

370 Assignment 1: Domain analysis and requirements specification

Diwash Bhattarai

A SMARTER APPROACH TO STUDENT WELL-BEING

1. INTRODUCTION

In today's demanding academic environment, students frequently encounter significant pressure from coursework, examinations, and various external responsibilities. This can lead to heightened stress, anxiety, and burnout, which detrimentally impact both academic performance and overall mental well-being. Many students grapple with effective time management, suffer from insufficient sleep, and feel overwhelmed by their workloads, often resulting in diminished mental health and reduced productivity. While traditional support systems exist, they often provide generic advice that fails to address the specific and unique challenges individual students face.

The Behavioral Optimization & Mental Wellness System aims to bridge this critical gap. By analyzing individual behavioral patterns and lifestyle data, the system is designed to provide students with personalized, actionable recommendations. These tailored insights will focus on optimizing study techniques, promoting effective wellness strategies, and offering practical time management tips, all derived from patterns of what has proven effective for others facing similar challenges. Unlike static advice, this system is envisioned to be dynamic, continuously adapting based on user feedback and evolving data to ensure ongoing relevance and efficacy. More than just a habit tracker, it empowers students to proactively manage their routines, mitigate stress, and cultivate sustainable habits essential for long-term academic success and personal well-being. The core of this system will be a predictive model capable of assessing a student's likelihood of experiencing depression, enabling timely and targeted support.

2. Glossary

Academic Pressure: The stress and anxiety experienced by students due to coursework, exams, and overall academic expectations.

Behavioral Optimization: The process of refining routines and habits to improve efficiency, reduce stress, and enhance overall performance and well-being.

Burnout: A state of emotional, physical, and mental exhaustion caused by excessive and prolonged stress.

CLI (Command Line Interface): A text-based interface used for interacting with a software program.

Depression: A mental health disorder characterized by persistent sadness, lack of motivation, and decreased interest in daily activities.

Feature Engineering: The process of creating new input variables (features) for a machine learning model from existing raw data to improve model performance or interpretability.

Functional Requirement: A statement that defines a specific function or behavior the system must perform.

Hyperparameter Tuning: The process of optimizing the parameters of a machine learning algorithm that are set prior to the learning process itself.

Lifestyle Habits: Behavioral factors such as sleep patterns, social engagement, physical activity, and diet, which may influence mental well-being.

Machine Learning (ML): A field of artificial intelligence that enables computer systems to learn from data and make predictions or decisions without being explicitly programmed for each task.

Non-Functional Requirement: A statement that describes a quality or constraint of the system, such as performance, security, or usability.

One-Hot Encoding: A process of converting categorical data variables into a numerical format suitable for machine learning algorithms, where each category becomes a new binary (0 or 1) feature.

Prediction Model: A machine learning algorithm that processes input data and forecasts an outcome, in this case, the likelihood of depression.

Random Forest Algorithm: A supervised learning model that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Recommendation System: A feature that provides personalized suggestions to students on how to reduce their risk of depression or improve well-being, based on data analysis and comparisons.

3. User Requirements

This section outlines the key needs of the users (students and potentially counselors)

UR1: Know My Current Mental Wellness Risk Level (Functional Requirement)

Description:

The system should tell the student's current risk level for experiencing depression (e.g., high, medium, or low) based on the information provided about their academics, lifestyle, and stress.

Rationale:

This core feature provides immediate awareness. Understanding the current risk level is the first step towards seeking help or making positive changes if needed. It allows for early identification of potential issues.

UR2: Get Advice on How to Improve My Well-being (Functional Requirement)

Description:

If the system indicates that a student might be at risk, or even if the student is doing okay, students want to receive personalized, simple, and actionable suggestions on what they can do to improve or maintain my mental well-being and study habits.

Rationale:

Simply knowing the risk isn't enough; the students need guidance. Personalized advice is more likely to be effective than generic tips, helping the students make practical changes to their daily routine or seek appropriate support.

UR3: Track Changes in My Risk Over Time (Functional Requirement)

Description:

A student will want to be able to see how their risk level changes if they use the system multiple times, perhaps by seeing a simple chart or history.

Rationale:

This helps the students understand if the changes they are making is having a positive impact or if certain periods (like exam season) increase their risk, allowing for better long-term planning and self-awareness.

UR4: Trust the System's Predictions (Non-Functional Requirement)

Description:

A user needs to feel confident that the risk level the system predicts is reasonably accurate and reliable.

