Project Report: Student Success AI Platform

Team Name: Orbit **Date:** November 20, 2025

Team Members

1. **M Tanusree Reddy**
2. **Nihal DR**
3. **P Devesh Reddy**
4. **Rachana Naidu**

1. Abstract

The **Student Success AI Platform** is a web-based educational tool designed to bridge the gap between academic performance tracking and personalized mentorship. Traditional grading systems often provide feedback too late for students to correct their course. Our solution utilizes Machine Learning (Random Forest Classification) to predict potential failure based on current behavioral metrics and integrates Generative AI (Google Gemini) to provide specific, actionable advice. The platform serves as a holistic dashboard, offering predictive analytics, GPA calculation, and study material analysis in a single, modern interface.

2. Introduction

2.1 Problem Statement

In modern educational institutions, students often lack real-time visibility into how their daily habits—such as attendance and study hours—impact their final results. Furthermore, while instructors can identify at-risk students, providing personalized, actionable counseling to every student individually is resource-intensive and often unscalable.

2.2 Proposed Solution

We propose an intelligent platform that:

1. **Predicts Outcomes:** Uses historical data patterns to forecast Pass/Fail results with high accuracy.
2. **Automates Mentorship:** Leverages Large Language Models (LLMs) to act as a virtual academic counselor, offering tailored advice.
3. **Centralizes Tools:** Consolidates essential student utilities (CGPA calculator, document analysis) into one responsive application.

3. System Analysis

3.1 Functional Requirements

- **User Interface:** The system must provide an interactive dashboard for inputting metrics (Attendance, Marks, etc.).

- **Prediction Engine:** The system must process inputs and return a binary classification (Pass/Fail) with a confidence score.
- **AI Integration:** The system must generate natural language advice based on numerical inputs.
- **Data Logging:** The system must maintain a history of user predictions for tracking progress.

3.2 Non-Functional Requirements

- **Responsiveness:** The UI must adapt to mobile and desktop screens.
- **Performance:** Predictions must be generated in under 2 seconds.
- **Usability:** The interface should support multiple languages and visual themes (Dark/Light mode).

4. System Architecture

The application follows a **Client-Server Architecture**:

4.1 Technology Stack

- **Frontend:** HTML5, CSS3 (custom variables for theming), Vanilla JavaScript.
- **Visualization:** Chart.js for radar and doughnut charts; Canvas Confetti for animations.
- **Backend:** Python (Flask) serving RESTful API endpoints.
- **Machine Learning:** Scikit-learn (Random Forest Classifier), Pandas, NumPy.
- **Generative AI:** Google Gemini Pro API.
- **Persistence:** CSV-based logging system (`student_history.csv`) and browser `localStorage` for session state.

4.2 Data Flow

1. **Input:** User adjusts sliders on the Frontend.
2. **Processing:** Data is sent via JSON to the Flask backend (`/predict` endpoint).
3. **Inference:**
   - The ML model scales the data and predicts the class (0 or 1) and probability.
   - Simultaneously, a prompt is constructed and sent to Google Gemini for textual advice.
4. **Response:** The combined result (Prediction + Advice) is returned to the Frontend for display.

5. Methodology

5.1 Machine Learning Model

We employed a **Random Forest Classifier** due to its robustness against overfitting and ability to handle non-linear relationships between features.

- **Training Data:** Generated 2,000 synthetic records simulating realistic student behaviors.
- **Features:** Attendance (%), Study Hours, Internal Marks, Assignments Submitted, Activities Participation.
- **Hyperparameters:** 200 Estimators, Max Depth 12.
- **Accuracy:** Achieved ~92% accuracy on the test set.

5.2 Generative AI Integration

To humanize the data, we integrated **Google Gemini Pro**.

- **Prompt Engineering:** The system constructs a dynamic prompt: *"Act as a supportive academic counselor. A student has [Stats]... Give 2 short, specific, actionable sentences..."*
- **Safety:** The AI output is strictly scoped to academic advice to ensure relevance and safety.

6. Key Features

6.1 Interactive Dashboard

- **Real-time Sliders:** Users can simulate "what-if" scenarios (e.g., "If I study 2 more hours, will I pass?").
- **Visual Feedback:**
  - **Skill Profile (Radar Chart):** visualizes strengths and weaknesses across different metrics.
  - **Probability Gauge:** Shows the model's confidence in the prediction.

6.2 Productivity Tools

- **CGPA Calculator:** A dynamic form allowing students to input course credits and grades to calculate Semester GPA.
- **Document Analyzer:** A client-side tool that analyzes text complexity and estimates reading time for study materials.

6.3 User Experience

- **Themes:** Toggle between Light and Dark modes for comfortable viewing.
- **Localization:** Native support for English, Spanish, French, and Hindi.
- **History Tracking:** A local log of past predictions allows users to see their improvement trajectory.

7. Results and Discussion

The system successfully provides instant feedback.

- **Accuracy:** The model correctly identifies "At Risk" students in 9 out of 10 test cases.
- **Engagement:** The inclusion of the "Path to Pass" feature (algorithmic suggestions) provides a clear motivation for students to improve specific metrics.
- **Performance:** API response times average 1.2 seconds, providing a near-instantaneous user experience.

8. Future Scope

1. **Database Integration:** Migrating from CSV/LocalStorage to a cloud database (Firebase/PostgreSQL) for permanent user accounts.
2. **Advanced OCR:** Implementing server-side PDF parsing to extract syllabus topics automatically.
3. **Institutional Dashboard:** Creating a view for teachers to aggregate data from multiple students to identify class-wide trends.

9. Conclusion

The Student Success AI Platform demonstrates the powerful synergy between predictive Machine Learning and Generative AI. By moving beyond simple grade prediction to offering personalized, actionable mentorship, the tool empowers students to take control of their academic journey before it is too late.

10. References

- Scikit-learn Documentation: https://scikit-learn.org/
- Google AI Studio (Gemini API): https://ai.google.dev/
- Flask Documentation: https://flask.palletsprojects.com/