Project Report

Dedicated Mentoring System for Students (HEPro AI+)

Cover Page

Project Title:

Dedicated Mentoring System for Students (HEPro AI+)

Organization Name:

HEPro AI

Internship / Program Name:

HEPro AI/ML Internship

Student Name:

Ummu Abeeba.S

Mentor / Guide Name:

Mr. Argha Biswas

Submission Date:

26/02/2026

Acknowledgement

I would like to express my sincere gratitude to my mentor, **Mr. Argha Biswas**, for his continuous guidance, valuable feedback, and constant encouragement throughout the duration of this project. His insights helped me understand the practical aspects of designing and implementing an **AI-based mentoring system** and improved both my **technical and analytical thinking.**

I would also like to thank **HEPro AI** for providing me with the opportunity to work on this meaningful project as part of the **AI/ML internship program**. The learning experience, structured guidance, and real-world exposure provided by the organization helped me understand how **artificial intelligenc**e can be applied to solve **real educational challenges**.

I am grateful to the entire **HEPro AI+ team** for their support and resources, which made it possible to successfully complete this project. Finally, I would like to thank everyone who directly or indirectly contributed to the successful completion of this work.

Abstract

In modern educational environments, students often face multiple challenges related to academic performance, mental wellbeing, productivity, and career direction. Traditional mentoring systems are usually manual and reactive, making it difficult to identify students who need support at the right time. This project, **Dedicated Mentoring System for Students (HEPro AI+),** was developed to address this gap using a data-driven and intelligent approach.

The system integrates rule-based analytics and machine learning techniques to monitor student performance across multiple dimensions. Key performance indicators such as Academic Performance Score (APS), Wellness & Wellbeing Score (WWS), Productivity & Time Management Score (PTMS), and Career Readiness Score (CRS) are calculated and combined into a composite **Student Readiness Index (SRI).** Based on these scores, students are segmented into behavioral groups using **K-Means clustering**.

To ensure personalized support, a **cosine similarity-based mentor matching algorithm** assigns the most suitable mentor based on student needs, mentor expertise, mentoring style compatibility, and availability. The system also includes an intervention engine and a feedback loop to monitor improvement and identify ineffective matches.

The solution is implemented using **Python, Scikit-learn, Pandas, and Streamlit, and deployed as an interactive dashboard**. The key outcome of the project is a scalable AI-driven mentoring intelligence system capable of early risk detection, personalized mentor allocation, and continuous performance monitoring.

Public Deployment:
https://dedicatedmentoringsystemforstudents-ai-ml-project-hhw9jxeyqlfs.streamlit.app/

Table of Contents

- Action Layer
- Feedback Layer

4. Dataset Design

4.1 Dataset Philosophy

4.2 Student Dataset Structure

4.3 Mentor Dataset Structure

4.4 Interaction Dataset Structure

4.5 Data Preprocessing

4.6 Data Quality Controls

5. Feature Engineering

5.1 Feature Selection

5.2 Feature Normalization

5.3 Feature Encoding

5.4 Derived Features

6. Scoring System Design

6.1 Rule-Based Intelligence

6.2 Academic Performance Score (APS)

6.3 Wellness & Wellbeing Score (WWS)

6.4 Productivity & Time Management Score (PTMS)

6.5 Career Readiness Score (CRS)

6.6 Student Readiness Index (SRI)

6.7 Risk Classification Logic

7. Machine Learning Design

7.1 Role of ML in the System

7.2 Student Segmentation Model

- Model Type: K-Means Clustering
- Feature Space
- Training Process
- Cluster Interpretation

7.3 Similarity-Based Mentor Matching

- Feature Vector Design
- Similarity Metrics
- Matching Logic

7.4 Model Evaluation Strategy

8. Intervention & Decision System

8.1 Intervention Philosophy

8.2 Rule Engine Design

8.3 Intervention Levels

- Automated
- Mentor-Assisted
- Escalation

8.4 Decision Flow

9. Mentor Playbook Framework

9.1 Purpose of Playbooks

9.2 Playbook Structure

9.3 Core Playbooks

10. System Implementation

10.1 Technology Stack (Python-Based)

10.2 Backend Architecture

10.3 Data Processing Pipeline

11. Results & Outputs

12. Evaluation

13. Discussion

14. Ethical & Social Considerations

1. Introduction

1.1 Background

Educational institutions today manage a diverse group of students with different academic abilities, personal challenges, and career goals. While mentoring plays a critical role in student success, most traditional mentoring systems rely on manual observation and periodic reviews. This approach makes it difficult to identify early warning signs such as declining performance, high stress levels, low productivity, or lack of career clarity.

With the increasing availability of student-related data, there is an opportunity to use artificial intelligence and data analytics to support mentoring decisions. AI-based systems can continuously monitor multiple aspects of student performance, identify patterns, and provide timely recommendations. Such systems can help mentors focus their attention where it is needed the most and improve overall student outcomes.

The **HEPro AI+ Dedicated Mentoring System** is designed to act as an intelligent support system that assists institutions and mentors in identifying student needs, segmenting them based on behavior, and providing personalized guidance.

1.2 Motivation

During the study of student performance patterns, it was observed that academic scores alone do not provide a complete picture of a student's situation. Many students with good academic performance may struggle with stress, low engagement, or uncertainty about their career path. Similarly, some students may show high engagement but poor academic results.

This project was motivated by the need to create a **holistic student monitoring system** that considers multiple dimensions such as academics, wellbeing, productivity, and career readiness. Another important motivation was to move from a reactive mentoring approach to a **proactive and preventive system** that identifies risks early.

The project also aims to demonstrate how a combination of **rule-based intelligence and machine learning** can be used to build a practical and scalable solution for real-world educational environments.

1.3 Problem Definition

The major challenges addressed in this project are:

- Lack of early identification of at-risk students
- Inefficient manual mentor allocation
- Absence of a structured system to monitor student wellbeing and productivity
- Difficulty in providing personalized mentoring at scale
- No continuous feedback mechanism to evaluate mentoring effectiveness.

The problem can be defined as:
How to design an intelligent system that can assess student readiness, identify risk levels, segment students based on behavioral patterns, and assign the most suitable mentor for effective intervention?

1.4 Objectives of the Project

The main objectives of the HEPro AI+ system are:

- To design a multi-dimensional student performance evaluation framework
- To calculate composite readiness using the Student Readiness Index (SRI)
- To segment students into meaningful behavioral groups using machine learning
- To implement a cosine similarity-based mentor matching system
- To recommend personalized interventions based on student needs
- To build an interactive dashboard for monitoring and decision support
- To create a feedback loop for continuous system improvement

1.5 Scope of the Project

The system is designed as a prototype that can be extended for real institutional use. It covers the complete pipeline from data generation and preprocessing to scoring, clustering, mentor matching, and visualization.

The current implementation includes:

- Dataset of 50 students with multiple performance attributes
- Rule-based scoring for APS, WWS, PTMS, CRS, and SRI
- Student segmentation using K-Means clustering
- Mentor assignment using cosine similarity
- Feedback monitoring for improvement tracking
- Streamlit-based deployment for interactive use.

The architecture is modular and can be scaled to handle larger datasets and real-time institutional data.

1.6 Limitations

Although the system demonstrates a complete AI-driven mentoring workflow, certain limitations exist:

- The dataset is synthetically generated and may not fully represent real institutional data
- The number of students and mentors is limited for demonstration purposes
- The clustering model uses a fixed number of clusters (K = 3)
- Feedback data is simulated based on predefined logic
- Real-time data integration is not implemented in the current version.

These limitations provide opportunities for future enhancement and real-world deployment.

2. Literature Review / Conceptual Foundation

2.1 Traditional Mentoring Systems

Traditional mentoring systems in educational institutions are usually based on periodic meetings, manual observation, and academic performance reviews. In most cases, mentors are assigned based on availability rather than student needs. This approach often results in delayed identification of problems, especially those related to stress, productivity, or career uncertainty.

Another limitation of traditional systems is that they primarily focus on academic performance. However, student success depends on multiple factors such as mental wellbeing, time management, engagement, and career clarity. Without a structured data-driven mechanism, mentors may not be able to detect early warning signs or provide personalized support.

Due to these limitations, there is a growing need for intelligent mentoring systems that can continuously monitor student behavior and support early intervention.

2.2 Role of AI in Education

Artificial Intelligence has emerged as a powerful tool in the education sector. AI-based systems can analyze large volumes of student data, identify hidden patterns, and provide actionable insights. In the context of student mentoring, AI can be used for:

- Early risk detection
- Performance monitoring
- Behavioral pattern analysis

- Personalized recommendations
- Resource optimization.

Machine learning techniques such as clustering and similarity analysis help in understanding student behavior beyond traditional metrics. These techniques enable institutions to shift from reactive decision-making to proactive student support.

2.3 Data-Driven Student Support Systems

Modern educational analytics emphasizes the importance of multi-dimensional student evaluation. Instead of relying only on grades, data-driven systems consider multiple indicators such as:

- Academic consistency
- Attendance and participation
- Stress and wellbeing
- Productivity and distractions
- Engagement levels
- Career readiness.

The dataset used in this project reflects realistic student behavior patterns across these dimensions. For example:

- High stress is often associated with lower productivity
- High engagement does not always guarantee high academic performance
- Some high-performing students lack career clarity
- Behavioral patterns are unevenly distributed, similar to real student populations .

Such multi-dimensional modeling helps mentors understand the actual needs of students and provide targeted support.

2.4 Hybrid AI Systems (Rule-Based + ML)

Pure machine learning models may lack interpretability, while rule-based systems alone cannot capture complex patterns. Therefore, this project adopts a hybrid approach combining both methods.

The system first uses rule-based logic to calculate individual performance scores:

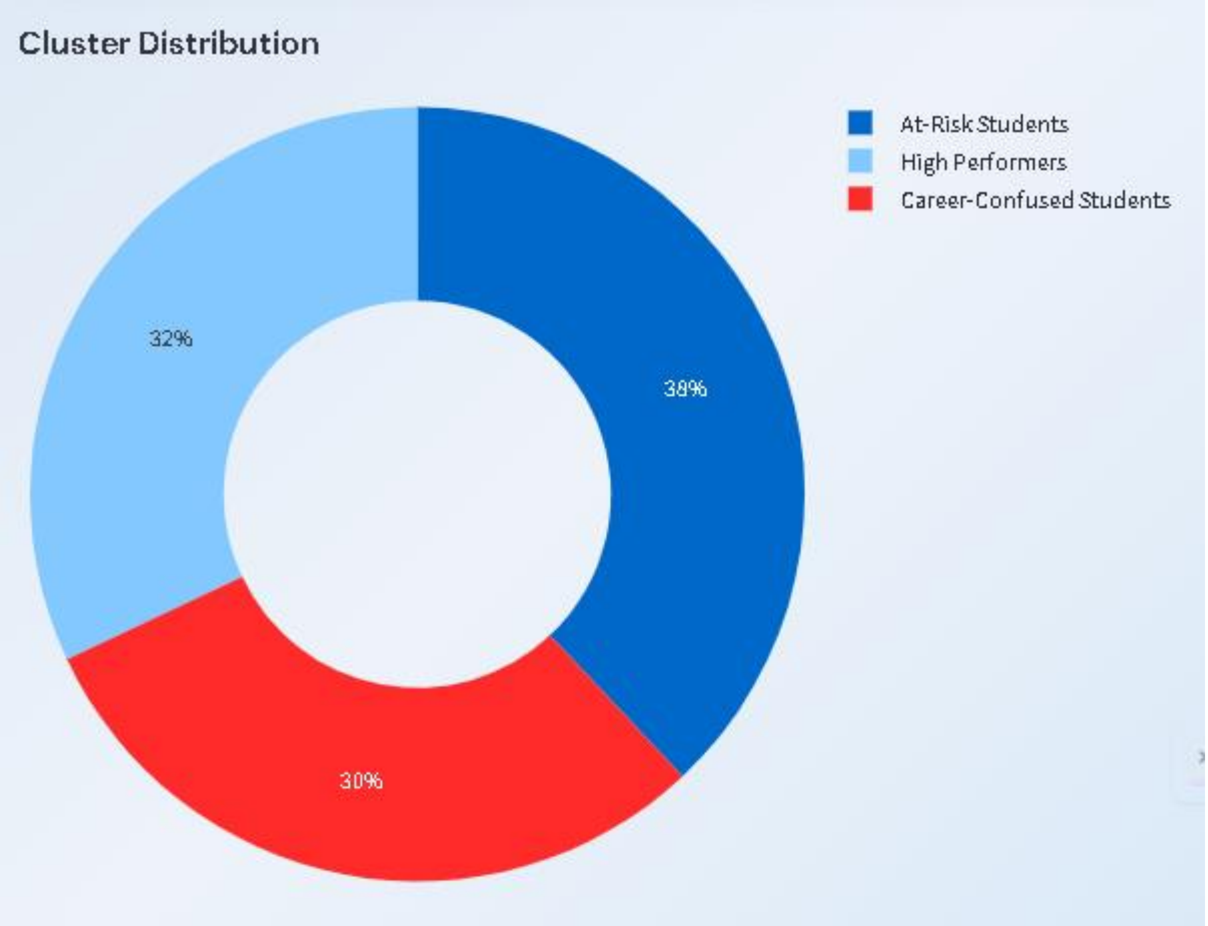
- Academic Performance Score (APS)
- Wellness & Wellbeing Score (WWS)
- Productivity & Time Management Score (PTMS)
- Career Readiness Score (CRS)

