AI Curriculum Planner: Adaptive Academic Advising for 100 Simulated Students

Introduction

This project implements an AI-powered system for personalized academic advising. Its objective is to simulate an advising system for 100 students using graph-based curriculum modeling and a heuristic-based planning algorithm for course recommendations. The system aims to respect university constraints and align recommendations with student interests.

PART 1: Curriculum and Student Simulation (Graph Modeling)

Curriculum Model

The university curriculum is modeled as a

directed graph using `networkx`.

- **Nodes:** Represent courses, including `course_id`, `name`, `credits`, and `interests`.
- **Edges:** Represent prerequisite relations (e.g., `CS101 -> CS201` means CS101 is a prerequisite for CS201).

Student Simulation Logic

100 students were simulated, each with unique profiles. Each student record includes:

- `student_id`
- `gpa` (randomly generated between 2.0 and 4.0).
- `interests` (1-3 random selections from a predefined list).
- `completed_courses` and their `grades`. (Initial completed courses were simplified by selecting from those with no prerequisites).

Modeled Constraints

The system enforces:

- **Course Load Limit:** Max 3-5 courses per term.
- **Prerequisite Completion:** Courses cannot be taken without completing their prerequisites.
- **Retake Policy:** Failed courses are prioritized for retaking.

PART 2: AI-Based Personalization Strategy

Personalization Strategy and Key Design Choices

heuristic-based planning algorithm was implemented for personalization due to time constraints, instead of Reinforcement Learning. This approach uses predefined rules to recommend next-term courses that:

- **Respect constraints:** Adhere to all academic rules.
- **Align with interests:** Prioritize courses matching student interests.
- **Maximize GPA/Graduation Likelihood:** Achieved by ensuring proper progression and prioritizing retakes.

Recommendation Process

1. `get_eligible_courses`: Identifies courses a student is qualified for, ensuring all prerequisites are met and including failed courses for retake.
2. `recommend_courses`:
   - Scores eligible courses based on common interests with the student.
   - Prioritizes failed courses for retake.
   - Selects the top 3-5 courses, filling remaining slots with highest-scoring courses based on interest alignment.

Example Results for 3 Students

Student ID: S001

- **Interests:** Game Development, Mobile Development, Databases
- **Completed:** CS101(A), EN101(A), PH101(F), MA101(C)
- **Eligible:** PH101, CS201, MA201, CS202
- **Recommended:** PH101, CS201, CS202, MA201
- **Reasoning:** PH101 is prioritized for retake. CS201 and CS202 are eligible and align with programming-related interests. MA201 is eligible.

Student ID: S003

- **Interests:** Web Development
- **Completed:** EN101(B), PH101(A), CS101(B), MA101(A)
- **Eligible:** CS201, MA201, CS202
- **Recommended:** CS201, CS202, MA201
- **Reasoning:** No retakes needed. All recommended courses are foundational second-level courses, eligible based on completed prerequisites.

Student ID: S005

- **Interests:** Web Development, Game Development, Data Science
- **Completed:** MA101(A), EN101(C), CS101(C), PH101(B)
- **Eligible:** CS201, MA201, CS202
- **Recommended:** CS201, CS202, MA201
- **Reasoning:** All recommendations are eligible. These general courses set the foundation for diverse paths, including Data Science, aligning with the student's interests.

Visualizations or Performance Metrics: The heuristic algorithm successfully generates personalized course recommendations, ensuring prerequisites are met, failed courses are prioritized, and course load limits are respected. While detailed performance metrics were beyond the scope of this heuristic approach and time constraints, the system demonstrates effective constraint-aware and interest-aligned advising.