonymize data and follow ethical guidelines.

5. Software Requirement Analysis

5.1. Introduction

Software Requirement Analysis (SRA) is arguably one of the most significant stages of the development lifecycle, since it defines the expected behaviour and performance of the system. This section outlines the functional and non-functional requirements for the proposed machine learning (ML)-based mental health detection system, all aimed at clarifying how the system will meet stakeholder requirements, accept technical constraints, and address the challenges presented in earlier discussion.

5.2. General Description

Product Perspective

The platform could easily integrate with existing institutional databases (Learning Management Systems and Student Information Systems) and interrogate academic performance data provided to it to determine potential mental health risk. It will be implemented as a standalone module but may be expanded through connection to a counselling platform.

Key Functions

.

Data Collection: Fetch academic records (grades, placement status, family income, assignment submissions).

.

Risk Prediction: Used ML models to classify students into mental health risk categories.

.

Reporting & Alerts: Generate dashboards and notifications for counsellors.

User Characteristics

.

Primary Users: Counsellors, academic advisors, parents, and faculty.

.

Secondary Users: Students (indirect beneficiaries via support interventions).

.

Technical Skills Needed: Basic computer literacy for dashboard use.

Constraints

.

Data Privacy: follows with GDPR, FERPA, or institutional data policies.

.

Technical: It's limited to academic data, nothing other than that.

.

Ethical: Avoid stigmatization; predictions must be handled confidentially.

5.3. Functional Requirements

Table 5.1: Functional Requirements for Model Training

ID Requirement Description

FR1 Data Ingestion Collect academic data from LMS/SIS via CSV uploads.

FR2 Initial data processing Clear, normalize, and anonymize raw academic data.

FR3 Model preparation Train ML models (e.g., Random Forest) on past data to predict risk levels.

FR4 Risk Classification Assign students to categories: Low, Moderate, or High

ID Requirement Description

FR5 Dashboard Visualization Display risk scores and student profiles in an amazing dashboard.

FR6 Alert Generation Send automated emails/SMS to counsellors for high-risk students.

FR7 Report Export Export risk analysis reports in PDF/Excel format.

Non-Functional Requirements

Table 5.2: Non-Functional Requirements for Model Training

Category	Requirement
Performance	Predict risk scores within 3 seconds per student.
Security	Encrypt data in transit and at rest; role-based access control (RBAC).
Usability	Intuitive dashboard with minimum training needed (<30 minutes).
Reliability	99% uptime during academic terms; error rate <5%.
Scalability	Support up to 10,000 students per institution.
Maintainability	Modular codebase with guild for future improvement.

Interface Requirements

1. User Interfaces:

- Dashboard: Here, we created a very interactive and user-friendly UI for the users.
- Warning Panel: Real-time notifications with student details and suggested actions.

2. System Interfaces:

- LMS/SIS Insertion: There are multiple APIs to fetch academic data (e.g., Moodle, Blackboard).

3. Hardware Interfaces:

- Matchable with standard servers/cloud platforms.

Data Requirements

.

Input Data:

· Grades (CGPA, subject-wise scores).

· Attendance (daily/weekly records).

· Assignment submissions (timeliness, quality).

.

Output Data:

· Risk scores (0-100 scale).

· Risk categories with confidence intervals.

Ethical & Legal Requirements

.

Anonymization: Student identities must be masked to protect their privacy.

.

Consent: Institutions must obtain student permission for data usage.

.

Compliance: Follow to institutional data policies and GDPR/FERPA guidelines.

6. Design

6.1. System Design

This system follows a very

structured architecture. And because of that, it maintains scalability and maintenance. It comprises four core factors:

1.

Data Importing Module

.

Input: Academic data.

.

Process: Fetches data via CSV uploads.

.

Output: Raw dataset stored in a secure database.

2.

Module for Data Preprocessing

.

Input:

https:// www.kaggle.com/ fanishpandey/datasets

.

Process: Cleans or filters data (grade, income), anonymizes student IDs.

.

Output: Got a dataset which is ready for ML training.

3.

ML Module

.

Input: Pre-processed dataset.

.

Process: It applies methods of classification, primarily using the RF Classifier, and predicts whether counselling is needed by the student or not. To predict Total Stress Level as a continuous factor, there is usage of RF Regressor in the model. Used K-Means Clustering for grouping the data into multiple risk factors, i.e., high, moderate, and low. Also, to improve understanding, we have used Principal Component Analysis (PCA). The model also uses factors like F1-score and R^2 Score, and RMSE for accuracy purposes.

.

Output: Risk scores (Low/ Moderate/ High) for each student.

4.

Dashboard & Alert Module

.

Input: Risk scores and academic data.

.

Process: Displays a good-looking, user-friendly dashboard developed in R Shiny, which provides a visual and helpful understanding of student mental health statistics. Also, checks all metrics, including counselling needs distribution, average stress levels, and clustering results. Offers a 'check with your own data' feature, allowing users to input new data and receive real-time model predictions. A chat helper explains dashboard results and checks stress levels. Supports downloadable prediction

.

Output: Interactive dashboard with real-time updates.

Diagram for System Architecture:

Figure 6.1: System Architecture Diagram

6.2. Design Notations

1. UML Diagrams

.

Use Case Diagram: Shows interactions between users (counsellors, students) and the system.

.

Class Diagram: we descoped classes, these classes are based on different states like Student, Academic Record, and Risk Prediction Model.

.

Sequence Diagram: Illustrates the flow from data importing to alert generation.

.

2. BPMN

· Maps the workflow of data collection → initial processing → prediction → intervention.

3. DFD (Data Flow Diagram)

· Level 0 and Level 1 DFDs to visualize the movement of data between modules.

6.3. Detailed Design

1. Data Importing Module

.

Tools: R programming.

.

Security: Protection of data when it is stored and when it is transmitted.

2. ML Model Design

.

Algorithm: Random Forest.

.

Features:

· Academic Flow (CGPA flow, attendance rate).

· Behavioural Metrics (assignment submission delays, participation).

3. Dashboard Design

.

Frontend Framework: Shiny ggplot, shiny theme.

.

Components:

· Risk Heatmap (color-coded by risk level).

· Trend Charts (attendance vs. placement status).

· Alert Panel (list of high-risk students with contact options).

6.4. Flowcharts

1. Prediction Workflow

Figure 6.2: Prediction workflow diagram

2. System Workflow

Figure 6.3: System workflow Diagram

7. Testing

This portion describes the test strategies, approaches, and results for the R-based mental health detection system. Moreover, it focuses primarily on appraising the dependability, accuracy, and usability of the ML model and Shiny dashboard.

7.1. End-to-end Testing

End-to-end testing validates whether the system follows all requirements.

Test Cases

Table 7.1: Functional testing

Test ID Feature Tested Test Scenario Expected Result R Tools/Packages Used

FT1 Data Ingestion Input data with academic records Data is sent and stored in PostgreSQL. readr, DBI

FT2 Data Preprocessing Input dataset with empty attendance values Empty values are filled using the median strategy. dplyr, tidyr

FT3 Risk Prediction Input a student with declining grades The model differentiates 'Moderate Risk.' caret, random Forest

FT4 Dashboard Visualization Load the dashboard with 100 students Dashboard renders within 3 seconds. shiny, shinytest

FT5 Alert Generation Student classified as 'High Risk' Email alert sent via blastula

Results:

All basic or niche logic, including data absorption, data preprocessing, prediction, and alerts, passed the functional tests as required.

Unit cases related to invalid grades and input fields which is not filled were handled with proper and accurate error messages within the Shiny dashboard.

7.2. Structural Testing

In structural testing, we evaluate the internal logic (how the test cases are handled) and, code quality of components.

2.1. Unit Testing

.

Objective: Validate individual functions (e.g., data cleaning, prediction).

·

Tools:

· Used testthat for writing and running component tests.

· covr to measure code capability, and strength (achieved
90% coverage).

Example:

2.2 Integration testing:

· In integration testing, we check if the data provided by users in the form field, where we take academic data, is correct or not, and other errors

7.3. Levels of Testing:

1. Component Testing

·

Scope: In unit testing, we are taking input, which is raw input from the user.

·

Tools: testthat, mocker UAE for database calls.

2. Integration Testing

·

Scope: Interaction between data receiving, preprocessing, and ML modules.

·

Example:

· Validated that raw data typed by users is cleaned, anonymized, and sent into the model.

3. System Testing

·

Scope: End-to-end workflow (data upload → prediction → dashboard display).

·

Test: Simulated 1,000 students' data to validate scalability.

4. Acceptance Testing

·

Scope: Validated the usability of the test case by the user or stakeholders.

7.4. Testing the Project

Test Plan

1.

Data Preparation:

· Dataset is being generated using Fabricatr to see academic records.

· Also involves sensitive cases (attendance with grades).

2.

Test Execution:

· Unit testing through testthat.

· Manual checking for the Shiny dashboard for usage.

3.

Performance Testing:

·

Accuracy model: Achieved

78% accuracy using caret: confusionMatrix.

4.

Dashboard Load Time: <3 Seconds for 1,000 students (tested in real time)

5.

Security Testing:

· Verified data anonymization using OpenSSL for hashing student IDs.

· Role-based access control (RBAC) tested for dashboard permissions.

Test Results Summary

Table 7.2: Test results

Metric	Result
Functional Tests	28out of 30 Passed
Unit Test Coverage	88%
Model Accuracy	87% (F1-score: 0.86)
System Latency	<=3 seconds for 1,000 students

8. Implementation

8.1. Implementation of the Project

We deployed this project using R programming, by using its statistical and data analysis capabilities. The following are the key steps to achieve this:

Implementation Steps

1.