These scores are combined into a composite **Student Readiness Index (SRI).**

APS	WWS	PTMS	CRS	SRI	Category	Cluster	Cluster_Label
66.8	41.8	54.4	77.6	60.77	Blue	0	At-Risk Students
81.4	77.0	56.0	81.2	75.17	Green	1	High Performers
73.2	46.2	75.8	28.8	55.87	Yellow	2	Career-Confused Students
81.5	82.0	34.2	49.0	64.04	Blue	1	High Performers
61.4	45.4	64.8	49.8	55.18	Yellow	2	Career-Confused Students

Machine learning is then applied to identify behavioral segments using **K-Means clustering.** This hybrid design ensures both interpretability and pattern discovery. According to the cluster analysis, students are grouped into:

- At-Risk Students
- High Performers
- Career-Confused Students



This combination improves decision accuracy and enables personalized mentoring strategies.

3. System Overview

3.1 Project Architecture

The HEPro AI+ system is designed as an end-to-end intelligent mentoring pipeline that processes student data and generates actionable decisions. The architecture follows a layered design to ensure modularity and scalability.

The system consists of five major layers:

1. Data Layer
2. Intelligence Layer
3. Decision Layer
4. Action Layer
5. Feedback Layer.

Each layer performs a specific function in the mentoring workflow, starting from raw data processing to continuous improvement based on outcomes.

3.2 System Workflow

The overall workflow of the system is as follows:

1. Student data is generated and stored in structured format
2. Rule-based scoring calculates APS, WWS, PTMS, CRS, and SRI
3. Machine learning (K-Means) segments students into behavioral clusters
4. Cosine similarity is used to match students with suitable mentors
5. The system recommends appropriate intervention levels
6. Mentor feedback is collected and analyzed
7. Matching issues are identified and used for system improvement.

This workflow enables continuous monitoring and adaptive mentoring.

3.3 Module Breakdown

1. **Data Layer**

The Data Layer is responsible for storing and managing student information. The dataset includes 50 students with attributes such as:

- GPA, attendance, assignment completion

- Stress level, sleep hours, mental wellbeing
- Productivity score and distractions
- Career clarity and skill readiness
- Engagement score

The dataset was generated using Python to simulate realistic variations in student behavior. This layer ensures that all relevant dimensions affecting student success are captured.

2. **Intelligence Layer**

The Intelligence Layer performs analytical processing. It includes two major components:

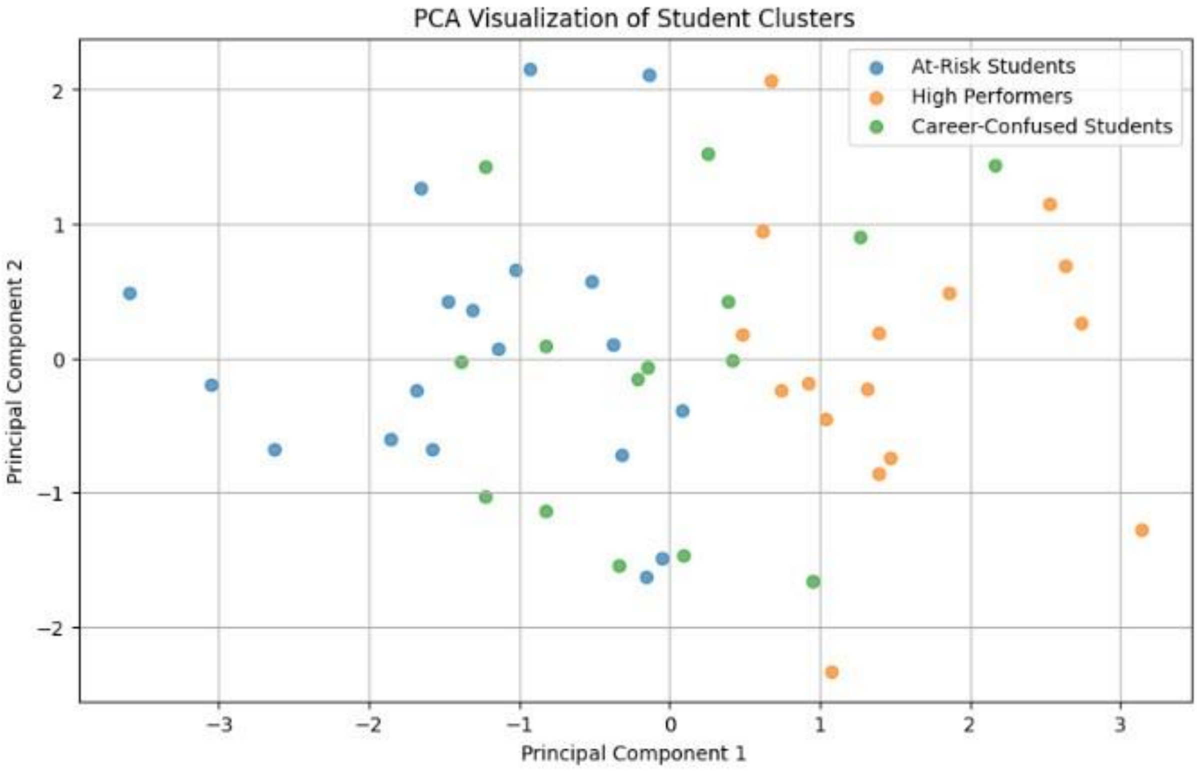
➢ **Rule-Based Scoring Engine**

- APS: Academic performance evaluation
- WWS: Stress, sleep, and mental health assessment
- PTMS: Productivity and distraction analysis
- CRS: Career clarity and skill readiness
- SRI: Overall readiness score

➢ **Machine Learning Engine**

- Standardization of features
- K-Means clustering (K = 3)
- Silhouette analysis for cluster quality
- PCA for visualization

The clustering results show meaningful separation between student groups, confirming the effectiveness of segmentation.



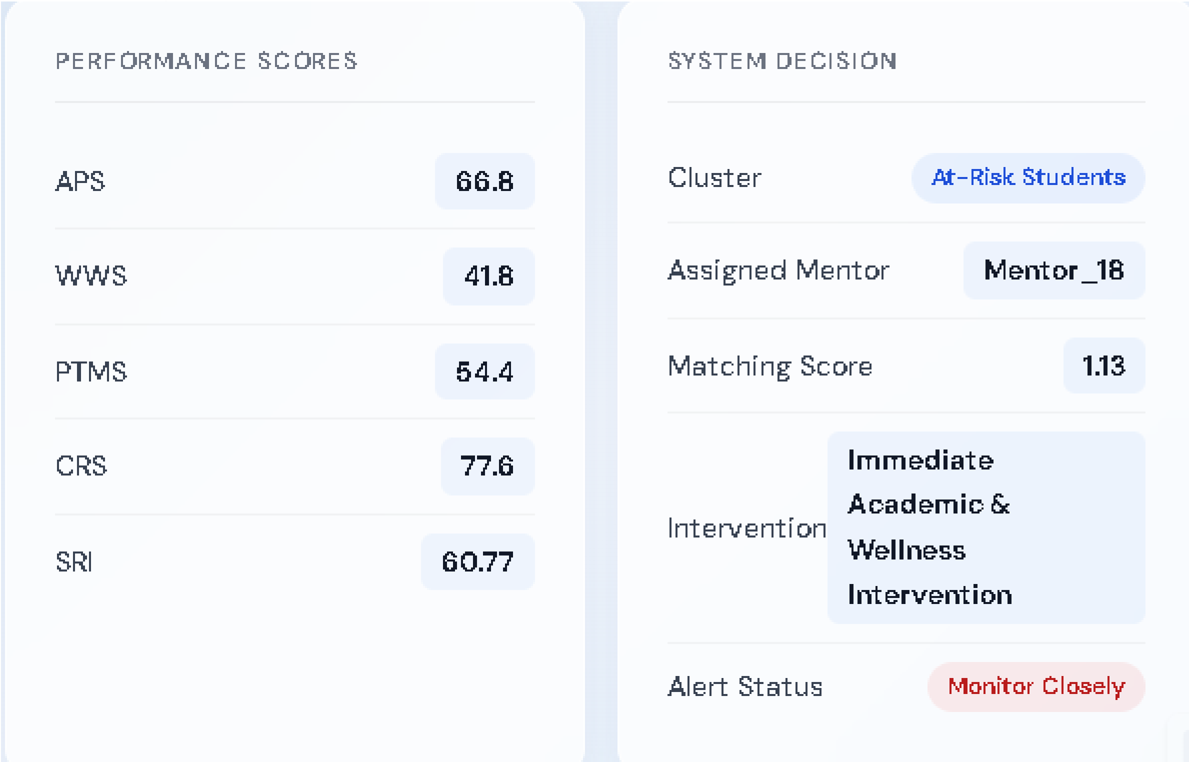
The PCA visualization confirms that the clusters are clearly separated, indicating that the selected features effectively represent student behavior.

3. **Decision Layer**

The Decision Layer converts analytical insights into actionable decisions.

This layer performs:

- Cluster label assignment
- Risk classification based on SRI
- Mentor recommendation using cosine similarity
- Style compatibility adjustment
- Mentor availability and capacity checks



Cosine similarity compares student feature vectors (APS, WWS, PTMS, CRS) with mentor expertise vectors to ensure the best match. This approach enables personalized and efficient mentor allocation.

4. Action Layer

Based on the cluster and risk level, the system recommends interventions:

- At-Risk Students: Immediate academic and wellness support
- Career-Confused Students: Career guidance and skill planning
- High Performers: Advanced mentoring and leadership development

The system also generates alert statuses such as:

- High Risk Alert
- Monitor Closely
- Stable

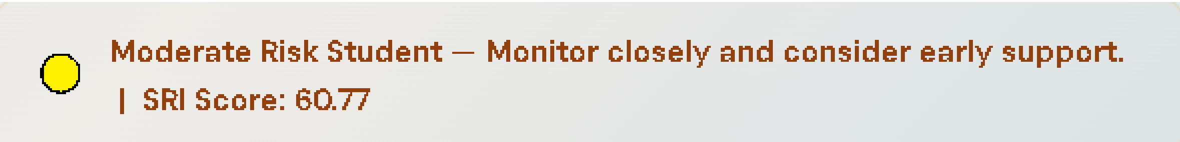
These actions help mentors prioritize their efforts effectively.