Rationale:

If the predictions are often wrong, the user won't trust the system or its recommendations, making it useless. High accuracy is crucial for user trust and the system's credibility.

UR5: Get Results Quickly (Non-Functional Requirement)

Description:

A student, when provides input of their information, wants to see the risk prediction and any advice without waiting for a long time.

Rationale:

A slow system is frustrating and discourages use. Quick feedback makes the interaction smooth and encourages regular check-ins.

UR6: Know My Personal Data is Safe (Non-Functional Requirement)

Description:

A student needs to be sure that their sensitive personal information they share about their mental state, habits, and academics is kept private and secure.

Rationale:

Mental health data is highly personal. The students will only use the system if they trust that their information is protected from unauthorized access and won't be misused.

4. System Requirements

This section details the system's specific functionalities and qualities, mapped to the user requirements.

(A) Functional Requirements:

ID: SR1

User Requirement Mapping: UR1: Know My Current Mental Wellness Risk Level

System Requirement: Risk Prediction: Predict the student's depression risk level.

Description:

The system shall accept student input. Based on this input, it will utilize a trained Random Forest machine learning model to predict the student's likelihood of experiencing depression, classifying it into a risk category and providing an associated probability score.

Inputs & Source:

Inputs:

- Age (integer)
- Academic Pressure (integer, scale 1-5)
- CGPA (float, scale 0-10)
- Study Satisfaction (integer, scale 1-5)
- Suicidal Thoughts (integer, 0 for No / 1 for Yes)
- Work/Study Hours (float, hours per day)
- Financial Stress (integer, scale 1-5)
- Family History of Mental Illness (integer, 0 for No / 1 for Yes)
- Sleep Ordinal (integer, representing categories like <5h, 5-6h, 7-8h, >8h)
- Gender (integer, 0 for Female/Other, 1 for Male)
- Degree Type (integer, representing simplified categories like Science/Tech vs. Non-Science/Arts)

Source: Direct user input via the Command Line Interface (CLI)

Outputs & Destination:

Outputs:

- Predicted Risk Level (e.g., "HIGH RISK", "LOW RISK")
- Probability of High Risk (float, e.g., 0.75)

Destination: Displayed on the Command Line Interface (CLI) screen for the user.

Precondition:

- User must have entered valid data for all required input fields.
- The system must have a trained Random Forest model loaded and accessible.
- Input data values must be within their expected ranges and formats.

Algorithm/Process:

- Receive raw simplified inputs from the user via the CLI.
- Internally transform these simplified inputs into the full numerical feature vector expected by the Random Forest model. This includes:
- Calculating derived features (e.g., Total_Stress from Academic and Financial Pressure).
- Mapping categorical choices (e.g., user's simplified Degree Type, Age) to the appropriate one-hot encoded Age_Group_* and Degree_Type_* columns, ensuring other related one-hot encoded columns are set to 0.
- Ensuring all other one-hot encoded features (e.g., for specific cities, regions, detailed degrees if they were part of model training but not direct CLI input) are set to 0 as a baseline.
- Feed the complete, ordered feature vector to the loaded Random Forest model.
- The model outputs a probability for the positive class (e.g., likelihood of being 'High Risk').
- Apply a predefined threshold (e.g., 0.4 or 0.5) to the probability to determine the final risk category (e.g., "HIGH RISK" or "LOW RISK").

Postcondition:

- The user is presented with their predicted depression risk level (e.g., High/Low).
- The user is presented with the calculated probability score associated with the high-risk category.

ID: SR2

User Requirement Mapping: UR2: Get Advice on How to Improve My Well-being

System Requirement: Personalized Recommendations: Provide well-being and study habit suggestions.

Description:

Based on the predicted depression risk level and key input factors provided by the student, the system shall generate and display a set of personalized, actionable recommendations aimed at improving or maintaining mental well-being and academic habits.

Inputs & Source:

Inputs:

- Predicted Risk Level (from SR1: e.g., "HIGH RISK", "LOW RISK")
- Specific raw user input values used for conditional advice, such as:
- Academic Pressure value
- CGPA value
- Work/Study Hours value
- Financial Stress value
- Suicidal Thoughts status (Yes/No)
- Sleep Ordinal value
- Source: Output of SR1 (predicted risk) and the preprocessed feature vector containing user inputs.

Outputs & Destination:

Outputs: A list of textual recommendations.