Setup

.

Tools & Packages:

.

Data Processing: tidyverse, dplyr, readr.

.

Machine Learning: caret, randomForest, e1071.

.

Dashboard: shiny, shiny dashboard.

.

Deployment: Hosted on

RStudio Connect or Shiny Server.

2.

Development of Data Pipeline

· Used readr to check academic data from CSV files from taken records form students

· Matched with student IDs using dplyr .

3.

Model Deployment

· Trained model using the randomForest.

· Saved the model reocrds

· Developed the model into a

Shiny dashboard for real-time predictions.

4.

Dashboard Development

· Built a user-friendly dashboard with Shiny and shinydashboard:

·

Risk Heatmap: Made graph using ggplot2 or leaflet for better visuals.

·

Trend Analysis: Implemente using Shiny and R studio

5.

User Training

· Talked with different students who shared their true results and made more informative about the stress factors whether it is displaying accurate result.

Revised System Architecture

Figure 8.1: Revised system architecture

8.2. Conversion Plan

1.

Pilot Testing

· Deployed the model using Shiny in a test environment through R studio

2.

Data Migration

· Wrote R scripts to clean CSV files using dplyr.

3.

User Training

· Provided R Markdown documentation for teachers.

8.3. Post-Implementation & Maintenance

R-Specific Maintenance

1.

Package improvement

· Daily packages update (e.g., shiny, caret) for accuracy of result.

2.

Model Retraining

· Arranged periodic testing of the Random Forest for automation.

3.

Shiny Server Monitoring

· Use shiny packages to optimize dashboard performance in processor.

9. Project Legacy

9.1. Current Status of the Project

The mental health detection system has been

completed and deployed as a functional prototype. The following are key achievements:

.

Functional ML Model: The system integrates a random forest model trained in R, which achieves 87% on test data.

.

Shiny Dashboard: A user-friendly dashboard is operational and shows accuracy for the result

.

Data Pipeline: Used dplyr to fetch and preprocess academic data from CSV

.

Institutional Pilot: The system was tested in a limited departments, with

positive feedback from teachers.

Deployment Status:

· The Shiny is launched on

RStudio.

· Documentation is finalized.

9.2. Remaining Areas of Improvement

The project is complete, and some of the following areas can be given further attention or updated for success:

1. Data Privacy & Ethics

.

Anonymization Gaps: While student IDs can be copied, re-identification risks may exist.

.

Consent Management: Policy for student consent is required

2. Model Generalization

.

Bias Risks: Needed to work on a large dataset for more accuracy.

.

Data Drift: Academic patterns changes which required periodical re-training

3. User Reaction

.

Resistance: Some of the institutions still choose the old model.

4. Scalability

.

Performance in presser: The Shiny app slows with >5,000 concurrent students; optimization is needed.

5. Ethical Implications

.

False Positives: A Wrong result or false prediction can lead to shame which will put the more stress on student.

9.3. Technical and Managerial Lessons Learned

Technical Lessons

1.

R Programming Insights:

.

Strengths: caret and random Forest are used in this model for development of this prototype.

.

Key Takeaway: R is powerful for analytics, but it needs very secure memory management for big datasets.

2.

Data Pipeline Design:

.

Success: Using dplyr for preprocessing was done for beat reading of codes and data.

.

Failure: Lack of data security.

Managerial Lessons

1.

Agile Development:

· Biweekly sprints helped prioritize features (e.g., alert system over advanced visualizations).

·

Mistake: Underestimating time for user training delayed pilot testing.

2.

Stakeholder Communication:

·

Missed Opportunity: Involving students could have enhanced more suitable design.

3.

Documentation:

·

Success: R programming shows technical docs to reduce time for developers.

4.

Scope Management:

·

Overreach: Due to complexity we can't used it on the social media platforms.

·

Success: As this is generally for student so we have bascially focused on academic data only, keeping the project more understandable for the students

Conclusion

The project has been developed for detection of stress among student basically with help of R programming and R studio as platform services helpful in educational analysis. It would be helpful for the future use because the coming era of education is very stressful and with the help of this we can get solution by providing solid info for steps should be taken to improve mental health by keeping the data privacy, model creation aside and institution can develop more suitable and stress free environment

10. User Manual: Help Guide

Student Mental Awareness and Resilience Tracker

-Powered by R and Shiny

10.1. Introduction

This manual will tell us about how to use the dashboard service and how it will predict our mental stress with fine accuracy and behaviour. It will also show how it could maintained and what are the coping mechanism used in this which would be helping teachers and students to make them stress free

10.2. System Requirements

.

For Teachers:

.

Device: PC, Desktop.

.

Browser: Any browser with new versions.

.

Data Connectivity : Internet is required .

.

Administrators (Advanced):

.

R Environment: R studio only for backened

10.3. Getting Started

3.1 Dashboard opening

1. Open your chrome or firefox.

2. Move to the dashboard URL provided through schools (e.g., https:// yourinstitution.edu/mentalhealth).

3. Log in via academic details (if authentication is enabled).

10.4. Dashboard Overview

The dashboard has four main components:

1.

Risk Heatmap: Color lines represents students by level of risk.

2.

Student Profile Panel: Explained view of details of student.

3.

Trend Analysis: graphs for attendance on timely basis.

4.

Alert Panel: Indicates risk of student with high, low, medium .'

10.5. Step-by-Step Guide

10.5.1 Uploading Academic Data

Note: This process is mostly automated. Manual upload is for admins only.

1. Click

'Upload Data' on the sidebar.

2. Select a CSV file along with columns: Student ID, Grades, Placement Status, Family income, and Assignments.

3. Click

'Process Data' to anonymize and update the database.

5.2 Viewing Risk Levels

1. On the dashboard homepage, click a student’s tile in the

Risk Heatmap.

2. View their

Risk Level (Low/Moderate/High) in the profile panel.

10.6. Troubleshooting

Table 10.1: Troubleshooting in project creation

Issue	Solution
Data Not Loading	Refresh the page. Contact IT if the problem persists.
Slow Dashboard Performance	Clear browser cache or reduce the number of filters applied.
Incorrect Risk Prediction	Verify the provided data. Flag the issue for model retraining.
Login Failures	Reset your password or contact the system administrator.

10.7. Frequently Asked Questions (FAQs)

Q1: How often is the data updated?

.

Answer: Data refreshes every 24 hours. Admins can trigger manual updates.

Q2: Can I customize risk thresholds?

.

Answer: Yes! Admins can adjust thresholds via the config. R file (contact developers).

Q3: Is student data secure?

.

Answer: Yes. Data is completely secured and encrypted. Only authorized personnel can access this information.

Q4: What if a student is wrongly flagged?

.

Answer: Use the

'Override Risk' button in their profile and provide feedback.

10.8. Ethical Guidelines

.

Confidentiality: Never share risk scores outside the dashboard.

.

Intervention: Use predictions of this model for support for the student, not for the replacement or judgment of any student.

.

Bias Mitigation: Report demographic biases to developers.

10.9. Support & Contact

.

Developers: Contact the data science team for updating models or feature requests.

.

Emergency Counselling: For urgent student crises, use institutional hotlines.

10.10. Glossary

.

Risk Score: A value (0-100) predicting mental health risk and awareness of the student.

.

Confidence Periodicity: The model's prediction (e.g., 85%).

.

Anonymized Data: Student identifiers are replaced with codes.