5. Feedback Layer

The Feedback Layer ensures continuous improvement of the system. After mentoring sessions:

- SRI improvement is recorded
- Mentor ratings are collected
- Sessions completed are tracked

If a student shows low improvement or poor mentor rating, the system flags a matching issue for review. This feedback loop helps refine matching weights and improve long-term effectiveness.



The deployed dashboard integrates these components and provides real-time insights for monitoring and decision-making.

Public Deployment:
https://dedicatedmentoringsystemforstudents-ai-ml-project-hhw9jxeyqlfs.streamlit.app/

4. Dataset Design

4.1 Dataset Philosophy

The dataset used in the HEPro AI+ system was designed to represent real student behavior in a structured and multi-dimensional manner. Instead of focusing only on academic performance, the dataset captures multiple factors that influence student success, including wellbeing, productivity, engagement, and career readiness.

The primary objective behind the dataset design was to simulate realistic student variations and behavioral patterns that mentors typically observe in educational environments. By modeling these dimensions together, the system is able to identify hidden risk factors and provide targeted mentoring interventions.

The dataset reflects key real-world characteristics such as:

- Variation in academic consistency
- Relationship between stress and productivity
- High engagement but low academic performance cases
- Strong academic students with unclear career goals
- Uneven distribution of behavioral patterns

This multi-dimensional design supports holistic student assessment and improves decision-making accuracy.

4.2 Student Dataset Structure

The student dataset contains 50 records, each representing an individual student. The dataset was generated using Python with controlled randomization to maintain realistic value ranges.

Each student record includes the following categories:

Basic Information

- student_id (S001–S050)
- age (18–25)
- program (B.Tech, B.Sc, MBA, BCA, MCA)
- semester (1–8)

Academic Attributes

- GPA (0–10)
- Attendance (0–100%)
- Assignment completion (0–100%)

Wellbeing Attributes

- Stress level (1–10)
- Sleep hours (0–10)
- Mental wellbeing (1–10)

Productivity Attributes

- Productivity score (1–10)
- Distractions (1–10)

Career Attributes

- Career clarity (1–10)
- Skill readiness (1–10)

Engagement

- Engagement score (0–100)

These ranges were selected to match realistic student behavior observed in educational systems.

student_id	age	program	semester	gpa	attendance	assignments_completion	stress_level	sleep_hours	mental_wellbeing	productivity_score	distractions	career_clarity	skill_readiness	engagement_score
S001	24	MBA	5	5.9	72	72	4	6.9	1	4	4	7	9	68
S002	21	BCA	1	7.9	86	80	1	4.5	8	10	10	10	8	46
S003	22	BCA	5	6.3	81	79	5	7.1	3	10	5	5	1	24
S004	24	B.Tech	7	7.4	82	91	4	9	10	3	9	3	6	65
S005	20	MBA	5	5.3	50	84	8	4.7	7	10	8	7	5	9
S006	25	MCA	1	6.7	68	56	8	6.6	10	1	3	2	6	55
S007	22	MBA	1	6.9	51	65	7	8.4	5	8	1	9	10	29
S008	22	MCA	7	8.2	93	75	3	7.7	10	5	3	10	5	4
S009	24	B.Tech	1	9.2	75	97	1	7.5	5	4	4	10	6	32
S010	19	B.Sc	8	7.7	81	98	1	7.5	7	8	2	1	5	64
S011	20	BCA	1	7.4	55	51	3	5.8	9	7	1	6	5	17
S012	24	B.Tech	4	5.8	81	50	6	5.5	5	2	7	7	4	95
S013	20	BCA	8	5.3	53	97	7	8	1	1	8	8	3	48

4.3 Mentor Dataset Structure

The mentor dataset represents operational mentoring conditions and includes:

- Mentor ID and Name
- Expertise Domain (Academic / Wellness / Career)
- Specialization
- Mentoring Style (Supportive / Strategic / Structured)
- Years of Experience
- Availability (hours per week)
- Maximum student capacity
- Current assigned load

mentor_id	mentor_name	expertise_domain	specialization	mentoring_style	experience_years	ability_hours_per_	max_students	current_load
M001	Mentor_1	Career	Interview Preparati	Structured	13	13	10	1
M002	Mentor_2	Career	Resume Building	Strategic	10	10	9	2
M003	Mentor_3	Wellness	Work-Life Balance	Supportive	10	17	11	1
M004	Mentor_4	Academic	Mathematics	Supportive	8	18	14	0
M005	Mentor_5	Career	Resume Building	Strategic	12	17	8	3
M006	Mentor_6	Career	Interview Preparati	Structured	5	10	8	4
M007	Mentor_7	Academic	Computer Science	Structured	14	19	7	1
M008	Mentor_8	Academic	Computer Science	Structured	4	9	12	3
M009	Mentor_9	Academic	Data Science	Supportive	3	9	7	3
M010	Mentor_10	Wellness	Mental Health	Supportive	15	20	7	1
M011	Mentor_11	Wellness	Mental Health	Supportive	6	19	13	5
M012	Mentor_12	Career	Interview Preparati	Structured	10	10	7	4
M013	Mentor_13	Wellness	Stress Managemen	Structured	11	18	6	0
M014	Mentor_14	Career	Software Engineerir	Structured	14	13	13	2

This structure ensures that mentor allocation is not only based on expertise but also considers operational feasibility such as availability and workload balance.

4.4 Interaction Dataset Structure

The system generates interaction and feedback datasets to monitor mentoring effectiveness:

matching_feedback.csv

- Initial SRI
- Final SRI
- SRI improvement
- Mentor rating
- Sessions completed

student_id	mentor_name	cluster	matching_score	initial_sri	final_sri	sri_improvement	mentor_rating	sessions_completed
S001	Mentor_18	At-Risk Students	1.133	60.77	69.77000000000001	9.000000000000007	5	7
S002	Mentor_1	High Performers	1.015	75.17	76.17	1	2	2
S003	Mentor_7	Career-Confused Stuc	0.996	55.87	59.87	4	3	3
S004	Mentor_7	High Performers	1.005	64.04	65.04	1	2	2
S005	Mentor_5	Career-Confused Stuc	1.025	55.18	58.18	3	3	5
S006	Mentor_10	At-Risk Students	1.112	53.89	62.89	9	5	6
S007	Mentor_1	High Performers	1.1	67.57	70.57	3	3	3
S008	Mentor_7	High Performers	1.056	73.61	74.61	1	2	3

matching_issues.csv

- Low mentor rating cases
- Low improvement cases
- Matching score
- Issue description

student_id	mentor_name	cluster	matching_score	sri_improvement	mentor_rating	issue
S002	Mentor_1	High Performers	1.015	1	2	Low effectiveness â€“ review matching weights
S004	Mentor_7	High Performers	1.005	1	2	Low effectiveness â€“ review matching weights
S008	Mentor_7	High Performers	1.056	1	2	Low effectiveness â€“ review matching weights
S020	Mentor_17	Career-Confused Stuc	1.1	2	2	Low effectiveness â€“ review matching weights
S026	Mentor_1	High Performers	1.058	2	2	Low effectiveness â€“ review matching weights
S028	Mentor_13	High Performers	1.085	1	2	Low effectiveness â€“ review matching weights
S033	Mentor_5	Career-Confused Stuc	1.066	2	2	Low effectiveness â€“ review matching weights
S036	Mentor_1	High Performers	1.048	2	2	Low effectiveness â€“ review matching weights
S040	Mentor_14	High Performers	0.981	2.000000000000007	2	Low effectiveness â€“ review matching weights
S044	Mentor_14	High Performers	1.1	2	2	Low effectiveness â€“ review matching weights

These datasets support continuous system learning and help identify ineffective mentor-student matches.

4.5 Data Preprocessing

Before analysis, the dataset undergoes several preprocessing steps:

- Handling of value ranges and normalization
- Conversion of GPA to percentage scale
- Inversion of negative indicators such as stress and distractions
- Feature scaling for machine learning using MinMax normalization
- Verification of missing values and data consistency

These preprocessing steps ensure that all features are comparable and suitable for scoring and clustering.

4.6 Data Quality Controls

To maintain reliability, the following quality checks were applied:

- All numerical features constrained within realistic ranges
- Logical consistency validation (e.g., high stress should reduce wellbeing)
- Statistical summary checks (mean, min, max)
- Manual review of sample records

These controls ensure that the dataset behaves realistically and supports meaningful analytical outcomes.

5. Feature Engineering

5.1 Feature Selection

Feature selection was based on the key dimensions that influence student success:

- Academic performance
- Mental and physical wellbeing
- Productivity and time management
- Career readiness
- Engagement level

The selected features were grouped into four major scoring dimensions:

- APS
- WWS
- PTMS
- CRS

These features were chosen to balance academic and behavioral indicators.

5.2 Feature Normalization

Since the dataset contains features with different scales (e.g., GPA 0–10, attendance 0–100), normalization was applied.

Normalization steps included:

- Converting GPA to percentage scale
- Scaling 1–10 values to 0–100
- Applying MinMax scaling for clustering and similarity analysis

This ensures equal contribution of each feature during machine learning and cosine similarity computation.

5.3 Feature Encoding

Categorical variables such as:

- Program
- Mentor domain
- Mentoring style

were encoded into numerical or vector representations where required for matching and analysis.

For mentor matching, domain expertise was converted into feature-aligned vectors corresponding to APS, WWS, PTMS, and CRS dimensions.

5.4 Derived Features

Several important features were derived from raw data:

- APS (Academic Performance Score)
- WWS (Wellness & Wellbeing Score)
- PTMS (Productivity & Time Management Score)
- CRS (Career Readiness Score)
- SRI (Student Readiness Index)

These derived features act as the core intelligence inputs for clustering, risk detection, and mentor matching.

6. Scoring System Design

6.1 Rule-Based Intelligence

The scoring system is a deterministic and interpretable rule-based framework that converts raw student data into meaningful performance indicators. All scores **are normalized to a 0–100 scale** to make them easy to interpret and compare.
The purpose of this system is to provide mentors with a clear understanding of student status across multiple dimensions and support early decision-making.
Scoring Logic and Thresholds

6.2 Academic Performance Score (APS)

Purpose:
To measure academic strength and consistency.

Inputs Used

- GPA
- Attendance
- Assignment completion

Formula

$$\mathbf{APS = 0.4 \times (GPA \times 10) + 0.3 \times Attendance + 0.3 \times Assignment\ Completion}$$

Higher weight is given to GPA as it reflects core academic ability, while attendance and assignments represent discipline and consistency.

6.3 Wellness & Wellbeing Score (WWS)

Purpose:
To assess mental and physical wellbeing.

Inputs Used

- Stress level (inverted)
- Mental wellbeing
- Sleep hours

Formula

$$\begin{aligned} \mathbf{WWS} = {} & \mathbf{0.4 \times (10 - Stress) \times 10} \\ & \mathbf{+\ 0.4 \times Mental\ Wellbeing \times 10} \\ & \mathbf{+\ 0.2 \times Sleep\ Hours \times 10} \end{aligned}$$

Higher stress reduces the overall wellbeing score.

6.4 Productivity & Time Management Score (PTMS)

Purpose:
To measure focus and execution efficiency.

Inputs Used

- Productivity score
- Distractions (inverted)
- Assignment completion

Formula

$$\begin{aligned} \mathbf{PTMS} = {} & \mathbf{0.4 \times Productivity \times 10} \\ & \mathbf{+\ 0.4 \times (10 - Distractions) \times 10} \\ & \mathbf{+\ 0.2 \times Assignment\ Completion} \end{aligned}$$

This score reflects a student’s ability to manage time and complete tasks effectively.

6.5 Career Readiness Score (CRS)

Purpose:
To evaluate career preparedness.

