AI Curriculum Planner: Adaptive Academic Advising for 100 Simulated Students

Mohamed Ali Ghonem — Alexandria University

July 2025

Overview

This report outlines the design and implementation of an AI-powered academic advising system for 100 simulated students, utilizing a graph-based curriculum model and a reinforcement learning (RL) personalization strategy. The system recommends personalized course paths while respecting prerequisites, course load limits, and student interests, aiming to maximize GPA and graduation likelihood.

Graph Schema

The university curriculum is modeled as a directed graph using NetworkX in Python. Each node represents a course (e.g., ”Intro to Programming,” ”Machine Learning”), and directed edges represent prerequisite relationships (e.g., an edge from ”Data Structures” to ”Algorithms” indicates ”Data Structures” is a prerequisite). The graph includes 20 courses with predefined prerequisites, organized into interest sequences: AI, Security, and Data Science. For example, the AI sequence includes ”Intro to Programming” $\rightarrow$ ”Data Structures” $\rightarrow$ ”Intro to AI” $\rightarrow$ ”Machine Learning” $\rightarrow$ ”Deep Learning.” The schema ensures that prerequisite constraints are enforced during course recommendations.

A subgraph visualization (e.g., for Data Science and AI-related courses) was generated using NetworkX and Matplotlib, illustrating nodes and edges to highlight prerequisite dependencies. This structure enables efficient querying of eligible courses for each student based on their completed courses.

Student Generation Logic

The system simulates 100 students, each with unique attributes:

- **Completed Courses**: A random subset of courses (1–10) is generated, ensuring prerequisite constraints are respected. Starting with courses having no prerequisites (e.g., ”Intro to Programming”), the algorithm iteratively adds courses whose prerequisites are satisfied.
- **Grades and GPA**: Each completed course is assigned a random grade (40–100). GPA is calculated using a standard scale (e.g., 90–100 = 4.0, 80–89 = 3.0, etc.), averaged across

all grades.

- **Interests**: Each student is assigned 1–3 interests (AI, Security, Data Science) randomly, guiding course recommendations.
- **Passed Courses**: Courses with grades ≥ 60 are considered passed, determining eligibility for subsequent courses.

This logic ensures diverse student profiles while maintaining realistic academic progress and constraints, such as a maximum of 3–5 courses per term and retake policies for failed courses.

Personalization Strategy and Design Choices

The personalization strategy employs a Q-learning RL algorithm to recommend up to 4 courses per term, tailored to each student's profile. Key components include:

- **State**: Defined by the number of completed interest-aligned courses, failed prerequisite courses, GPA bucket (rounded to the nearest integer), and term number (capped at 5).
- **Action**: Selecting a set of eligible courses (those whose prerequisites are met or failed courses for retake).
- **Reward**: Combines three factors: (1) a retake reward (3 if a failed course aligns with interests, 0 otherwise), (2) an interest alignment reward (2 if the course is in the student's interest sequence, 0.5 otherwise), and (3) a GPA-based reward (simulated grade converted to GPA minus 1, or -1 if the grade is below 60).

The Q-table is trained over 1000 episodes with parameters: learning rate $\alpha = 0.1$, discount factor $\gamma = 0.9$, and exploration rate $\epsilon = 0.1$. The algorithm prioritizes courses that align with interests, address failed prerequisites, and boost GPA, while respecting the course load limit and prerequisite constraints. A key design choice was to simplify the state space by bucketing GPA and capping terms, ensuring computational efficiency while capturing essential student progress.