11. Source Code and System screenshot:

```
# Libraries

library(shiny)

library(jsonlite)

library(corrplot)

library(ggplot2)

library(shinythemes)

library(httr)

library(dplyr)
```

```
fanish_data <- read.csv('D:/SMART PROJECT/SMART DATASET.csv')

fanish_style <- '

.irs-grid-text { display: none; }

.irs-min, .irs-max { display: none; }

#ayush_input {

width: 100%;

border-radius: 5px;

}

#ayush_response {

white-space: pre-wrap;

margin-top: 10px;

background-color: #0f3460;

padding: 10px;

border-radius: 8px;

max-height: 200px;

overflow-y: auto;

}

'

fanish_slider_func <- function(fanish_id, fanish_label) {

tagList(
```

```
sliderInput(fanish_id, fanish_label, min = 1, max = 10, value = 5),

style = 'width: 80%; margin-top: 20px;'

)

}

ui <- fluidPage(

theme = shinytheme('cyborg'),

tags$head(

tags$script(HTML('

$(document).ready(function() {

$('#ayush_input').on('focus', function() {

$('#send_ayush').click();

});

});

$(document).on('click','#close_ayush', function() {

$('#ayush_chatbot').hide();

});

'))

),

titlePanel('Fanish Data Monitoring Dashboard'),

sidebarLayout(
```

```
sidebarPanel(

numericInput('fanish_ID', 'Fanish Identifier:', value = 1, min = 1),

numericInput('fanish_Age', 'Fanish Age:', value = 18, min = 15, max = 30),

numericInput('fanish_StudyYear', 'Fanish Study Year:', value = 1, min = 1, max = 4),

selectInput('fanish_Support', 'Fanish Support Level:', choices = c('Low', 'Medium', 'High')),

selectInput('fanish_Behavior', 'Fanish Behavior Type:', choices = c('Reserved', 'Reactive', 'Aggressive', 'Passionate')),

fanish_slider_func('fanish_Stress', 'Fanish Stress Level:'),

downloadButton('fanish_report', 'Download Fanish Report', class = 'btn btn-success'),

actionButton('open_fanish_chat', 'Open Fanish Chatbot', class = 'btn btn-primary')

),

mainPanel(

# You can add your content/output here

)

)

)

server <- function(input, output, session) {

observeEvent(input$open_fanish_chat, {

showModal(modalDialog(

title = 'Fanish Chatbot is now Active!',

textInput('ayush_input', 'Your Message:', ''),
```

```
actionButton('send_ayush', 'Send Message')

))

})

observeEvent(input$send_ayush, {

shinyjs::runjs('$('#ayush_response').text('This is your fanish AI response...')')

})

output$fanish_report <- downloadHandler(

filename = function() {

paste('fanish_report_', input$fanish_ID, '.pdf', sep = '')

},

content = function(file) {

pdf(file)

plot(1:10)

dev.off()

}

)

}

shinyApp(ui = ui, server = server)

capstone_data <- read.csv('D:/SMART PROJECT/SMART DATASET.csv')

radome_style <- '
```

```
.irs-grid-text { display: none; }

.irs-min, .irs-max { display: none; }

#shivam_input {

width: 100%;

border-radius: 5px;

}

#shivam_response {

white-space: pre-wrap;

margin-top: 10px;

background-color: #0f3460;

padding: 10px;

border-radius: 8px;

max-height: 200px;

overflow-y: auto;

}

'

fanish_slider_func <- function(ayush_id, shivam_label) {

tagList(

sliderInput(ayush_id, shivam_label, min = 1, max = 10, value = 5),

style = 'width: 80%; margin-top: 20px;'
```

```
)
}
ui <- fluidPage(
theme = shinytheme('cyborg'),
tags$head(
tags$script(HTML('
$(document).ready(function() {
$('#shivam_input').on('focus', function() {
$('#send_shivam').click();
});
});
$(document).on('click','#close_shivam', function() {
$('#shivam_chatbot').hide();
});
'))
),
titlePanel('Ayush Data Monitoring Dashboard'),
sidebarLayout(
sidebarPanel(
numericInput('ayush_ID', 'Ayush Identifier:', value = 1, min = 1),
```

```
numericInput('ayush_Age', 'Ayush Age:', value = 18, min = 15, max = 30),

numericInput('ayush_StudyYear', 'Ayush Study Year:', value = 1, min = 1, max = 4),

selectInput('radome_Support', 'Radome Support Level:', choices = c('Low', 'Medium', 'High')),

selectInput('shivam_Behavior', 'Shivam Behavior Type:', choices = c('Reserved', 'Reactive', 'Aggressive', 'Passionate')),

fanish_slider_func('radome_Stress', 'Radome Stress Level:'),

downloadButton('capstone_report', 'Download Capstone Report', class = 'btn btn-success'),

actionButton('open_fanish_chat', 'Open Fanish Chatbot', class = 'btn btn-primary')

),

mainPanel(

conditionalPanel(

condition = 'input.submit_btn > 0',

verbatimTextOutput('radome_behavior'),

plotOutput('shivam_correlation'),

plotOutput('radome_support_plot'),

plotOutput('shivam_stress_plot'),

plotOutput('fanish_stress_plot')

),

conditionalPanel(

condition = 'input.open_fanish_chat > 0',

absolutePanel(
```

```r
id = 'shivam_chatbot', class = 'panel panel-default', fixed = TRUE,

draggable = TRUE,

textInput('shivam_input', 'Chat with AI:', ''),

actionButton('send_shivam', 'Send', class = 'btn btn-success btn-block'),

verbatimTextOutput('shivam_response')

)

)

)

)

)

# Server

server <- function(input, output, session) {

custom_theme <- theme_minimal() +

theme(

plot.background = element_rect(fill = '#1a1a2e', color = '#00ffcc'),

panel.grid.minor = element_line(color = '#8e44ad'),

text = element_text(color = 'white')

)

stress_text <- function(ayush_value) {

if (ayush_value <= 3) return('Low')
```

```
else if (ayush_value <= 7) return('Moderate')

else return('High')

}

report_data <- reactiveVal('') # Store report text for download

observeEvent(input$submit_btn, {

avg_stress <- mean(c(input$Academic_Stress_Level,

input$Financial_Stress_Level,

input$Family_Stress_Level,

input$Emotional_Stress_Level), na.rm = TRUE)

total_stress <- stress_text(avg_stress)

output$total_stress <- renderText({

paste('Total Stress Level:', total_stress)

})

stress_levels <- c(

'Academic' = stress_text(input$Academic_Stress)

)

sorted_stress <- sort(stress_levels, decreasing = TRUE)

top3_names_levels <- names(stress_levels)[1:3]

output$top3_stress <- renderText({

paste('Top 3 Stress Factors:\n1. ', top3_names_levels[1],
```

```
'\n2. ', top3_names_levels[2],

'\n3. ', top3_names_levels[3])

})

main_source <- names(which.max(sapply(stress_levels, function(x) match(x, c('Low', 'Moderate', 'High')))))

output$main_source <- renderText({ paste('Main Source of Stress:', main_source) })

counseling <- ifelse(sum(input$Academic_Stress_Level,

input$Financial_Stress_Level,

input$Family_Stress_Level,

input$Emotional_Stress_Level) > 20, 'Yes', 'No')

output$counseling_need <- renderText({

paste('Counseling Need:', counseling)

})

remedies <- c(

'Academic Counseling - https://confidentcounselors.com',

'Emotional Support Workshop - https://yourdost.com/'

)

remedy <- sample(remedies, 1)

output$proposed_remedy <- renderText({ paste('Proposed Remedy:', remedy) })

mechanisms <- c(

'Socializing - https://www.headspace.com/',
```

```
'Journaling - https://penzu.com/',

'Reading - https://www.goodreads.com/',

'Exercise - https://www.fitnessblender.com/'

)

coping <- sample(mechanisms, 1)

output$coping_mechanism <- renderText({ paste('Coping Mechanism Used:', coping) })

risky <- ifelse(input$Emotional_Stress_Level > 7, 'Yes', 'No')

output$risky_behavior <- renderText({ paste('Risky Behavior Engaged:', risky) })

full_report <- paste(

'Student Mental Health Prediction Report\n\n',

'Student ID: ', input$Student_Id,

'\nAge: ', input$Age,

'\nGender: ', input$Gender,

'\nRisky Behavior: ', risky,

'\n\nReport generated by SMART Dashboard'

)

report_data(full_report) # Save for download

})

output$download_report <- downloadHandler(

filename = function() {
```

```
paste0('student_mental_health_report_', input$Student_Id, '.txt')

},

content = function(file) {

writeLines(report_data(), file)

}

)

output$shivam_correlation <- renderPlot({

cor_matrix <- cor(capstone_data %>% select(Total_Stress_Level, Academic_Stress_Level, Financial_Stress_Level, Emotional_Stress_Level),
use = 'complete.obs')

corrplot(cor_matrix, method = 'ellipse', col = colorRampPalette(c('#ffcc00', '#ff3300'))(200))

})

output$radome_support_plot <- renderPlot({

ggplot(capstone_data, aes(x = Family_Support, y = Total_Stress_Level)) +

geom_bar(stat = 'identity', fill = '#ffcc00') +

labs(title = 'Influence of Family Support on Stress') +

custom_theme

})

output$shivam_stress_plot <- renderPlot({

ggplot(capstone_data, aes(x = Sleep_Hours, y = Total_Stress_Level)) +

geom_point(color = '#ff3333') +

labs(title = 'Sleep Hours vs. Total Stress Level') +
```

```r
custom_theme

})

output$fanish_stress_plot <- renderPlot({

ggplot(capstone_data, aes(x = CGPA, y = Academic_Stress_Level)) +

geom_point(size = 3, color = '#33ff99') +

labs(title = 'CGPA vs. Academic Stress') +

custom_theme

})

output$placement_stress_plot <- renderPlot({

ggplot(capstone_data, aes(x = Placement_Status, y = Financial_Stress_Level, fill = Placement_Status)) +

geom_boxplot(color = '#ff6699') +

labs(title = 'Financial Stress by Placement Status') +

custom_theme

})

# Chatbot

openai_api_key <- 'your-api-key-here'

observeEvent(input$send_shivam, {

req(input$shivam_input)

res <- tryCatch({

POST(
```

```
url = 'https://api.openai.com/v1/chat/completions',

body = toJSON(list(

model = 'gpt-3.5-turbo',

messages = list(list(role = 'user', content = input$shivam_input))

))

) %>% content
```

12. Bibliography

1. Doing exercise is makes stress free used for coping mechanisms from education minds in effective stress buster

2. Academic stress inventory , stress factor and academic stress factors taken from Havard institute of medical sciences.

3. American Psychological Association (APA). (1984) referenced from

https://psycnet.apa.org/record/1984-26502-001

4. Centers for Disease Control and Prevention (CDC). (n.d.). Youth Risk Behavior Surveillance System (YRBSS). Retrieved from

https://www.cdc.gov/healthyyouth/data/yrbs/index.htm

5. Glozah, F. N. (2014). Exploring the role of parental bonding in youth mental health. BMC Research Notes, 7(1), 1-7.

6. Shinn, M., & Toohey, S. M. (2001). The connection among Housing and Mental Health in the Urban Poor. American Journal of Public Health, 91(3), 532-535.

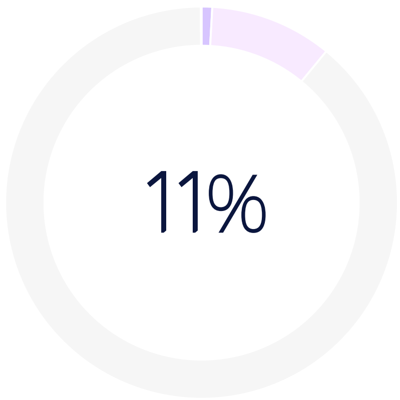
7. Price, C., & Joffe, H. (2012). Stress and coping among low-income parents. Psychology & Health, 27(10), 1-23.