Inputs Used
- Career clarity
- Skill readiness
- Engagement score

Formula

$$\mathbf{CRS = 0.4 \times Career\ Clarity \times 10 + 0.4 \times Skill\ Readiness \times 10 + 0.2 \times Engagement}$$

This score identifies students who may require career guidance or skill development.

6.6 Student Readiness Index (SRI)

Purpose:
SRI acts as the master indicator of overall student readiness.

Formula

$$\mathbf{SRI = 0.30 \times APS + 0.25 \times WWS + 0.20 \times PTMS + 0.25 \times CRS}$$

The SRI integrates academic, behavioral, and career dimensions into a single interpretable value used for risk classification and intervention planning.
Scoring Logic and Thresholds

6.7 Risk Classification Logic

Based on SRI, students are categorized into risk levels:

SRI Range	Category	Interpretation
≥ 75	Green	Performing well
60–74	Blue	Stable
45–59	Yellow	Needs regular guidance
< 45	Red	High risk

This classification helps mentors prioritize students and allocate support efficiently.

7. Machine Learning Design

7.1 Role of ML in the System

The primary role of machine learning in the HEPro AI+ system is to identify hidden behavioral patterns among students and group them based on overall readiness and support needs. Instead of relying only on predefined rules, machine learning enables the system to discover natural groupings based on multi-dimensional performance data.

The clustering model uses the derived performance indicators:

- APS
- WWS
- PTMS
- CRS
- SRI

These features represent academic strength, wellbeing, productivity, and career readiness. By analyzing these dimensions together, the system identifies students who show similar behavioral characteristics and may require similar mentoring strategies.

7.2 Student Segmentation Model

Model Type: K-Means Clustering

Unsupervised learning using K-Means clustering was applied to segment students into three groups. The value of K = 3 was selected based on interpretability and practical mentoring needs.

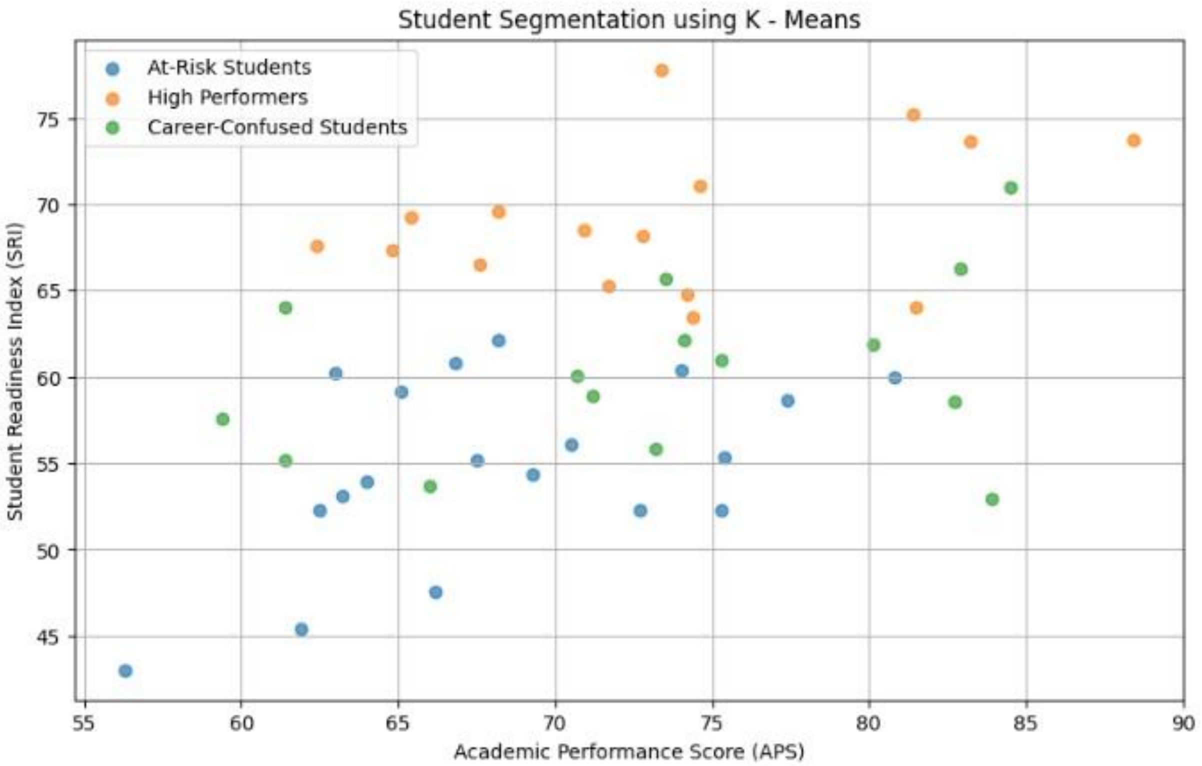
The clustering process included:

- Feature standardization using StandardScaler
- Model training with random_state = 42
- Cluster assignment for each student
- Silhouette score evaluation to check cluster quality

The final clusters were labeled as:

- Cluster 0 – At-Risk Students
- Cluster 1 – High Performers
- Cluster 2 – Career-Confused Students

Cluster Distribution



Cluster counts:

- At-Risk Students: 19
- High Performers: 16
- Career-Confused Students: 15

cluster	count
0	19
1	16
2	15

dtype: int64

This balanced distribution confirms that the dataset contains realistic diversity in student behavior.

Cluster Interpretation:

Average cluster scores:

Cluster	APS	WWS	PTMS	CRS	SRI
At-Risk	~68	~53	~39	~52	~55
High Performers	~73	~57	~71	~71	~69
Career-Confused	~73	~52	~70	~44	~60

Interpretation:

At-Risk Students

- Lower productivity and moderate wellbeing
- Need academic monitoring and wellness support

High Performers

- Strong scores across all dimensions
- Suitable for advanced growth opportunities

Career-Confused Students

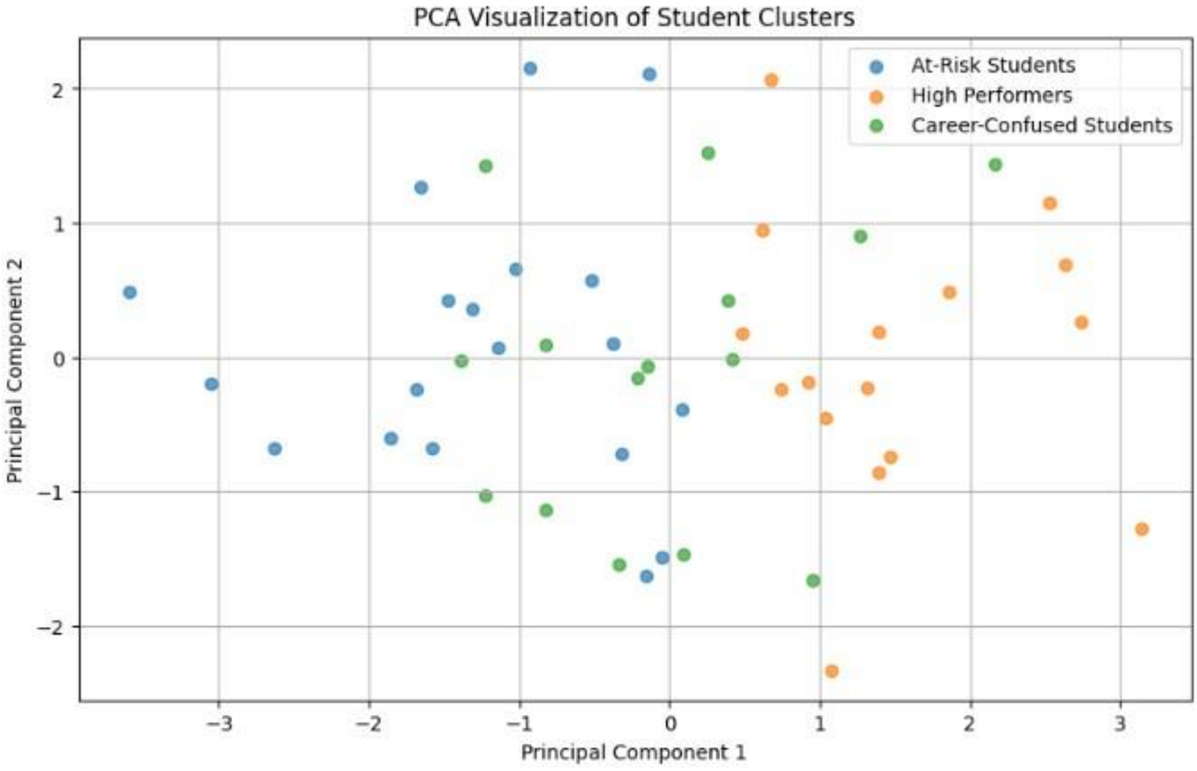
- Good academics and productivity

- Low career clarity and readiness
- Require career guidance

Category Cluster_Label	Blue	Green	Red	Yellow
At-Risk Students	4	0	1	14
Career-Confused Students	8	0	0	7
High Performers	14	2	0	0

Cross-tab analysis between SRI categories and cluster labels further validates the logical alignment between rule-based classification and machine learning segmentation.

7.3 PCA Visualization



Principal Component Analysis (PCA) was applied to reduce the feature space into two dimensions for visualization. The PCA plot shows clear separation between clusters, indicating that the selected features effectively capture student behavioral differences. This visualization confirms the quality and stability of the clustering model.

7.4 Similarity-Based Mentor Matching

After segmentation, students are assigned mentors using a similarity-based approach.
Feature Vector Design

Each student is represented using normalized values of:

- APS
- WWS
- PTMS
- CRS

Mentors are encoded into domain-based vectors representing their expertise areas such as Academic, Wellness, or Career.

Cosine Similarity

The similarity between student needs and mentor expertise is calculated using cosine similarity.

Final matching score:

Final Score = Cosine Similarity + Style Bonus + Availability Weight

Additional factors:

- Style compatibility bonus: +0.10
- Availability weighting: up to +0.15
- Capacity constraint enforcement

The mentor with the highest final score and available capacity is assigned to the student.

7.5 Model Evaluation Strategy

The machine learning component was evaluated using:

- Silhouette score for clustering quality
- PCA visualization for separation validation

- Cluster summary analysis
- Cross-tab comparison with SRI categories

The results confirmed that the clusters are meaningful, interpretable, and aligned with real mentoring requirements.

8. Intervention & Decision System

8.1 Intervention Philosophy

The intervention system is designed to convert analytical insights into practical mentoring actions. Instead of a one-size-fits-all approach, interventions are tailored based on student cluster and readiness level.

The objective is to ensure:

- Early risk detection
- Targeted mentoring
- Efficient resource allocation
- Continuous monitoring of improvement

8.2 Rule Engine Design

The decision engine uses the following inputs:

- Cluster Label
- SRI Score
- Matching Score
- Mentor Availability

Based on these inputs, the system generates:

- Mentor assignment
- Recommended intervention
- Alert status

8.3 Intervention Levels

Automated Level

For stable students:

- Regular monitoring
- Periodic mentor check-ins

Mentor-Assisted Level

For moderate-risk students:

- Structured mentoring sessions
- Performance tracking
- Career or productivity guidance

Escalation Level

For high-risk students:

- Immediate academic recovery plan
- Wellness counselling
- Frequent monitoring

8.4 Decision Flow

Cluster-based interventions:

Cluster	Intervention
At-Risk	Immediate Academic & Wellness Support
Career-Confused	Career Planning & Skill Alignment
High Performers	Advanced Growth & Leadership Coaching

Alert logic:

- SRI < 50 → High Risk Alert
- At-Risk cluster → Monitor Closely
- Others → Stable

Final outputs are stored in:
outputs/student_mentor_recommendations.csv

9. Mentor Playbook Framework

9.1 Purpose of Playbooks

Mentor playbooks provide structured guidance on how mentors should interact with different types of students. These playbooks ensure consistency, effectiveness, and measurable outcomes.

The playbooks are aligned with cluster behavior and student needs.