Destination: Displayed on the Command Line Interface (CLI) screen for the user.

Precondition:

A depression risk prediction (SR1) must have been successfully generated for the user.

Algorithm/Process):

- Receive the predicted risk level and the user's (transformed) input features.
- Apply a rule-based logic engine:
- Provide general advice based on the overall predicted risk (High or Low).
- Conditional Advice (Examples):
- IF Suicidal Thoughts == YES: Prioritize and strongly advise seeking immediate professional help and provide crisis contact information.
- IF Academic Pressure > 3 (on a 1-5 scale): Suggest stress reduction techniques related to academics, time management, and seeking academic support.
- IF CGPA < 7.0: Offer tips for improving study habits, seeking academic resources, or forming study groups.
- IF Work/Study Hours > 8: Advise on the importance of work-life-study balance and strategies to manage workload.
- IF Financial Stress > 3 (on a 1-5 scale): Suggest resources for financial planning or support.
- IF Sleep_Ordinal indicates insufficient sleep: Recommend sleep hygiene practices.
- Compile the applicable recommendations into a list.

Postcondition:

The user is presented with a list of personalized and actionable recommendations.

ID: SR3

User Requirement Mapping: UR3: Track Changes in My Risk Over Time

System Requirement: Display Risk History and Current Prediction.

Description:

The system will maintain a simple history of the single user's past depression risk probabilities. When the user requests a new prediction, the system will first display their recent risk history before showing the current prediction. This allows the user to see trends in their self-assessed risk over time.

Inputs & Source:

Inputs for current prediction: Same as SR1 (Age, Academic Pressure, etc.).

Inputs for history display: Data from a local text file (risk_history.txt) storing past prediction dates and probabilities.

Source: User input via CLI for current prediction; local text file for history.

Outputs & Destination:

Outputs:

- A list of past risk probabilities with timestamps (if available).
- The current predicted risk level (e.g., "HIGH RISK", "LOW RISK").
- The current probability of high risk.

Destination: Displayed sequentially on the Command Line Interface (CLI) screen.

Precondition:

- User must enter valid data for all required input fields for a new prediction.
- The system must have a trained Random Forest model loaded and accessible.
- The risk_history.txt file may or may not exist. If it doesn't, no history will be shown for the first prediction.

Algorithm/Process:

- When the user initiates a prediction:
- Attempt to read risk_history.txt.
- If the file exists and contains data, display the last few entries (e.g., date and probability) to the user.
- Receive raw simplified inputs from the user via the CLI for the current assessment (as per SR1).
- Internally transform these simplified inputs into the full numerical feature vector (as per SR1).
- Feed the feature vector to the trained Random Forest model.
- Model outputs a probability for the positive class.
- Convert probability to a risk category.
- Display the current predicted risk level and probability.
- Append the current prediction's date/timestamp and probability to risk_history.txt.

Postcondition:

- The user is shown their recent risk history (if any).
- The user is presented with their current predicted depression risk level and probability.
- The current prediction result (date, probability) is saved to the local history file for future reference.

(B) Non-Functional Requirements:

ID	User Requirement (Description)	System Requirement (Verifiable)	Method for Verification
SR4	System Accuracy: Trust the system's predictions. (Maps to UR4)	The machine learning model (Random Forest) must achieve at least 80% accuracy in predicting depression risk when evaluated on a held-out, unseen test dataset.	Execute the C++ program's evaluation module on the test dataset. Calculate accuracy Ensuring the value is ≥ 0.80 .
SR5	Response Time: Get results quickly. (Maps to UR5)	The system, when running via the CLI, must process user inputs for a single prediction and display the risk level and initial recommendations within 5 seconds of the user submitting the final input.	Conduct 10 timed test runs using the CLI. For each run, manually time from the moment the last piece of required input is entered by the user to when the prediction and recommendations are displayed on the screen. Average the times. Ensuring the average is ≤ 5 seconds.
SR6	Data Privacy: Know my personal data is safe. (Maps to UR6)	For the current CLI single-user version, no personal identifiable information (like name) is explicitly requested or stored beyond the session, other than the anonymized risk history. The risk history file (risk_history.txt) will contain only timestamps, probabilities, and risk levels, without direct user identifiers.	Review the C++ code to confirm no personally identifiable information is written to any persistent storage, except for the risk_history.txt which contains anonymized prediction outcomes. Inspect the contents of risk_history.txt to ensure it only contains the defined anonymous data points.