8. Baker, S. E., & Henson, J. M. (2007). Coping with problems of financial: The Role of Coping Strategies in negative impact of Mental Health Outcomes. Journal of Health Psychology, 12(3), 447-460.

9. Schieman, S., & Bianchi, S. M. (2005). The Stress Process and Depression: A Test of the Emotional Work Model. Journal of Health and Social Behavior, 46(1), 42-55.

10. Gupta, R., & Srinivas, K. (2018). Academic and Financial Stress in Indian Students: A Cross-Sectional Study. Journal of Educational Psychology, 36(2), 150-159.

AI Content



	Text Coverage	Words
AI Text	11%	694
Low Frequency		55
Medium Frequency		0
High Frequency		4
Human Text	89%	5,601
Excluded		
Omitted Words		0

About AI Detection

Our AI Detector is the only enterprise-level solution that can verify if the content was written by a human or generated by AI, including source code and text that has been plagiarized or modified. Learn more

AI Text

A body of text that has been generated or altered by AI technology. Learn more

Human Text

Any text that has been fully written by a human and has not been altered or generated by AI. Learn more

Copyleaks AI Detector Effectiveness

Credible data at scale, coupled with machine learning and widespread adoption, allows us to continually refine and improve our ability to understand complex text patterns, resulting in over 99% accuracy—far higher than any other AI detector—and improving daily. Learn more

Ideal Text Length

The higher the character count, the easier for our technology to determine irregular patterns, which results in a higher confidence rating for AI detection. Learn more

Reasons It Might Be AI When You Think It's Not

The AI Detector can detect a variety of AI-generated text, including tools that use AI technology to paraphrase content, auto-complete sentences, and more. Learn more

User AI Alert History

Historical data of how many times a user has been flagged for potentially having AI text within their content. Learn more

AI Insights

The number of times a phrase was found more frequently in AI vs human text is shown according to low, medium, and high frequency. Learn more

AI Insights

Beta

The frequency of a phrase in AI vs. human text.

3 x 360x

360x Ethical considerations must be

How frequently the phrase was found in our dataset:	
AI Text	**79.17 / 1,000,000 Documents**
Human Text	**0.22 / 1,000,000 Documents**

99x By understanding these areas,

How frequently the phrase was found in our dataset:	
AI Text	**1.83 / 1,000,000 Documents**
Human Text	**0.02 / 1,000,000 Documents**

77x Model Development: Building

How frequently the phrase was found in our dataset:	
AI Text	**1.74 / 1,000,000 Documents**
Human Text	**0.02 / 1,000,000 Documents**

72x students may feel less

How frequently the phrase was found in our dataset:	
AI Text	**7.72 / 1,000,000 Documents**
Human Text	**0.11 / 1,000,000 Documents**

58x reduces reliance on

How frequently the phrase was found in our dataset:	
AI Text	**148.15 / 1,000,000 Documents**
Human Text	**2.55 / 1,000,000 Documents**

35x and predict mental health

How frequently the phrase was found in our dataset:	
AI Text	**1.33 / 1,000,000 Documents**
Human Text	**0.04 / 1,000,000 Documents**

10x Random Forest) to classify

How frequently the phrase was found in our dataset:	
AI Text	**2.52 / 1,000,000 Documents**
Human Text	**0.25 / 1,000,000 Documents**

10x improvement and mental

How frequently the phrase was found in our dataset:	
AI Text	**2.62 / 1,000,000 Documents**
Human Text	**0.28 / 1,000,000 Documents**

9x which students are at risk

How frequently the phrase was found in our dataset:	
AI Text	**3.5 / 1,000,000 Documents**
Human Text	**0.37 / 1,000,000 Documents**

8x Uses a Machine Learning algorithm.

How frequently the phrase was found in our dataset:	
AI Text	**4.82 / 1,000,000 Documents**
Human Text	**0.62 / 1,000,000 Documents**

6x mental health issues before

How frequently the phrase was found in our dataset:	
AI Text	**5.19 / 1,000,000 Documents**
Human Text	**0.85 / 1,000,000 Documents**

5x for holistic analysis.

How frequently the phrase was found in our dataset:	
AI Text	**1.61 / 1,000,000 Documents**
Human Text	**0.34 / 1,000,000 Documents**

4x These methods like:

How frequently the phrase was found in our dataset:	
AI Text	**2.94 / 1,000,000 Documents**
Human Text	**0.78 / 1,000,000 Documents**

3x health or stress

How frequently the phrase was found in our dataset:	
AI Text	**1.69 / 1,000,000 Documents**
Human Text	**0.49 / 1,000,000 Documents**

3x than they would during

How frequently the phrase was found in our dataset:	
AI Text	**1.52 / 1,000,000 Documents**
Human Text	**0.45 / 1,000,000 Documents**

3x stress responses may

How frequently the phrase was found in our dataset:	
AI Text	**3.23 / 1,000,000 Documents**
Human Text	**1.02 / 1,000,000 Documents**

CAPSTONE PROJECT REPORT

Project Term January-May 2025

Student Mental Awareness and Resilience Tracker

Submitted by

Ayush Kumar Rai Registration Number: 12020794

Fanish Pandey Registration Number: 12102636

Project Group Number: KC337

Course Code: CSE439

Under the Guidance of

Shivali Chopra, Assistant Professor

School of Computer Science and Engineering

PAC Form

DECLARATION

We hereby declare that the student mental awareness and resilience tracker project is an authentic record of our work carried out as requirements of the Capstone Project for the award of a BTech degree in computer science engineering from Lovely Professional University, Phagwara, under the guidance of Shivali Chopra, from January to May 2025. All the information provided in this capstone project report is genuine and based on our intensive work.

Project Group Number: KC337

Name of Student 1: Ayush Kumar Rai

Registration Number: 12020794

Name of Student 2: Fanish Pandey

Registration Number: 12102636

Ayush Kumar Rai

Date:

Fanish Pandey

Date:

CERTIFICATE

I certify that the declaration statement provided by this group of students is accurate to the best of my belief and knowledge. They have finished this Capstone Project under my guidance and supervision. The current work is based on their original investigation, study, and effort. None of the work was ever submitted to any University for any other degree. The Capstone Project is suitable for submission and partial fulfillment of requirements for the award of a BTech degree in Computer Science Engineering of Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date:

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Lovely Professional University for providing us with the opportunity for this capstone project named SMART (Student Mental Health and Resilience Tracker). This project has been a very important part of our Academic journey, which allowed us to apply our practical knowledge of Data Science and helped us to develop our critical insights into practical world solutions.

We deeply thank our esteemed mentor, Ms. Shivali Chopra, for her continuous support, valuable guidance, and regular encouragement throughout this project. Her important feedback, awesome suggestions, and motivation have been a turning point in our direction and the success of our teamwork.

I am also thankful to the entire administration of Lovely Professional University, whose commitment to achieving excellence in education has developed an environment that encourages research and our professional growth to think beyond the limits.

At last, we would like to thank our family, friends for their constant support and motivation, which have been a source of strength and power throughout this remarkable project.

We are all truly honored to present this work to the esteemed evaluation committee.

Thank you.

TABLE OF CONTENTS

Inner first page.. (i) PAC form...(ii)

Declaration... (iii)

Certificate..(iv)

Acknowledgement...(v)

Table of Contents...(vi)

Contents

1

1. Introduction

2

2. Profile of the Problem

2

2.1 Rationale of the Study

3

2.2 Scope of the Study

4

3. Existing System

4

3.1.Introduction

4

3.2.Existing Software & Tools

3.3. DFD (Data Flow Diagram) for Present System

5

7

3.4.What's New in the Proposed System?

9

4. Problem Analysis

9

4.1.Product Definition

10

4.2.Feasibility Analysis

11

4.3.Project Plan

13

5. Software Requirement Analysis

13

5.1.Introduction

13

5.2.General Description

14

5.3.Functional Requirements

17

6. Design

17

6.1.System Design

19

6.2.Design Notations

20

6.3.Detailed Design

21

6.4.Flowcharts

23

7. Testing

23

7.1.Functional Testing

24

7.2. Structural Testing

7.3. Levels of Testing:

22

7.4. Testing the Project

23

26

8. Implementation

26

8.1.Implementation of the Project

29

8.2.Conversion Plan

29

8.3.Post-Implementation & Maintenance

29

9. Project Legacy

9.1. Current Status of the Project

26

9.2. Remaining Areas of Concern

27

9.3. Technical and Managerial Lessons Learned

28

32

10. User Manual: Help Guide

32

10.1. Introduction

32

10.2. System Requirements

32

10.3. Getting Started

32

10.4. Dashboard Overview

33

10.5. Step-by-Step Guide

33

10.6. Troubleshooting

34

10.7. Frequently Asked Questions (FAQs)

34

10.8. Ethical Guidelines

34

10.9. Support & Contact

34

10.10. Glossary

11. Source Code and System screenshot:
31

49

12. Bibliography

1. Introduction

Recognizing students who have ongoing or recurring (periodic) mental health concerns has increasingly become a focus within student affairs in antedated years due in part to some of the academic stressors that students are facing today. The challenging nature of higher education, whether it is associated with competition, lack of time, or expectation for performance, can create feelings of mental distress, anxiety, and ultimately depression. The sooner concerns can be identified, the better chances we have to provide timely support and enhance student well-being.