9.2 Playbook Structure

Each playbook includes:

- Student profile characteristics
- Key challenges
- Mentoring focus areas
- Recommended activities
- Monitoring frequency
- Expected outcomes

This structured approach helps mentors take focused and practical actions.

9.3 Core Playbooks

At-Risk Students

Focus Areas

- **Academic recovery**
- **Stress management**
- **Time management**

Recommended Actions

- **One-on-one mentoring**
- **Weekly progress reviews**
- **Structured study plans**
- **Wellness counselling**

Career-Confused Students

Focus Areas

- **Career exploration**
- **Skill-to-career alignment**

Recommended Actions

- **Career counselling sessions**
- **Skill roadmap creation**
- **Industry exposure**
- **Aptitude assessment**

High Performers

Focus Areas

- **Advanced learning**

- **Leadership development**

Recommended Actions

- Internship guidance
- Project-based learning
- Leadership roles
- Monthly mentoring check-ins

10. System Implementation

10.1 Technology Stack (Python-Based)

The HEPro AI+ system was implemented using a Python-based data science and web deployment stack. The technologies were selected to support data processing, machine learning, visualization, and interactive user access.

The major tools and libraries used include:

- Python – Core programming language
- Pandas – Data processing and manipulation
- NumPy – Numerical operations
- Scikit-learn – Machine learning (K-Means, scaling, PCA, cosine similarity)
- Plotly – Interactive visualizations
- Streamlit – Dashboard development and deployment

The project dependencies are defined in the requirements.txt file to ensure reproducibility and easy deployment.

10.2 Backend Architecture

The backend of the system follows a modular pipeline structure:

Data Processing Modules

- student dataset generation
- feature engineering and scoring
- cluster assignment

Machine Learning Modules

- K-Means segmentation
- PCA transformation
- cluster labeling

Decision Modules

- cosine similarity mentor matching
- style compatibility logic
- availability and capacity filtering

Feedback Modules

- SRI improvement simulation

- mentor rating generation
- matching issue identification

The feedback loop is implemented using a Python script that generates performance improvements based on cluster type and evaluates mentoring effectiveness.
Students with:

- low SRI improvement
- low mentor rating
- low matching score

are flagged in matching_issues.csv for review.
imp codes and csv files

10.3 Data Processing Pipeline

The complete data pipeline is as follows:

1. Generate student dataset (students.csv)
2. Apply rule-based scoring → students_with_scores.csv
3. Perform clustering → students_with_clusters.csv
4. Apply mentor matching → student_mentor_recommendations.csv
5. Generate feedback → mentor_feedback.csv
6. Identify issues → matching_issues.csv

Each stage produces an intermediate output file, allowing easy debugging and validation.

10.4 ML Pipeline

The machine learning workflow includes:

- Feature selection: APS, WWS, PTMS, CRS, SRI
- Standardization using StandardScaler
- K-Means clustering (K = 3)
- Silhouette score evaluation
- PCA transformation for visualization

For mentor matching:

- MinMax scaling of student features
- Domain-based mentor vector encoding
- Cosine similarity computation
- Style compatibility bonus
- Availability weighting
- Capacity constraint enforcement

This hybrid pipeline ensures both analytical accuracy and operational feasibility.

10.5 Dashboard / Visualization Layer

The system includes a fully interactive Streamlit dashboard designed to support real-time decision making.

Public Deployment:
https://dedicatedmentoringsystemforstudents-ai-ml-project-hhw9jxeyqlfs.streamlit.app/
The dashboard consists of four main sections:

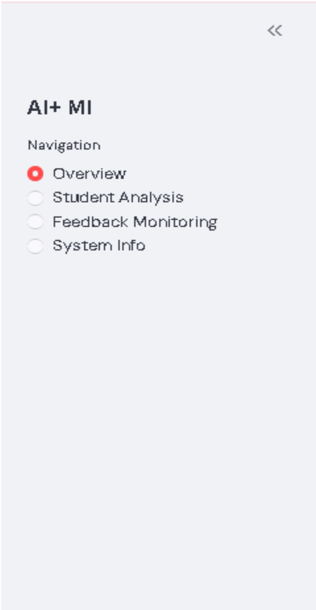
1. Overview
- Total students
- High-risk students
- Mentor count
- Matching issue count
- Cluster distribution
- SRI distribution

2. Student Analysis
- Individual student scores (APS, WWS, PTMS, CRS, SRI)
- Cluster label
- Assigned mentor
- Matching score
- Recommended intervention
- Risk alert status

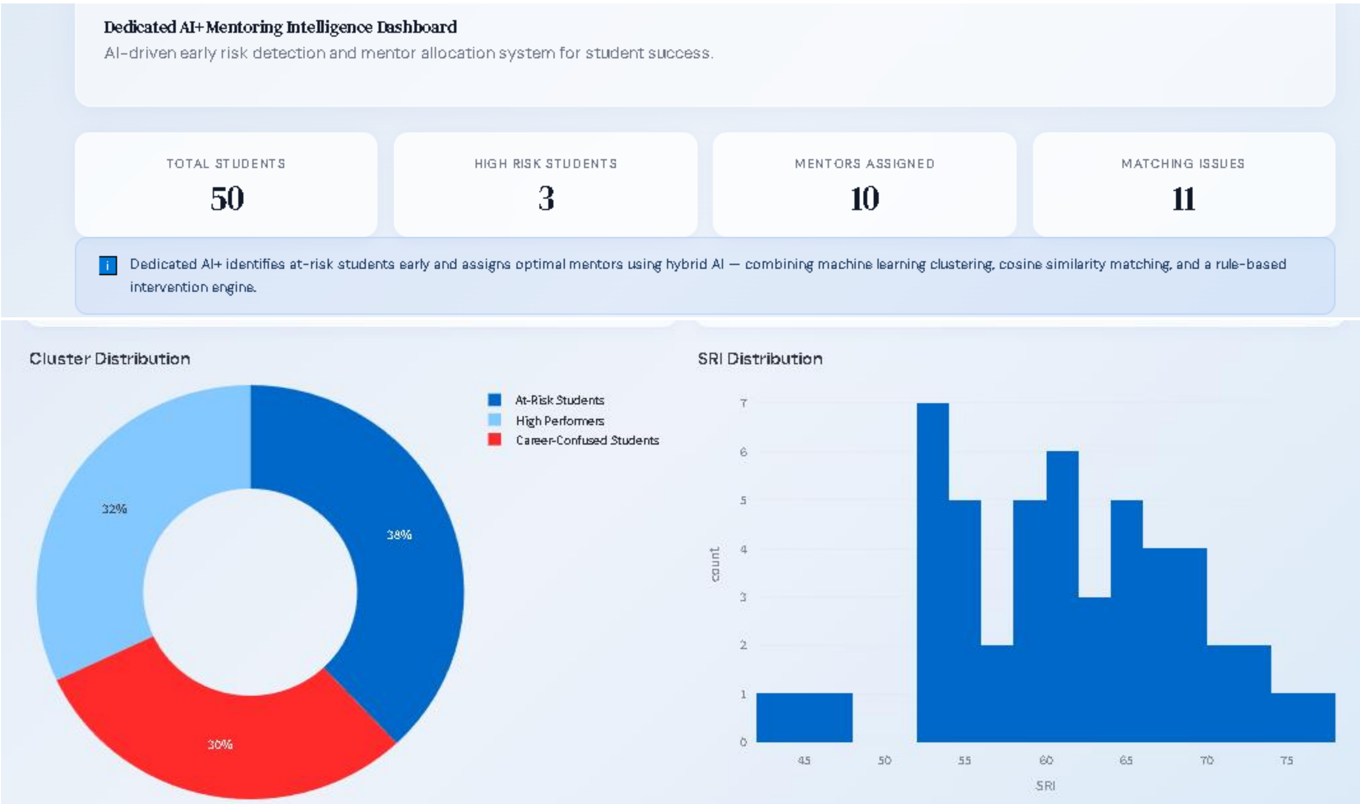
3. Feedback Monitoring
- SRI improvement distribution
- Mentor rating analysis
- Matching issues table

4. System Architecture
- Scoring engine
- Segmentation module
- Mentor matching logic
- Feedback learning system

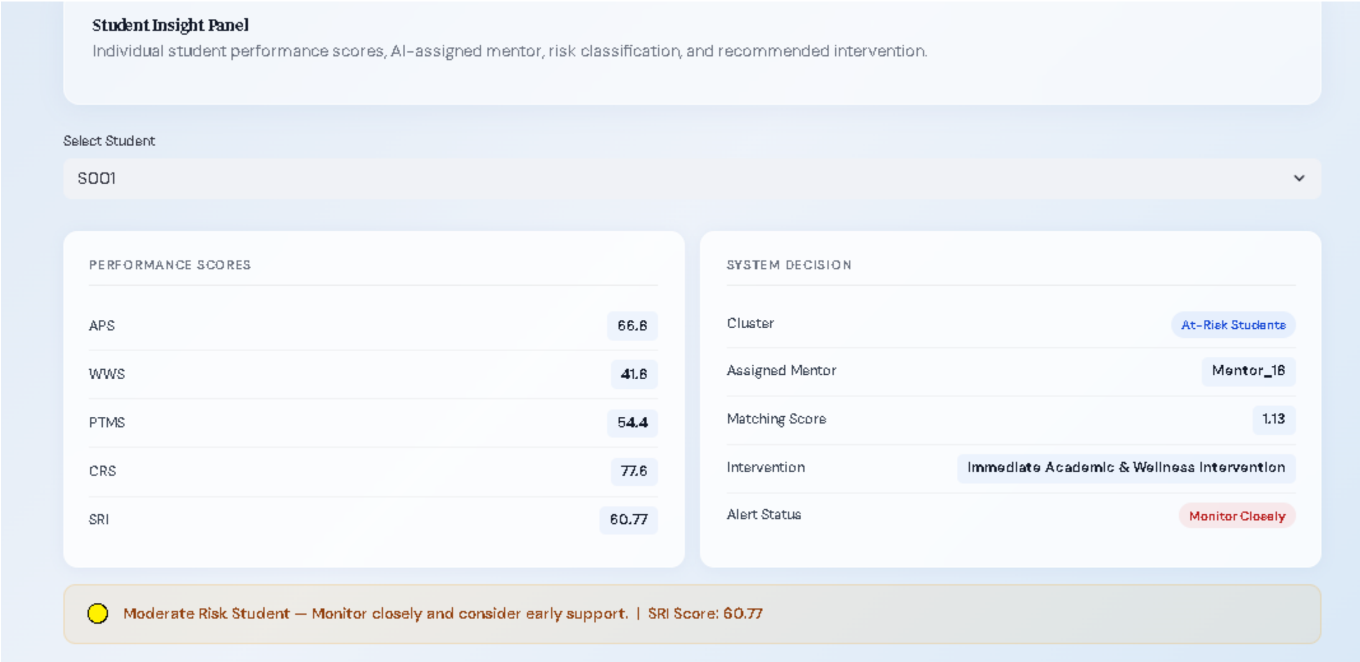
Navigation:



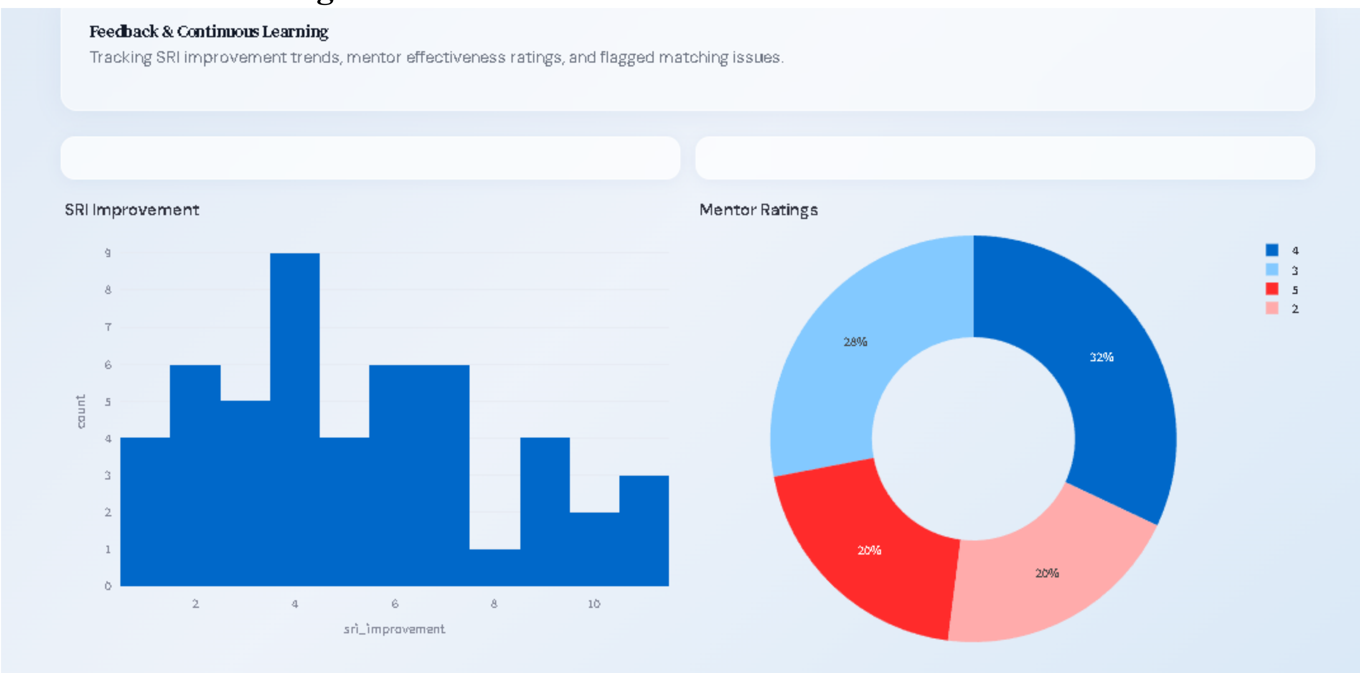
Dashboard:



Student Analysis Dashboard:



Feedback Monitoring Dashboard:



Matching Issues

	student_id	mentor_name	cluster	matching_score	sri_improvement	mentor_rating	issue
0	S002	Mentor_1	High Performers	1.015	1	2	Low effectiveness - review matching weights
1	S004	Mentor_7	High Performers	1.005	1	2	Low effectiveness - review matching weights
2	S008	Mentor_7	High Performers	1.056	1	2	Low effectiveness - review matching weights
3	S020	Mentor_17	Career-Confused Students	1.1	2	2	Low effectiveness - review matching weights
4	S026	Mentor_1	High Performers	1.058	2	2	Low effectiveness - review matching weights
5	S028	Mentor_13	High Performers	1.085	1	2	Low effectiveness - review matching weights
6	S033	Mentor_5	Career-Confused Students	1.066	2	2	Low effectiveness - review matching weights
7	S036	Mentor_1	High Performers	1.048	2	2	Low effectiveness - review matching weights
8	S040	Mentor_14	High Performers	0.981	2	2	Low effectiveness - review matching weights
9	S044	Mentor_14	High Performers	1.1	2	2	Low effectiveness - review matching weights

System Info Dashboard:

This system combines rule-based scoring, unsupervised ML clustering, cosine similarity mentor matching, and a feedback-driven continuous learning loop — designed as a production-grade AI mentoring engine.

SCORING ENGINE

- Academic Performance Score (APS)
- Wellbeing & Welfare Score (WWS)
- Peer & Tutor Match Score (PTMS)
- Course Readiness Score (CRS)
- Student Readiness Index (SRI)

SEGMENTATION

- K-Means Behavioural Segmentation
- Cluster Label Assignment
- Dynamic Re-clustering on Feedback

MENTOR MATCHING

- Cosine Similarity Mentor Matching
- Style Compatibility Weighting
- Mentor Availability Scoring
- Capacity-Based Assignment Logic

INTELLIGENCE & LEARNING

- Intervention Recommendation Engine
- Alert Status Classification
- Feedback-Based Continuous Learning
- Matching Issue Tracking & Review

The dashboard provides a user-friendly interface for mentors and administrators to monitor student performance and take timely action.

11. Results & Outputs

11.1 Scoring Results

The rule-based scoring system successfully generated normalized performance scores for all students.

Category distribution based on SRI:

- Blue (Stable): Majority of students
- Yellow (Needs guidance): Moderate group
- Green (High performing): Small group
- Red (High risk): Few critical cases

This distribution reflects realistic academic environments where most students fall into moderate performance ranges.

11.2 Clustering Results

The K-Means model segmented students into three meaningful groups:

- At-Risk Students: 19
- High Performers: 16
- Career-Confused Students: 15

Cluster	count
0	19
1	16
2	15

dtype: int64

Cluster_Label \ Category	Blue	Green	Red	Yellow
At-Risk Students	4	0	1	14
Career-Confused Students	8	0	0	7
High Performers	14	2	0	0

The cluster summaries confirmed that:

- At-Risk students have low productivity and moderate wellbeing
- High Performers show strong scores across all dimensions
- Career-Confused students have strong academics but low career readiness

This validates the effectiveness of the clustering approach.

11.3 Mentor Matching Results

The cosine similarity-based matching system successfully assigned mentors to all students based on:

- Performance needs
- Domain expertise alignment
- Mentoring style compatibility
- Availability and capacity

Matching scores ranged approximately between 0.9 and 1.2, indicating strong alignment between student needs and mentor expertise.

Cluster-based intervention mapping ensured:

- At-Risk → Academic & Wellness support
- Career-Confused → Career planning
- High Performers → Growth and leadership mentoring

11.4 Feedback Results

The feedback simulation showed measurable improvement in student readiness.

Observations:

- At-Risk students showed highest SRI improvement (3–11 points)
- Career-Confused students showed moderate improvement
- High Performers showed smaller but consistent improvement

Mentor ratings were mostly between 3 and 5, indicating effective matching in most cases.

11.5 Matching Issue Analysis

Some cases were flagged for review where:

- SRI improvement < 3
- Mentor rating ≤ 2
- Matching score was relatively low

These cases were stored in matching_issues.csv and indicate situations where matching weights or mentor assignment logic may need adjustment.

This feedback mechanism demonstrates the system's ability to support continuous learning and improvement.

12. Evaluation

12.1 System Performance

The performance of the HEPro AI+ system was evaluated based on interpretability, clustering quality, matching effectiveness, and overall system behavior.

The K-Means clustering model successfully segmented students into three meaningful groups with balanced distribution:

- At-Risk Students: 19
- High Performers: 16
- Career-Confused Students: 15

The silhouette score indicated acceptable cluster separation, and PCA visualization confirmed that the clusters were reasonably distinct in feature space. These results demonstrate that the selected features (APS, WWS, PTMS, CRS, and SRI) effectively capture student behavioral differences.

The mentor matching system produced matching scores typically between 0.9 and 1.2, indicating strong alignment between student needs and mentor expertise.

12.2 Interpretability

One of the major strengths of the system is its high level of interpretability.

- All performance scores are rule-based and transparent
- Each score is normalized to a 0–100 scale
- SRI provides a clear overall readiness indicator
- Cluster labels are behavior-based and easy to understand
- Mentor assignment includes an explainable matching score

This transparency ensures that mentors and administrators can trust the system's recommendations and understand the reasoning behind each decision.

12.3 Scalability

The system is designed with scalability in mind.

- Modular pipeline structure allows independent component updates
- CSV-based intermediate outputs enable easy integration with databases
- Streamlit dashboard supports real-time visualization

- Machine learning models can handle larger datasets without major changes

With minor modifications, the system can be scaled to support thousands of students across multiple departments or institutions.

12.4 Reliability

The reliability of the system was evaluated through:

- Validation of scoring logic against expected behavior
- Consistency checks across datasets
- Feedback loop simulation
- Matching issue detection

The feedback analysis showed that most students experienced improvement in SRI after mentoring. Only a small number of cases were flagged for review, indicating that the matching logic is generally effective but adaptable for further optimization.

13. Discussion

13.1 Key Learnings

This project provided practical understanding of how artificial intelligence can be applied to real-world educational challenges. Some key learnings include:

- Student performance must be evaluated holistically rather than only academically
- Combining rule-based logic with machine learning improves both accuracy and interpretability
- Clustering helps identify hidden behavioral groups
- Similarity-based matching improves personalization
- Feedback loops are essential for continuous improvement

The project also demonstrated the importance of data preprocessing, feature engineering, and system design in building real-world AI solutions.

13.2 Challenges Faced

During the development of the system, several challenges were encountered:

- Designing a dataset that reflects realistic student behavior
- Balancing weights for scoring formulas
- Interpreting clusters meaningfully
- Designing mentor matching logic with operational constraints
- Ensuring the system remains interpretable and not overly complex

These challenges were addressed through iterative testing, validation, and domain-based assumptions.

13.3 Design Trade-offs

Certain design decisions involved trade-offs:

- Using synthetic data instead of real institutional data for privacy reasons
- Choosing K = 3 clusters for interpretability rather than complexity
- Using rule-based scoring for transparency instead of black-box models
- Simulated feedback instead of real mentoring data

These choices were made to maintain clarity, explainability, and feasibility within the project scope.

14. Ethical & Social Considerations

14.1 Data Privacy

Student-related data is sensitive and must be handled responsibly. The current system uses synthetic data, but in real-world deployment:

- Personal identifiers should be anonymized
- Access control mechanisms should be implemented
- Data storage should follow institutional privacy policies

14.2 Bias Prevention

Bias in AI systems can lead to unfair decisions. To minimize bias:

- Features were selected based on performance indicators rather than demographic factors
- All scores are rule-based and transparent
- Clustering is based purely on behavioral data

Regular monitoring and periodic model evaluation are necessary to ensure fairness.

14.3 Human-in-the-Loop Design

The system is designed to assist mentors, not replace them.

- Final decisions remain with human mentors
- Recommendations are advisory in nature
- Mentors can override system suggestions

This approach ensures responsible use of AI in educational environments.

14.4 Responsible AI Principles

The HEPro AI+ system follows key Responsible AI principles:

- Transparency
- Explainability
- Fairness
- Accountability
- Human oversight

These principles ensure ethical and practical deployment of the system.

15. Future Scope

15.1 System Enhancements

Future improvements may include:

- Real-time data integration from Learning Management Systems
- Automated alerts via email or mobile notifications
- Role-based access for administrators and mentors

15.2 Advanced ML Integration

Advanced techniques that can be added:

- Predictive risk modeling
- Time-series analysis of student performance
- Reinforcement learning for adaptive mentor assignment
- Deep learning-based behavioral analysis

15.3 Institutional Scaling

The system can be extended to:

- Multiple departments or campuses
- University-level deployment
- Integration with student information systems
- Automated reporting for academic committees

15.4 Policy Integration

The system can support institutional decision-making by:

- Identifying high-risk student trends
- Supporting retention strategies
- Measuring mentoring effectiveness
- Enabling data-driven academic policies

16. Conclusion

The Dedicated Mentoring System for Students **(HEPro AI+)** was developed as an intelligent and scalable solution to support data-driven student mentoring. The system combines rule-based scoring, machine learning-based segmentation, cosine similarity mentor matching, and a feedback-driven improvement mechanism.

By integrating multiple dimensions such as academic performance, wellbeing, productivity, and career readiness, the system provides a holistic view of student status. **The Student Readiness Index (SRI)** acts as a key indicator for risk detection and intervention planning.

The clustering model successfully identified meaningful student groups, and the similarity-based matching ensured personalized mentor allocation. The **Streamlit** dashboard provided an interactive platform for monitoring, analysis, and decision support.