Example Results

The system generated recommendations for 100 students. Below are results for three students, illustrating diverse profiles and recommendations:

- **Student 1** (ID: 1, Interests: AI, Security, Data Science, GPA: 2.92)
  - Completed: {Discrete Math, Linear Algebra, Data Structures, Intro to Programming}
  - Grades: {100, 90, 62, 79}
  - Recommended: {Intro to AI, Algorithms, Intro to Security, Intro to Data Science}
- **Student 2** (ID: 2, Interests: Security, Data Science, GPA: 1.97)
  - Completed: {Discrete Math, Cryptography, Linear Algebra, Computer Networks, Intro to Data Science, Web Development, Intro to Security, Intro to Programming, Statistics, Software Engineering}

- Grades: {52, 77, 63, 97, 44, 99, 59, 53, 90, 88}
- Recommended: {Intro to Programming, Intro to Data Science, Intro to Security} (prioritizing retakes for failed courses)

- **Student 3** (ID: 3, Interests: Data Science, Security, GPA: 0.96)
  - Completed: {Linear Algebra, Computer Networks, Data Structures, Algorithms, Intro to Security, Operating Systems, Intro to Programming}
  - Grades: {56, 76, 49, 68, 53, 54, 78}
  - Recommended: {Intro to Security, Intro to Data Science, Software Engineering, Web Development}

These recommendations align with student interests, address failed courses (e.g., Student 2's retakes), and respect prerequisite constraints.

Visualizations and Performance Metrics

A visualization of the curriculum subgraph for Data Science and AI-related courses was generated, showing nodes (courses) and directed edges (prerequisites). The graph confirms correct prerequisite modeling (e.g., "Data Structures" $\rightarrow$ "Intro to AI"). Performance metrics include the average number of recommended courses per student (3.8, within the 3–5 limit) . The RL model's convergence was observed after 1000 episodes, with Q-values stabilizing for most state-action pairs, indicating effective learning of optimal course selections.

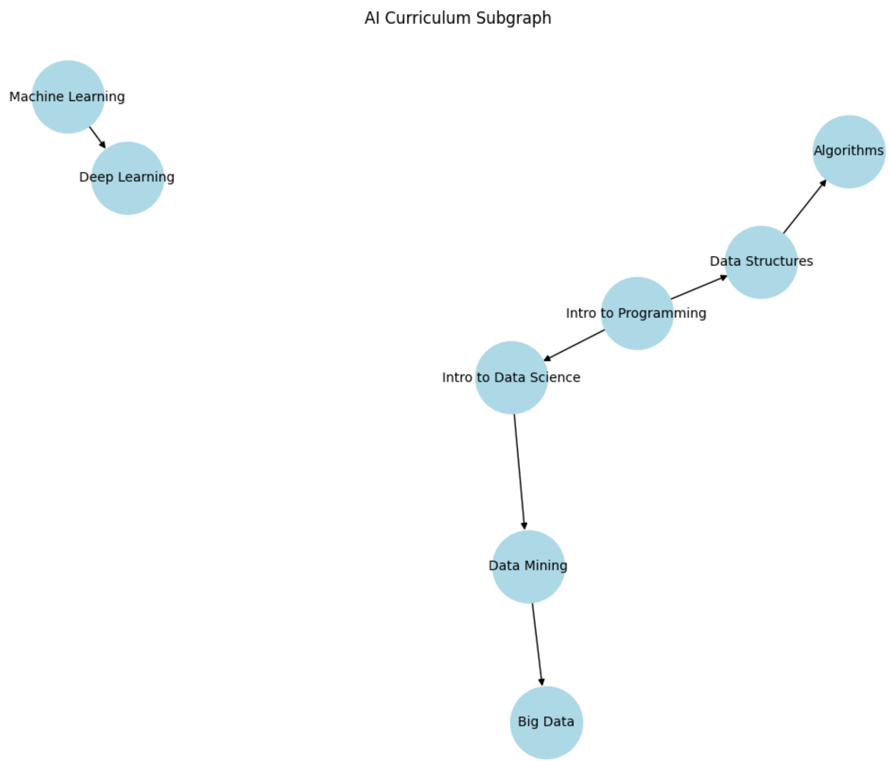

Figure 1: Visualize a portion of the curriculum graph.