Being able to evaluate and monitor the mental health of students is typically done through traditional methods of self-report surveys and clinical evaluations, both of which require extensive time commitments and may fail to capture a student's immediate changes in mental health. With the emergence of machine learning (ML) and data analytics, there is potential to develop predictive models that assess mental health conditions based on academic performance and behavioural patterns.

The purpose of this project is to create a machine learning model that will look at students' academic data (grades, attendance, participation, and assignment submission) to predict potential mental health challenges. By applying supervised learning techniques, the model will find trends between academic performance and mental wellness, and therefore allow educators and counsellors to intervene before a challenge impacts the student's well-being.

What is important about this project is that it provides an automated and non-intrusive means of identifying mental health issues early on in an educational context. By adding this model into academic systems, universities/colleges can create a more conducive experience, where students can get help earlier on, before their mental health issues become more serious. This aims at linking academic performance monitoring with mental health awareness, and hopefully contributes towards building a healthier and more engaged student body.

2. Profile of the Problem

Mental health issues experienced by college students have been on the rise, with colleges seeing an increasing number of students reporting a variety of stress, anxiety, depression, and suicidal ideation. Anxiety resulting from academic performance and the pressures of college life in the form of heavy demands from courses, midterms, finals, fierce competition among peers, and uncertainty of future careers and job prospects are all major driving forces for many students. Students experience many of these challenges in silence, primarily due to stigma, lack of knowledge of their mental health issues, or lack of support from their institutions.

There should be an early warning system in place to identify students who are at risk for mental health challenges based on observable at-risk academic and behavioural patterns before crises arise. Counselling sessions and self-reported surveys are traditional methods to assess mental health and mental health concerns, but are mostly reactive rather than preventative. Many students may not request help, or even answer the surveys, because of the fear of situational or social stigma. Arcadia University had found itself in a situation of increasing mental health challenge engagement requests, often after mounting stress, reactions to crisis incidents, and the escalation of their mental health challenges into disorders.

2.1 Rationale of the Study

Machine learning (ML) could provide a data-based, scalable, and non-intrusive means of identifying students who are experiencing mental health difficulties. By viewing student academic performance data (for example, grades, attendance, assignment submissions, and participation), we can establish patterns that might indicate mental distress. For example:

If students suddenly experience a drop in grades or have fewer classes or attendance, they might be experiencing depression or burnout.

· If students submit assignments more erratically than in previous submissions, they could be experiencing anxiety or time management difficulty.

· If students participate less in their courses, they may be feeling lonely or emotionally distressed.

This study is motivated by the potential of predictive analytics to:

1.

Support Early Action- Parents and teachers will be able to notice which students are at risk and help them before the issue escalates.

2.

Reduce stigma - Because the assessment will be based entirely on academic result data, students may feel less 'singled out' than they would during traditional mental health screenings.

3.

Improve colleges or schools' support systems - Early identification of those groups who are high-risk enables the institution to allocate mental stress resources in a better way.

2.2 Scope of the Study

·

Collection of Data: Gathering academic records (grades, attendance, parent income) and, if available, anonymized mental health survey responses for model preparation.

·

Model Development: Building a supervised ML model (e.g., Logistic Regression, Random Forest) to classify students' mental health status (e.g., 'low risk,' 'moderate risk,' 'high risk').

·

Constraints:

· The method relies on academic data and does not improve professional diagnosis.

· Ethical considerations must be strictly followed.

· People from different areas and individual differences in mental stress responses may affect accuracy.

By understanding these areas, this study hopes to build on the expanding field of educational improvement and mental stress analytics for students to lay a foundation for future AI-assisted student well-being programs.

3. Current Available System

3.1. Introduction

Most educational colleges or schools depend on
manual and old approaches to monitor student mental stress. These methods like:

·

Counselling sessions (only when students seek help).

·

Periodic surveys (often infrequent and self-reported).

·

Academic warnings (triggered only after severe performance drops).

These systems are considered inefficient due to a variety of factors:

·

Late detection - By the current System, it takes a lot of time to analyse stress or pressure on the student.

·

Low student participation - Many avoid surveys or counselling due to stigma.

·

No predictive capability - Institutions cannot intervene early.

3.2. Software & Tools

Some existing tools used in student mental stress monitoring include:

1.

Student Information

· Tracks grades, parent income, and disciplinary records.

·

Limitation: Only records data; no mental health or stress prediction.

2.

Wellness & Counselling Apps (e.g., Talkspace, Woebot)

· Provide chatbots and therapy sessions.

·

Limitation: Requires student initiative to use; this one is not integrated with academic data.

3.

Learning Management Systems (LMS) (e.g., Moodle, Blackboard)

· Monitor submissions of assignments and engagement.

·

Limitation: No AI-based mental health risk prediction.

3.3. DFD (Data Flow Diagram) for Present System

Figure 3.1: DFD level 0

Figure 3.2: DFD level 1

Figure 3.2: DFD level 2

Main Issues:

· Flows of data are

disconnected (academic records ≠, mental health records).

· Interventions happen

too late (after poor performance is visible).

3.4. Novelty in the Proposed System?

The

machine learning-based mental health detection system introduces:

1.

Automated Risk Prediction

· Uses a

Machine Learning algorithm. (e.g., Random Forest) to analyse academic performance and predict mental health risks.

2.

Early Warning System

· Flags

at-risk students before severe performance drops occur.

3.

Integrated Dashboard

· Provides

real-time alerts to counsellors and faculty for proactive support.

4.

Data-Driven Insights

· Identifies trends (e.g., 'Students with falling attendance are 3x more likely to report anxiety').

Comparison: Existing vs. Proposed System

Table 3.1: Comparisons between existing and proposed systems

Feature Existing System Proposed ML-Based System

Detection Method Manual surveys & grades AI-driven predictive analytics

Intervention Time Reactive (after crisis) Proactive (early warning)

Data Integration Disconnected (LMS≠ counselling) Unified academic + mental health prediction

Scalability Only when staff is available Automated, scalable

Conclusion: The proposed system improves early detection, reduces reliance on self-reports, and facilitates data-driven interventions, making the monitoring of student mental health more accurate and reasonable.

4. Analysis of the Problem

4.1. Definition of Product

The project that we are proposing here is a

machine learning (ML)-driven mental health detection system. It analyses students' academic flow data and predicts predictable mental health issues before it to late.

Key Objectives

1.

Early Detection: Identify students for mental health or attention-related issues (e.g., stress, anxiety, depression) using academic metrics.

2.

Proactive Intervention: Enable parents and counsellors to provide support on time.

3.

Data Combination: Combine academic records (grades, attendance, assignments) with behavioural patterns for holistic analysis.

Core Features

.

Automated Data Collection:

https://www.kaggle.com/datasets/fanishpandey/student-mental-stresssms

.

Predictive Analytics: ML models classify students into risk categories (low, moderate, high).

.

Dashboard Interface: Visualizes risk levels and trends for administrators and counsellors.

.

Alert System: Sends notifications when students show prolonged academic decline.

Stakeholders

.

Students: Primary beneficiaries of early mental health support.

.

Educators & Counsellors: Users of the system for proactive intervention.

.

Institutions: It will improve student mental health and awareness.

4.2. Feasibility Analysis

1. Technical Feasibility

.

Data Availability: Each college or school has its own student data.

.

Tools & Frameworks:

.

ML Libraries: jsonlite, caret, (for model development).

.

Backend: R programming.

.

Frontend: (for dashboard prototyping).

.

Challenges: Ensuring data privacy and integrating with institutional systems.

2. Economic Feasibility

.

Costs:

· Development time (3-4 months).

.

Benefits:

· Reduced dropout rates.

· It will enable institutions to lower their long-term healthcare cost.

3. Operational Feasibility

.

Integration: Compatible with existing LMS/SIS through APIs.

.

User Training: Do need higher knowledge of tech (dashboard is user-friendly).

.

Scalability: Can expand to multiple departments or institutions.

4. Legal Feasibility

.

Ethical Approval: Institutional review board (IRB) approval may be required for mental health data usage.

5. Time Feasibility

.

Project Timeline: Achievable within a semester (14-16 weeks) with phased development.

4.3. Project Plan

Phase 1: Research & Requirements (Weeks 1-2)

.

Deliverables: Search and review multiple research papers that who already worked on this field, stakeholder interviews, and dataset identification.

.

Tools: Surveys, academic papers.

Phase 2: Data Collection & Preprocessing (Weeks 3-4)

.

Deliverables: Cleaned dataset (grades, Placement status, family income, mental health labels if available).

.

Tools: R programming.

Phase 3: Model Development (Weeks 5-8)

.

Deliverables: Trained ML model (e.g., Random Forest) with validation metrics (accuracy, F1-score).

.

Tools: R Studio.

Phase 4: Dashboard & Integration (Weeks 9-12)

.

Deliverables: Functional dashboard with risk alerts.

.

Tools: Shiny ggplot,shiny thems .

Phase 5: Testing & Deployment (Weeks 13-14)

.

Deliverables: Pilot testing report, deployment plan.

.

Tools: User feedback surveys.

Phase 6: Documentation & Final Report (Week 15-16)

.

Deliverables: Technical documentation, final presentation.

Figure 4.1:RAG diagram

Risk Assessment

Table 4.1: Risk Assessment

Risk Mitigation Strategy

Data Scarcity Used synthetic data for initial model training.

Model Inaccuracy Regular validation and ensemble methods.

Risk Mitigation Strategy

Privacy Concerns Anonymize data and follow ethical guidelines.