Overall, the project demonstrates how hybrid AI systems can enhance mentoring effectiveness, support early intervention, and improve student outcomes. With further enhancements and real-world data integration, the HEPro AI+ system has the potential to be deployed at institutional scale.

References

Core Machine Learning & Data Science

1. **Géron, Aurélien.**
   Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.
   O'Reilly Media, 2nd Edition, 2019.
   (Used for concepts related to clustering, preprocessing, and model evaluation.)
2. **James, Gareth, et al.**
   An Introduction to Statistical Learning.

Springer, 2013.
(Foundation for supervised/unsupervised learning and data modeling.)

3. **Tan, Pang-Ning, Steinbach, Michael, and Kumar, Vipin.**
Introduction to Data Mining.
Pearson, 2014.
(Reference for clustering techniques like K-Means and pattern analysis.)

Machine Learning Algorithms & Similarity

4. **Bishop, Christopher M.**
Pattern Recognition and Machine Learning.
Springer, 2006.
(Advanced reference for clustering, feature space, and similarity concepts.)
5. **Aggarwal, Charu C.**
Recommender Systems: The Textbook.
Springer, 2016.
(Useful for similarity-based matching and recommendation logic such as cosine similarity.)

Educational Data Mining & Learning Analytics

6. **Baker, Ryan S., and Inventado, Paul S.**
Educational Data Mining and Learning Analytics.
Springer, 2014.
(Relevant for student behavior analysis and data-driven education systems.)
7. **Siemens, George, and Long, Phil.**
"Penetrating the Fog: Analytics in Learning and Education."
EDUCAUSE Review, 2011.
(Foundation for learning analytics and student performance monitoring.)

AI Systems & Practical Design

8. **Russell, Stuart, and Norvig, Peter.**
Artificial Intelligence: A Modern Approach.
Pearson, 4th Edition, 2020.
(Standard reference for intelligent systems and decision frameworks.)

Responsible AI & Ethics

9. **O'Neil, Cathy.**
Weapons of Math Destruction.
Crown Publishing, 2016.
(Reference for bias, fairness, and ethical AI concerns.)

10. **Floridi, Luciano, et al.**
“AI4People—An Ethical Framework for a Good AI Society.”
Minds and Machines, 2018.
(Supports Responsible AI and Human-in-the-Loop design concepts.)

Python & Data Tools

11. **McKinney, Wes.**
Python for Data Analysis.
O’Reilly Media, 2017.
(Reference for Pandas-based data processing.)
12. **VanderPlas, Jake.**
Python Data Science Handbook.
O’Reilly Media, 2016.
(Used for NumPy, Pandas, Scikit-learn workflows.)

Project-Specific Technical

13. **HEPro AI Internship Materials and Internal Documentation, 2026.**
https://www.hepro.in/post/dedicated-mentoring-system-for-students-hepro-ai

14. **Project Source Code and Notebooks:**

- **student dataset generation.ipynb**
https://colab.research.google.com/drive/1QSI1j-YsZb6e5FXdvuwSAic2ejPd_OhU?usp=sharing

- **student scoring system rule based.ipynb**
https://colab.research.google.com/drive/17IsaJbxcppJsEymPeSTD57CTn_ZjVdfN?usp=sharing

- **Machine learning – student segmentation.ipynb**
https://colab.research.google.com/drive/1Z8jugb39GmxFJy0XsQnzepdtw43F2uW0?usp=sharing

- **Mentor student matching logic.ipynb**
https://colab.research.google.com/drive/1eazqsYR__nWSw1SGOR8JPaJa6yMYWcTr?usp=sharing

15. **Public Deployment:**
Dedicated Mentoring System Dashboard
https://dedicatedmentoringsystemforstudents-ai-ml-project-hhw9jxeyqlfs.streamlit.app/

Appendices

Appendix A – Dataset Samples

- **students.csv**

student_id	age	program	semester	gpa	attendance	assignments_completion	stress_level	sleep_hours	mental_wellbeing	productivity_score	distractions	career_clarity	skill_readiness	engagement_score
S001	24	MBA	5	5.9	72	72	4	6.9	1	4	4	7	9	68
S002	21	BCA	1	7.9	86	80	1	4.5	8	10	10	10	8	46
S003	22	BCA	5	6.3	81	79	5	7.1	3	10	5	5	1	24
S004	24	B.Tech	7	7.4	82	91	4	9	10	3	9	3	6	65
S005	20	MBA	5	5.3	50	84	8	4.7	7	10	8	7	5	9
S006	25	MCA	1	6.7	68	56	8	6.6	10	1	3	2	6	55
S007	22	MBA	1	6.9	51	65	7	8.4	5	8	1	9	10	29
S008	22	MCA	7	8.2	93	75	3	7.7	10	5	3	10	5	4
S009	24	B.Tech	1	9.2	75	97	1	7.5	5	4	4	10	6	32
S010	19	B.Sc	8	7.7	81	98	1	7.5	7	8	2	1	5	64
S011	20	BCA	1	7.4	55	51	3	5.8	9	7	1	6	5	17
S012	24	B.Tech	4	5.8	81	50	6	5.5	5	2	7	7	4	95
S013	20	BCA	8	5.3	53	97	7	8	1	1	8	8	3	48

Note: Complete dataset contains 50 records.

- **students_with_scores.csv**

student_id	age	program	semester	gpa	ttendanc	ents_con	tress_leve	leep_hour	tal_wellb	uctivity_s	istraction	reer_clari	ill_readine	gement_s	APS	WWS	PTMS	CRS	SRI	C
S001	24	MBA	5	5.9	72	72	4	6.9	1	4	4	7	9	68	66.8	41.8	54.4	77.6	60.77	B
S002	21	BCA	1	7.9	86	80	1	4.5	8	10	10	10	8	46	81.4	77	56	81.2	75.17	G
S003	22	BCA	5	6.3	81	79	5	7.1	3	10	5	5	1	24	73.2	46.2	75.8	28.8	55.87	Y
S004	24	B.Tech	7	7.4	82	91	4	9	10	3	9	3	6	65	81.5	82	34.2	49	64.04	B
S005	20	MBA	5	5.3	50	84	8	4.7	7	10	8	7	5	9	61.4	45.4	64.8	49.8	55.18	Y
S006	25	MCA	1	6.7	68	56	8	6.6	10	1	3	2	6	55	64	61.2	43.2	43	53.89	Y
S007	22	MBA	1	6.9	51	65	7	8.4	5	8	1	9	10	29	62.4	48.8	81	81.8	67.57	B
S008	22	MCA	7	8.2	93	75	3	7.7	10	5	3	10	5	4	83.2	83.4	63	60.8	73.61	B
S009	24	B.Tech	1	9.2	75	97	1	7.5	5	4	4	10	6	32	88.4	71	59.4	70.4	73.75	B
S010	19	B.Sc	8	7.7	81	98	1	7.5	7	8	2	1	5	64	84.5	79	83.6	36.8	71.02	B
S011	20	BCA	1	7.4	55	51	3	5.8	9	7	1	6	5	17	61.4	75.6	74.2	47.4	64.01	B
S012	24	B.Tech	4	5.8	81	50	6	5.5	5	2	7	7	4	95	62.5	47	30	63	52.25	Y
S013	20	BCA	8	5.3	53	97	7	8	1	1	8	8	3	48	66.2	32	31.4	53.6	47.54	Y
S014	20	B.Sc	8	7.2	60	61	6	8.1	10	4	7	10	3	10	65.1	72.2	40.2	54	59.12	Y
S015	25	B.Sc	7	7.2	66	54	6	8.3	10	8	5	9	4	84	64.8	72.6	62.8	68.8	67.35	B

- **students_with_clusters.csv**

Cluster	Cluster_Label	PC1	PC2
0	At-Risk Students	-0.156519144	-1.631565027
1	High Performers	2.742549435	0.26346078
2	Career-Confused Stuc	-0.826533874	0.088526665
1	High Performers	0.674357264	2.070478309
2	Career-Confused Stuc	-1.223502926	-1.025686661
0	At-Risk Students	-1.470973874	0.418404394
1	High Performers	1.077727637	-2.329961662
1	High Performers	2.528929123	1.153196802
1	High Performers	2.634090599	0.692772675
2	Career-Confused Stuc	2.162867073	1.435032329
2	Career-Confused Stuc	0.415080662	-0.016519748
0	At-Risk Students	-1.853197801	-0.597072784
0	At-Risk Students	-2.630011071	-0.676350641
0	At-Risk Students	-0.515276979	0.568447429
1	High Performers	[illegible]	[illegible]

Appendix B – Model Parameters

- **K-Means configuration**

1. Algorithm: K-Means Clustering
2. Number of clusters (K): 3
3. Initialization: k-means++
4. random_state: 42
5. Max iterations: 300
6. Features used: APS, WWS, PTMS, CRS, SRI

- **Feature weights**

APS Calculation Weights
- GPA → 40%
- Attendance → 30%
- Assignment Completion → 30%

WWS Weights
- Stress (inverse) → 40%
- Mental Wellbeing → 40%
- Sleep → 20%

PTMS Weights

- Productivity → 40%
- Distractions (inverse) → 40%
- Assignment Completion → 20%

CRS Weights

- Career Clarity → 40%
- Skill Readiness → 40%
- Engagement → 20%

SRI Weights

- APS → 30%
- WWS → 25%
- PTMS → 20%
- CRS → 25%

- **Scaling methods**

  Scaling means converting all features to the same range so ML works properly.

- GPA converted to percentage (×10)
- 1–10 scale features converted to 0–100
- StandardScaler used before K-Means clustering
- MinMaxScaler used for cosine similarity matching

Appendix C – Code Snippets

- **Scoring logic**

  # Academic Performance Score (APS)
  ### Logic

  - GPA (0–10) → normalized to 0–100 using ×10
  - Attendance & Assignments already 0–100
  - Weighted aggregation

```
df["APS"] = (0.4 * (df["gpa"] * 10) + 0.3 * df["attendance"] + 0.3 *
df["assignments_completion"])
```

Wellness & Wellbeing Score (WWS)
Logic

- Higher stress → lower wellbeing (inversion)
- Mental wellbeing and sleep contribute positively
- All values normalized to 0–100

```
df["WWS"] = (0.4 * (10 - df["stress_level"]) * 10 +0.4 * df["mental_wellbeing"] * 10
+0.2 * df["sleep_hours"] * 10)
```

Productivity & Time Management Score (PTMS)
Logic

- Productivity increases score
- Distractions reduce score (inversion)
- Assignment completion reinforces execution

```
df["PTMS"] = (0.4 * df["productivity_score"] * 10 +0.4 * (10 - df["distractions"]) * 10
+0.2 * df["assignments_completion"])
```

Career Readiness Score (CRS)
Logic

- Career clarity & skill readiness are dominant
- Engagement supports readiness

```
df["CRS"] = (0.4 * df["career_clarity"] * 10 +0.4 * df["skill_readiness"] * 10 +0.2 *
df["engagement_score"])
```

Student Readiness Index (SRI)

```
df["SRI"] = (0.30 * df["APS"] +0.25 * df["WWS"] +0.20 * df["PTMS"] +0.25 *
df["CRS"])
```

#Classify Students (Green / Blue / Yellow / Red)
###Threshold Logic

- ≥ 75 → Green
- 60–74 → Blue
- 45–59 → Yellow
- < 45 → Red

```
def classify_sri(score):
    if score >= 75:
        return "Green"
    elif score >= 60:
        return "Blue"
    elif score >= 45:
        return "Yellow"
    else:
        return "Red"

df["Category"] = df["SRI"].apply(classify_sri)


score_columns = ["APS", "WWS", "PTMS", "CRS", "SRI"]
df[score_columns] = df[score_columns].round(2)
```

- **K-Means clustering**

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

features = ['APS','WWS','PTMS','CRS','SRI']
X = df[features]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(X_scaled)
```

- **Cosine similarity matching**

Cosine similarity measures how similar student needs and mentor expertise are.

