Personalized Student Recommendations

Project Overview

Analyzes quiz performance data to provide personalized recommendations for students.

Uses two datasets:

Current Quiz Data: Details of a user’s latest quiz submission.

Historical Quiz Data: Performance data from the last 5 quizzes for each user.

Identifies weak areas, improvement trends, and performance gaps.

Provides actionable recommendations for students.

Setup Instructions

Prerequisites

Python 3.

Required Python libraries: pandas, numpy, requests, matplotlib, seaborn

Installation

Clone the repository:

bash

Copy

git clone https://github.com/your-username/personalized-student-recommendations.git

cd personalized-student-recommendations

Install the required libraries:

bash

Copy

pip install -r requirements.txt

Replace the placeholder API endpoints in the script with the actual URLs provided in the assignment:

current\_quiz\_url = "https://actual-api-endpoint.com/current\_quiz"

historical\_quiz\_url = "https://actual-api-endpoint.com/historical\_quiz"

Run the script:

bash

Copy

python main.py

Approach

1. Data Fetching

Fetches data from two API endpoints: one for the current quiz and another for historical quiz data

Implements error handling for connection issues or invalid responses.

2. Data Analysis

Analyzes data to identify patterns in student performance by:

Topics: Average scores and accuracy by topic

Difficulty Levels: Performance across different difficulty levels.

Response Accuracy: Accuracy of responses for each topic.

3. Insights Generation

Identifies weak areas by analyzing topics and difficulty levels with the lowest scores.

Visualizes improvement trends using a rolling average of scores over the last 5 quizzes.

4. Recommendations

Provides actionable recommendations, such as:

Topics to focus on (based on low accuracy).

Difficulty levels to practice (based on weak performance).

5. Student Persona

Defines a student persona based on their overall performance:

High Achiever: Average score > 80.

Average Achiever: Average score between 50 and 80.

Needs Improvement: Average score < 50.

Screenshots of Key Visualizations

1. Average Score by Topic

File: topic\_performance.png

Insight: Bar chart showing the average score for each topic. Topics with the lowest scores are identified as weak areas.

2. Average Score by Difficulty Level

File: difficulty\_performance.png

Insight: Bar chart showing the average score for each difficulty level. Difficulty levels with the lowest scores are highlighted for improvement.

Insights Summary

Weak Areas

Weak Topics: Topics with the lowest average scores (e.g., "Organic Chemistry", "Physics").

Weak Difficulty Level: Difficulty levels where the student struggles the most (e.g., "Hard").

Improvement Trends

The student's rolling average score shows a [positive/negative/stable] trend over the last 5 quizzes.

Recommendations

Topics to Focus On: Topics with the lowest accuracy (e.g., "Organic Chemistry").

Difficulty Level to Focus On: Difficulty levels where the student needs more practice (e.g., "Hard").

Student Persona

Based on the analysis, the student is categorized as a [High Achiever/Average Achiever/Needs Improvement].

Submission

Source Code: The Python script (main.py) and requirements.txt.

README: This document with setup instructions, project overview, and approach description.

Screenshots: Key visualizations saved as PNG files.

Video: A short video demonstrating the script.