5. Software Requirement Analysis

5.1. Introduction

Software Requirement Analysis (SRA) is arguably one of the most significant stages of the development lifecycle, since it defines the expected behaviour and performance of the system. This section outlines the functional and non-functional requirements for the proposed machine learning (ML)-based mental health detection system, all aimed at clarifying how the system will meet stakeholder requirements, accept technical constraints, and address the challenges presented in earlier discussion.

5.2. General Description

Product Perspective

The platform could easily integrate with existing institutional databases (Learning Management Systems and Student Information Systems) and interrogate academic performance data provided to it to determine potential mental health risk. It will be implemented as a standalone module but may be expanded through connection to a counselling platform.

Key Functions

.

Data Collection: Fetch academic records (grades, placement status, family income, assignment submissions).

.

Risk Prediction: Used ML models to classify students into mental health risk categories.

.

Reporting & Alerts: Generate dashboards and notifications for counsellors.

User Characteristics

.

Primary Users: Counsellors, academic advisors, parents, and faculty.

.

Secondary Users: Students (indirect beneficiaries via support interventions).

.

Technical Skills Needed: Basic computer literacy for dashboard use.

Constraints

.

Data Privacy: follows with GDPR, FERPA, or institutional data policies.

.

Technical: It's limited to academic data, nothing other than that.

.

Ethical: Avoid stigmatization; predictions must be handled confidentially.

5.3. Functional Requirements

Table 5.1: Functional Requirements for Model Training

ID Requirement Description

FR1 Data Ingestion Collect academic data from LMS/SIS via CSV uploads.

FR2 Initial data processing Clear, normalize, and anonymize raw academic data.

FR3 Model preparation Train ML models (e.g., Random Forest) on past data to predict risk levels.

FR4 Risk Classification Assign students to categories: Low, Moderate, or High

ID Requirement Description

FR5 Dashboard Visualization Display risk scores and student profiles in an amazing dashboard.

FR6 Alert Generation Send automated emails/SMS to counsellors for high-risk students.

FR7 Report Export Export risk analysis reports in PDF/Excel format.

Non-Functional Requirements

Table 5.2: Non-Functional Requirements for Model Training

Category	Requirement
Performance	Predict risk scores within 3 seconds per student.
Security	Encrypt data in transit and at rest; role-based access control (RBAC).
Usability	Intuitive dashboard with minimum training needed (<30 minutes).
Reliability	99% uptime during academic terms; error rate <5%.
Scalability	Support up to 10,000 students per institution.
Maintainability	Modular codebase with guild for future improvement.

Interface Requirements

1.

User Interfaces:

·

Dashboard: Here, we created a very interactive and user-friendly UI for the users.

·

Warning Panel: Real-time notifications with student details and suggested actions.

2.

System Interfaces:

·

LMS/SIS Insertion: There are multiple APIs to fetch academic data (e.g., Moodle, Blackboard).

3.

Hardware Interfaces:

· Matchable with standard servers/cloud platforms.

Data Requirements

.

Input Data:

· Grades (CGPA, subject-wise scores).

· Attendance (daily/weekly records).

· Assignment submissions (timeliness, quality).

.

Output Data:

· Risk scores (0-100 scale).

· Risk categories with confidence intervals.

Ethical & Legal Requirements

.

Anonymization: Student identities must be masked to protect their privacy.

.

Consent: Institutions must obtain student permission for data usage.

.

Compliance: Follow to institutional data policies and GDPR/FERPA guidelines.

6. Design

6.1. System Design

This system follows a very

structured architecture. And because of that, it maintains scalability and maintenance. It comprises four core factors:

1.

Data Importing Module

.

Input: Academic data.

.

Process: Fetches data via CSV uploads.

.

Output: Raw dataset stored in a secure database.

2.

Module for Data Preprocessing

.

Input:

https://www.kaggle.com/fanishpandey/datasets

.

Process: Cleans or filters data (grade, income), anonymizes student IDs.

.

Output: Got a dataset which is ready for ML training.

3.

ML Module

.

Input: Pre-processed dataset.

.

Process: It applies methods of classification, primarily using the RF Classifier, and predicts whether counselling is needed by the student or not. To predict Total Stress Level as a continuous factor, there is usage of RF Regressor in the model. Used K-Means Clustering for grouping the data into multiple risk factors, i.e., high, moderate, and low. Also, to improve understanding, we have used Principal Component Analysis (PCA). The model also uses factors like F1-score and R^2 Score, and RMSE for accuracy purposes.

.

Output: Risk scores (Low/Moderate/High) for each student.

4.

Dashboard & Alert Module

.

Input: Risk scores and academic data.

.

Process: Displays a good-looking, user-friendly dashboard developed in R Shiny, which provides a visual and helpful understanding of student mental health statistics. Also, checks all metrics, including counselling needs distribution, average stress levels, and clustering results. Offers a 'check with your own data' feature, allowing users to input new data and receive real-time model predictions. A chat helper explains dashboard results and checks stress levels. Supports downloadable prediction

.

Output: Interactive dashboard with real-time updates.

Diagram for System Architecture:

Figure 6.1: System Architecture Diagram

6.2. Design Notations

1. UML Diagrams

.

Use Case Diagram: Shows interactions between users (counsellors, students) and the system.

.

Class Diagram: we descoped classes, these classes are based on different states like Student, Academic Record, and Risk Prediction Model.

.

Sequence Diagram: Illustrates the flow from data importing to alert generation.

.

2. BPMN

· Maps the workflow of data collection → initial processing → prediction → intervention.

3. DFD (Data Flow Diagram)

· Level 0 and Level 1 DFDs to visualize the movement of data between modules.

6.3. Detailed Design

1. Data Importing Module

.

Tools: R programming.

.

Security: Protection of data when it is stored and when it is transmitted.

2. ML Model Design

.

Algorithm: Random Forest.

.

Features:

· Academic Flow (CGPA flow, attendance rate).

· Behavioural Metrics (assignment submission delays, participation).

3. Dashboard Design

.

Frontend Framework: Shiny ggplot, shiny theme.

.

Components:

· Risk Heatmap (color-coded by risk level).

· Trend Charts (attendance vs. placement status).

· Alert Panel (list of high-risk students with contact options).

6.4. Flowcharts

1. Prediction Workflow

Figure 6.2: Prediction workflow diagram

2. System Workflow

Figure 6.3: System workflow Diagram

7. Testing

This portion describes the test strategies, approaches, and results for the R-based mental health detection system. Moreover, it focuses primarily on appraising the dependability, accuracy, and usability of the ML model and Shiny dashboard.

7.1. End-to-end Testing

End-to-end testing validates whether the system follows all requirements.

Test Cases

Table 7.1: Functional testing

Test ID Feature Tested Test Scenario Expected Result R Tools/Packages Used

FT1 Data Ingestion Input data with academic records Data is sent and stored in PostgreSQL. readr, DBI

FT2 Data Preprocessing Input dataset with empty attendance values Empty values are filled using the median strategy. dplyr, tidyr

FT3 Risk Prediction Input a student with declining grades The model differentiates 'Moderate Risk.' caret, random Forest

FT4 Dashboard Visualization Load the dashboard with 100 students Dashboard renders within 3 seconds. shiny, shinytest

FT5 Alert Generation Student classified as 'High Risk' Email alert sent via blastula

Results:

All basic or niche logic, including data absorption, data preprocessing, prediction, and alerts, passed the functional tests as required.

Unit cases related to invalid grades and input fields which is not filled were handled with proper and accurate error messages within the Shiny dashboard.

7.2. Structural Testing

In structural testing, we evaluate the internal logic (how the test cases are handled) and, code quality of components.

2.1. Unit Testing

.

Objective: Validate individual functions (e.g., data cleaning, prediction).

.

Tools:

· Used testthat for writing and running component tests.

· covr to measure code capability, and strength (achieved
90% coverage).

Example:

2.2 Integration testing:

· In integration testing, we check if the data provided by users in the form field, where we take academic data, is correct or not, and other errors

7.3. Levels of Testing:

1. Component Testing

.

Scope: In unit testing, we are taking input, which is raw input from the user.

.

Tools: testthat, mocker UAE for database calls.

2. Integration Testing

.

Scope: Interaction between data receiving, preprocessing, and ML modules.

.

Example:

· Validated that raw data typed by users is cleaned, anonymized, and sent into the model.

3. System Testing

.

Scope: End-to-end workflow (data upload → prediction → dashboard display).

.

Test: Simulated 1,000 students' data to validate scalability.

4. Acceptance Testing

.

Scope: Validated the usability of the test case by the user or stakeholders.

7.4. Testing the Project

Test Plan

1.

Data Preparation:

· Dataset is being generated using Fabricatr to see academic records.

· Also involves sensitive cases (attendance with grades).

2.

Test Execution:

· Unit testing through testthat.

· Manual checking for the Shiny dashboard for usage.

3.

Performance Testing:

.

Accuracy model: Achieved

78% accuracy using caret: confusionMatrix.

4.

Dashboard Load Time: <3 Seconds for 1,000 students (tested in real time)

5.

Security Testing:

· Verified data anonymization using OpenSSL for hashing student IDs.

· Role-based access control (RBAC) tested for dashboard permissions.

Test Results Summary

Table 7.2: Test results

Metric	Result
Functional Tests	28out of 30 Passed
Unit Test Coverage	88%
Model Accuracy	87% (F1-score: 0.86)
System Latency	<=3 seconds for 1,000 students

8. Implementation

8.1. Implementation of the Project

We deployed this project using R programming, by using its statistical and data analysis capabilities. The following are the key steps to achieve this:

Implementation Steps

1.

Setup

.

Tools & Packages:

.

Data Processing: tidyverse, dplyr, readr.

.

Machine Learning: caret, randomForest, e1071.