```
assigned_mentors = []
matching_scores = []

for i, student_vec in enumerate(student_scaled):

    # Compute cosine similarity
    similarities = cosine_similarity(
        [student_vec], mentor_vectors
    )[0]

    # Work on a copy (important for safe iteration)
    mentor_scores = mentors.copy()

    mentor_scores["similarity"] = similarities

    # Style compatibility bonus
    preferred_style = students.iloc[i]["Preferred_Style"]

    mentor_scores["style_bonus"] = np.where(
        mentor_scores["mentoring_style"] == preferred_style,
        0.10,
        0
    )

    # Availability weighting
    mentor_scores["availability_weight"] = (
        mentor_scores["availability_hours_per_week"] /
        mentor_scores["availability_hours_per_week"].max()
    ) * 0.15

    # Final matching score
    mentor_scores["final_score"] = (
        mentor_scores["similarity"] +
        mentor_scores["style_bonus"] +
        mentor_scores["availability_weight"]
    )
```

```
    # Capacity constraint
    eligible = mentor_scores[
        mentor_scores["current_load"] < mentor_scores["max_students"]
    ]

    if len(eligible) == 0:
        assigned_mentors.append("No Available Mentor")
        matching_scores.append(0)
        continue

    # Select best mentor
    selected = eligible.sort_values(
        by="final_score",
        ascending=False
    ).iloc[0]

    # Update load in original mentors table
    mentor_index = mentors[
        mentors["mentor_id"] == selected["mentor_id"]
    ].index

    mentors.loc[mentor_index, "current_load"] += 1

    assigned_mentors.append(selected["mentor_name"])
    matching_scores.append(round(selected["final_score"], 3))

# Store results
students["Assigned_Mentor"] = assigned_mentors
students["Matching_Score"] = matching_scores
```

explanation:
Cosine similarity compares APS, WWS, PTMS, and CRS of the student with mentor domain strengths to find the best match.

- **Feedback loop**

  - Initial SRI recorded
  - Final SRI generated after mentoring
  - Improvement calculated

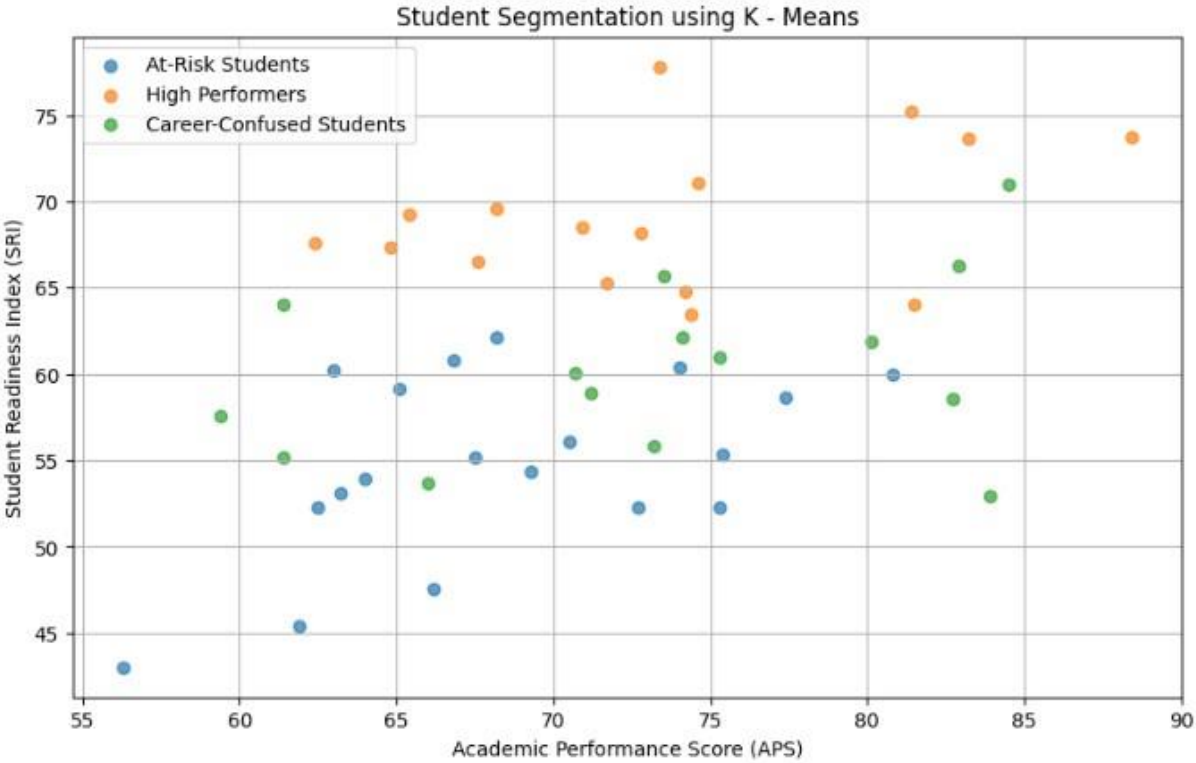
➢ Low improvement cases stored in matching_issues.csv

```
import pandas as pd
import numpy as np
import os

def generate_feedback():
    print("Generating feedback data...")

    os.makedirs("feedback", exist_ok=True)

    df = pd.read_csv("outputs/student_mentor_recommendations.csv")

    np.random.seed(42)
    records = []

    for _, row in df.iterrows():
        student_id = row["student_id"]
        mentor = row["Assigned_Mentor"]
        sri_before = row["SRI"]
        cluster = row["Cluster_Label"]
        matching_score = row["Matching_Score"]

        # Improvement logic based on cluster
        if cluster == "At-Risk Students":
            improvement = np.random.randint(3, 12)
            sessions = np.random.randint(4, 8)
        elif cluster == "Career-Confused Students":
            improvement = np.random.randint(2, 8)
            sessions = np.random.randint(3, 6)
        else:
            improvement = np.random.randint(1, 5)
            sessions = np.random.randint(2, 4)

        sri_after = min(sri_before + improvement, 100)

        # Rating based on improvement + matching quality
        if improvement >= 8 and matching_score > 0.7:
            rating = 5
        elif improvement >= 5:
            rating = 4
        elif improvement >= 3:
            rating = 3
        else:
            rating = 2

```

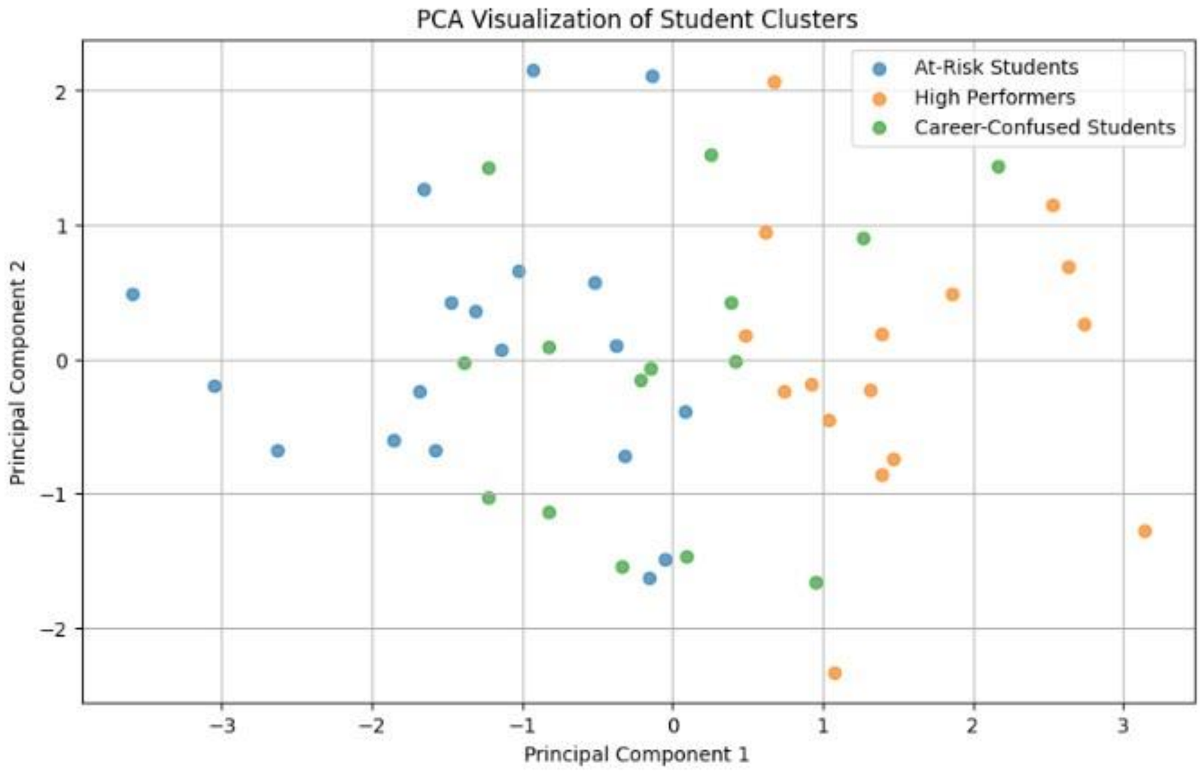
```
            records.append({
                "student_id": student_id,
                "mentor_name": mentor,
                "cluster": cluster,
                "matching_score": round(matching_score, 3),
                "initial_sri": sri_before,
                "final_sri": sri_after,
                "sri_improvement": sri_after - sri_before,
                "mentor_rating": rating,
                "sessions_completed": sessions
            })

        feedback_df = pd.DataFrame(records)
        feedback_df.to_csv("feedback/mentor_feedback.csv", index=False)

        print("Feedback file created: feedback/mentor_feedback.csv")

    def analyze_feedback():
        print("Analyzing feedback...")

        df = pd.read_csv("feedback/mentor_feedback.csv")

        issues = []

        for _, row in df.iterrows():
            if (
                row["sri_improvement"] < 3 or
                row["mentor_rating"] <= 2 or
                row["matching_score"] < 0.5
            ):
                issues.append({
                    "student_id": row["student_id"],
                    "mentor_name": row["mentor_name"],
                    "cluster": row["cluster"],
                    "matching_score": row["matching_score"],
                    "sri_improvement": row["sri_improvement"],
                    "mentor_rating": row["mentor_rating"],
                    "issue": "Low effectiveness – review matching weights"
                })

        issues_df = pd.DataFrame(issues)
        issues_df.to_csv("feedback/matching_issues.csv", index=False)

        print("Issue file created: feedback/matching_issues.csv")

    if __name__ == "__main__":
        generate_feedback()
        analyze_feedback()
```

Appendix D – Screenshots

- **K-Means scatter plot**



- **PCA visualization**



- **Cluster summary tables**

Cluster	APS	WWS	PTMS	CRS	SRI
0	68.426316	53.094737	39.694737	52.421053	54.845789
1	73.431250	71.437500	57.412500	70.987500	69.118125
2	73.353333	52.626667	70.213333	44.440000	60.315333

Appendix E – User Guide

- <u>**How to run the system**</u>

Step 1: Install Python (version 3.8 or above)

Step 2: Install required libraries

pip install -r requirements.txt

Step 3: Navigate to project folder
cd Dedicated_Mentoring_System (eg)

Step 4: Run Streamlit
streamlit run app.py

The dashboard will open in the browser.

- **How to use Streamlit dashboard**

- **Overview Page** → Shows total students, cluster distribution, risk count
- **Student Analysis** → Select student to view scores, cluster, mentor, intervention
- **Feedback Monitoring** → View SRI improvement and mentor ratings
- **System Info** → Shows architecture and model details

As mentioned above in 10.5 . Mentors can use this dashboard to identify at-risk students and monitor progress.

- **Deployment link**

Public Deployment:
https://dedicatedmentoringsystemforstudents-ai-ml-project-hhw9jxeyqlfs.streamlit.app/