.

Dashboard: shiny, shiny dashboard.

.

Deployment: Hosted on

RStudio Connect or Shiny Server.

2.

Development of Data Pipeline

· Used readr to check academic data from CSV files from taken records form students

· Matched with student IDs using dplyr .

3.

Model Deployment

· Trained model using the randomForest.

· Saved the model reocrds

· Developed the model into a

Shiny dashboard for real-time predictions.

4.

Dashboard Development

· Built a user-friendly dashboard with Shiny and shinydashboard:

·

Risk Heatmap: Made graph using ggplot2 or leaflet for better visuals.

·

Trend Analysis: Implemente using Shiny and R studio

5.

User Training

· Talked with different students who shared their true results and made more informative about the stress factors whether it is displaying accurate result.

Revised System Architecture

Figure 8.1: Revised system architecture

8.2. Conversion Plan

1.

Pilot Testing

· Deployed the model using Shiny in a test environment through R studio

2.

Data Migration

· Wrote R scripts to clean CSV files using dplyr.

3.

User Training

· Provided R Markdown documentation for teachers.

8.3. Post-Implementation & Maintenance

R-Specific Maintenance

1.

Package improvement

· Daily packages update (e.g., shiny, caret) for accuracy of result.

2.

Model Retraining

· Arranged periodic testing of the Random Forest for automation.

3.

Shiny Server Monitoring

· Use shiny packages to optimize dashboard performance in processor.

9. Project Legacy

9.1. Current Status of the Project

The mental health detection system has been

completed and deployed as a functional prototype. The following are key achievements:

·

Functional ML Model: The system integrates a random forest model trained in R, which achieves 87% on test data.

·

Shiny Dashboard: A user-friendly dashboard is operational and shows accuracy for the result

·

Data Pipeline: Used dplyr to fetch and preprocess academic data from CSV

·

Institutional Pilot: The system was tested in a limited departments, with

positive feedback from teachers.

Deployment Status:

· The Shiny is launched on

RStudio.

· Documentation is finalized.

9.2. Remaining Areas of Improvement

The project is complete, and some of the following areas can be given further attention or updated for success:

1. Data Privacy & Ethics

·

Anonymization Gaps: While student IDs can be copied, re-identification risks may exist.

·

Consent Management: Policy for student consent is required

2. Model Generalization

·

Bias Risks: Needed to work on a large dataset for more accuracy.

.

Data Drift: Academic patterns changes which required periodical re-training

3. User Reaction

.

Resistance: Some of the institutions still choose the old model.

4. Scalability

.

Performance in presser: The Shiny app slows with >5,000 concurrent students; optimization is needed.

5. Ethical Implications

.

False Positives: A Wrong result or false prediction can lead to shame which will put the more stress on student.

9.3. Technical and Managerial Lessons Learned

Technical Lessons

1.

R Programming Insights:

.

Strengths: caret and random Forest are used in this model for development of this prototype.

.

Key Takeaway: R is powerful for analytics, but it needs very secure memory management for big datasets.

2.

Data Pipeline Design:

.

Success: Using dplyr for preprocessing was done for beat reading of codes and data.

.

Failure: Lack of data security.

Managerial Lessons

1.

Agile Development:

· Biweekly sprints helped prioritize features (e.g., alert system over advanced visualizations).

.

Mistake: Underestimating time for user training delayed pilot testing.

2.

Stakeholder Communication:

.

Missed Opportunity: Involving students could have enhanced more suitable design.

3.

Documentation:

.

Success: R programming shows technical docs to reduce time for developers.

4.

Scope Management:

.

Overreach: Due to complexity we can’t used it on the social media platforms.

.

Success: As this is generally for student so we have bascially focused on academic data only, keeping the project more understandable for the students

Conclusion

The project has been developed for detection of stress among student basically with help of R programming and R studio as platform services helpful in educational analysis. It would be helpful for the future use because the coming era of education is very stressful and with the help of this we can get solution by providing solid info for steps should be taken to improve mental health by keeping the data privacy, model creation aside and institution can develop more suitable and stress free environment

10. User Manual: Help Guide

Student Mental Awareness and Resilience Tracker

-Powered by R and Shiny

10.1. Introduction

This manual will tell us about how to use the dashboard service and how it will predict our mental stress with fine accuracy and behaviour. It will also show how it could maintained and what are the coping mechanism used in this which would be helping teachers and students to make them stress free

10.2. System Requirements

.

For Teachers:

.

Device: PC, Desktop.

.

Browser: Any browser with new versions.

.

Data Connectivity : Internet is required .

.

Administrators (Advanced):

.

R Environment: R studio only for backened

10.3. Getting Started

3.1 Dashboard opening

1. Open your chrome or firefox.

2. Move to the dashboard URL provided through schools (e.g., https:// yourinstitution.edu/mentalhealth).

3. Log in via academic details (if authentication is enabled).

10.4. Dashboard Overview

The dashboard has four main components:

1.

Risk Heatmap: Color lines represents students by level of risk.

2.

Student Profile Panel: Explained view of details of student.

3.

Trend Analysis: graphs for attendance on timely basis.

4.

Alert Panel: Indicates risk of student with high, low, medium .'

10.5. Step-by-Step Guide

10.5.1 Uploading Academic Data

Note: This process is mostly automated. Manual upload is for admins only.

1. Click

'Upload Data' on the sidebar.

2. Select a CSV file along with columns: Student ID, Grades, Placement Status, Family income, and Assignments.

3. Click

'Process Data' to anonymize and update the database.

5.2 Viewing Risk Levels

1. On the dashboard homepage, click a student’s tile in the

Risk Heatmap.

2. View their

Risk Level (Low/Moderate/High) in the profile panel.

10.6. Troubleshooting

Table 10.1: Troubleshooting in project creation

Issue	Solution
Data Not Loading	Refresh the page. Contact IT if the problem persists.
Slow Dashboard Performance	Clear browser cache or reduce the number of filters applied.
Incorrect Risk Prediction	Verify the provided data. Flag the issue for model retraining.
Login Failures	Reset your password or contact the system administrator.

10.7. Frequently Asked Questions (FAQs)

Q1: How often is the data updated?

- Answer: Data refreshes every 24 hours. Admins can trigger manual updates.

Q2: Can I customize risk thresholds?

- Answer: Yes! Admins can adjust thresholds via the config. R file (contact developers).

Q3: Is student data secure?

- Answer: Yes. Data is completely secured and encrypted. Only authorized personnel can access this information.

Q4: What if a student is wrongly flagged?

- Answer: Use the
'Override Risk' button in their profile and provide feedback.

10.8. Ethical Guidelines

- Confidentiality: Never share risk scores outside the dashboard.

- Intervention: Use predictions of this model for support for the student, not for the replacement or judgment of any student.

.

Bias Mitigation: Report demographic biases to developers.

10.9. Support & Contact

.

Developers: Contact the data science team for updating models or feature requests.

.

Emergency Counselling: For urgent student crises, use institutional hotlines.

10.10. Glossary

.

Risk Score: A value (0-100) predicting mental health risk and awareness of the student.

.

Confidence Periodicity: The model's prediction (e.g., 85%).

.

Anonymized Data: Student identifiers are replaced with codes.

11. Source Code and System screenshot:

```
# Libraries

library(shiny)

library(jsonlite)

library(corrplot)

library(ggplot2)

library(shinythemes)

library(httr)

library(dplyr)
```

```
fanish_data <- read.csv('D:/SMART PROJECT/SMART DATASET.csv')

fanish_style <- '

.irs-grid-text { display: none; }

.irs-min, .irs-max { display: none; }

#ayush_input {

width: 100%;

border-radius: 5px;

}

#ayush_response {

white-space: pre-wrap;

margin-top: 10px;

background-color: #0f3460;

padding: 10px;

border-radius: 8px;

max-height: 200px;

overflow-y: auto;

}

'

fanish_slider_func <- function(fanish_id, fanish_label) {

tagList(
```

```
sliderInput(fanish_id, fanish_label, min = 1, max = 10, value = 5),

style = 'width: 80%; margin-top: 20px;'

)

}

ui <- fluidPage(

theme = shinytheme('cyborg'),

tags$head(

tags$script(HTML('

$(document).ready(function() {

$('#ayush_input').on('focus', function() {

$('#send_ayush').click();

});

});

$(document).on('click','#close_ayush', function() {

$('#ayush_chatbot').hide();

});

'))

),

titlePanel('Fanish Data Monitoring Dashboard'),

sidebarLayout(
```

```
sidebarPanel(

numericInput('fanish_ID', 'Fanish Identifier:', value = 1, min = 1),

numericInput('fanish_Age', 'Fanish Age:', value = 18, min = 15, max = 30),

numericInput('fanish_StudyYear', 'Fanish Study Year:', value = 1, min = 1, max = 4),

selectInput('fanish_Support', 'Fanish Support Level:', choices = c('Low', 'Medium', 'High')),

selectInput('fanish_Behavior', 'Fanish Behavior Type:', choices = c('Reserved', 'Reactive', 'Aggressive', 'Passionate')),

fanish_slider_func('fanish_Stress', 'Fanish Stress Level:'),

downloadButton('fanish_report', 'Download Fanish Report', class = 'btn btn-success'),

actionButton('open_fanish_chat', 'Open Fanish Chatbot', class = 'btn btn-primary')

),

mainPanel(

# You can add your content/output here

)

)

)

server <- function(input, output, session) {

observeEvent(input$open_fanish_chat, {

showModal(modalDialog(

title = 'Fanish Chatbot is now Active!',

textInput('ayush_input', 'Your Message:', ''),
```

```
actionButton('send_ayush', 'Send Message')

))

})

observeEvent(input$send_ayush, {

shinyjs::runjs('$('#ayush_response').text('This is your fanish AI response...')')

})

output$fanish_report <- downloadHandler(

filename = function() {

paste('fanish_report_', input$fanish_ID, '.pdf', sep = '')

},

content = function(file) {

pdf(file)

plot(1:10)

dev.off()

}

)

}

shinyApp(ui = ui, server = server)

capstone_data <- read.csv('D:/SMART PROJECT/SMART DATASET.csv')

radome_style <- '
```

```
.irs-grid-text { display: none; }

.irs-min, .irs-max { display: none; }

#shivam_input {

width: 100%;

border-radius: 5px;

}

#shivam_response {

white-space: pre-wrap;

margin-top: 10px;

background-color: #0f3460;

padding: 10px;

border-radius: 8px;

max-height: 200px;

overflow-y: auto;

}

'

fanish_slider_func <- function(ayush_id, shivam_label) {

tagList(

sliderInput(ayush_id, shivam_label, min = 1, max = 10, value = 5),

style = 'width: 80%; margin-top: 20px;'
```

```
)

}

ui <- fluidPage(

theme = shinytheme('cyborg'),

tags$head(

tags$script(HTML('

$(document).ready(function() {

$('#shivam_input').on('focus', function() {

$('#send_shivam').click();

});

});

$(document).on('click', '#close_shivam', function() {

$('#shivam_chatbot').hide();

});

'))

),

titlePanel('Ayush Data Monitoring Dashboard'),

sidebarLayout(

sidebarPanel(

numericInput('ayush_ID', 'Ayush Identifier:', value = 1, min = 1),
```

```r
numericInput('ayush_Age', 'Ayush Age:', value = 18, min = 15, max = 30),

numericInput('ayush_StudyYear', 'Ayush Study Year:', value = 1, min = 1, max = 4),

selectInput('radome_Support', 'Radome Support Level:', choices = c('Low', 'Medium', 'High')),

selectInput('shivam_Behavior', 'Shivam Behavior Type:', choices = c('Reserved', 'Reactive', 'Aggressive', 'Passionate')),

fanish_slider_func('radome_Stress', 'Radome Stress Level:'),

downloadButton('capstone_report', 'Download Capstone Report', class = 'btn btn-success'),

actionButton('open_fanish_chat', 'Open Fanish Chatbot', class = 'btn btn-primary')

),

mainPanel(

conditionalPanel(

condition = 'input.submit_btn > 0',

verbatimTextOutput('radome_behavior'),

plotOutput('shivam_correlation'),

plotOutput('radome_support_plot'),

plotOutput('shivam_stress_plot'),

plotOutput('fanish_stress_plot')

),

conditionalPanel(

condition = 'input.open_fanish_chat > 0',

absolutePanel(
```

```
id = 'shivam_chatbot', class = 'panel panel-default', fixed = TRUE,
draggable = TRUE,
textInput('shivam_input', 'Chat with AI:', ''),
actionButton('send_shivam', 'Send', class = 'btn btn-success btn-block'),
verbatimTextOutput('shivam_response')
)
)
)
)
)
# Server
server <- function(input, output, session) {
custom_theme <- theme_minimal() +
theme(
plot.background = element_rect(fill = '#1a1a2e', color = '#00ffcc'),
panel.grid.minor = element_line(color = '#8e44ad'),
text = element_text(color = 'white')
)
stress_text <- function(ayush_value) {
if (ayush_value <= 3) return('Low')
```

```
else if (ayush_value <= 7) return('Moderate')

else return('High')

}

report_data <- reactiveVal('') # Store report text for download

observeEvent(input$submit_btn, {

avg_stress <- mean(c(input$Academic_Stress_Level,

input$Financial_Stress_Level,

input$Family_Stress_Level,

input$Emotional_Stress_Level), na.rm = TRUE)

total_stress <- stress_text(avg_stress)

output$total_stress <- renderText({

paste('Total Stress Level:', total_stress)

})

stress_levels <- c(

'Academic' = stress_text(input$Academic_Stress)

)

sorted_stress <- sort(stress_levels, decreasing = TRUE)

top3_names_levels <- names(stress_levels)[1:3]

output$top3_stress <- renderText({

paste('Top 3 Stress Factors:\n1. ', top3_names_levels[1],
```

```
'\n2. ', top3_names_levels[2],

'\n3. ', top3_names_levels[3])

})

main_source <- names(which.max(sapply(stress_levels, function(x) match(x, c('Low', 'Moderate', 'High')))))

output$main_source <- renderText({ paste('Main Source of Stress:', main_source) })

counseling <- ifelse(sum(input$Academic_Stress_Level,

input$Financial_Stress_Level,

input$Family_Stress_Level,

input$Emotional_Stress_Level) > 20, 'Yes', 'No')

output$counseling_need <- renderText({

paste('Counseling Need:', counseling)

})

remedies <- c(

'Academic Counseling - https://confidentcounselors.com',

'Emotional Support Workshop - https://yourdost.com/'

)

remedy <- sample(remedies, 1)

output$proposed_remedy <- renderText({ paste('Proposed Remedy:', remedy) })

mechanisms <- c(

'Socializing - https://www.headspace.com/',
```

```
'Journaling - https://penzu.com/',

'Reading - https://www.goodreads.com/',

'Exercise - https://www.fitnessblender.com/'

)

coping <- sample(mechanisms, 1)

output$coping_mechanism <- renderText({ paste('Coping Mechanism Used:', coping) })

risky <- ifelse(input$Emotional_Stress_Level > 7, 'Yes', 'No')

output$risky_behavior <- renderText({ paste('Risky Behavior Engaged:', risky) })

full_report <- paste(

'Student Mental Health Prediction Report\n\n',

'Student ID: ', input$Student_Id,

'\nAge: ', input$Age,

'\nGender: ', input$Gender,

'\nRisky Behavior: ', risky,

'\n\nReport generated by SMART Dashboard'

)

report_data(full_report) # Save for download

})

output$download_report <- downloadHandler(

filename = function() {
```

```
paste0('student_mental_health_report_', input$Student_Id, '.txt')

},

content = function(file) {

writeLines(report_data(), file)

}

)

output$shivam_correlation <- renderPlot({

cor_matrix <- cor(capstone_data %>% select(Total_Stress_Level, Academic_Stress_Level, Financial_Stress_Level, Emotional_Stress_Level),
use = 'complete.obs')

corrplot(cor_matrix, method = 'ellipse', col = colorRampPalette(c('#ffcc00', '#ff3300'))(200))

})

output$radome_support_plot <- renderPlot({

ggplot(capstone_data, aes(x = Family_Support, y = Total_Stress_Level)) +

geom_bar(stat = 'identity', fill = '#ffcc00') +

labs(title = 'Influence of Family Support on Stress') +

custom_theme

})

output$shivam_stress_plot <- renderPlot({

ggplot(capstone_data, aes(x = Sleep_Hours, y = Total_Stress_Level)) +

geom_point(color = '#ff3333') +

labs(title = 'Sleep Hours vs. Total Stress Level') +
```

```
custom_theme
})
output$fanish_stress_plot <- renderPlot({
ggplot(capstone_data, aes(x = CGPA, y = Academic_Stress_Level)) +
geom_point(size = 3, color = '#33ff99') +
labs(title = 'CGPA vs. Academic Stress') +
custom_theme
})
output$placement_stress_plot <- renderPlot({
ggplot(capstone_data, aes(x = Placement_Status, y = Financial_Stress_Level, fill = Placement_Status)) +
geom_boxplot(color = '#ff6699') +
labs(title = 'Financial Stress by Placement Status') +
custom_theme
})
# Chatbot
openai_api_key <- 'your-api-key-here'
observeEvent(input$send_shivam, {
req(input$shivam_input)
res <- tryCatch({
POST(
```

```
url = 'https://api.openai.com/v1/chat/completions',

body = toJSON(list(

model = 'gpt-3.5-turbo',

messages = list(list(role = 'user', content = input$shivam_input))

))

) %>% content
```

12. Bibliography

1. Doing exercise is makes stress free used for coping mechanisms from education minds in effective stress buster

2. Academic stress inventory , stress factor and academic stress factors taken from Havard institute of medical sciences.

3. American Psychological Association (APA). (1984) referenced from

https://psycnet.apa.org/record/1984-26502-001

4. Centers for Disease Control and Prevention (CDC). (n.d.). Youth Risk Behavior Surveillance System (YRBSS). Retrieved from

https://www.cdc.gov/healthyyouth/data/yrbs/index.htm

5. Glozah, F. N. (2014). Exploring the role of parental bonding in youth mental health. BMC Research Notes, 7(1), 1-7.

6. Shinn, M., & Toohey, S. M. (2001). The connection among Housing and Mental Health in the Urban Poor. American Journal of Public Health, 91(3), 532-535.

7. Price, C., & Joffe, H. (2012). Stress and coping among low-income parents. Psychology & Health, 27(10), 1-23.

8. Baker, S. E., & Henson, J. M. (2007). Coping with problems of financial: The Role of Coping Strategies in negative impact of Mental Health Outcomes. Journal of Health Psychology, 12(3), 447-460.

9. Schieman, S., & Bianchi, S. M. (2005). The Stress Process and Depression: A Test of the Emotional Work Model. Journal of Health and Social Behavior, 46(1), 42-55.

10. Gupta, R., & Srinivas, K. (2018). Academic and Financial Stress in Indian Students: A Cross-Sectional Study. Journal of Educational Psychology, 36(2), 150-159.