Matiks <> AIMS-DTU Ideathon 2

Submission by: Mayank Jangid

The proposed solution aims to revolutionize math learning by simulating a 1:1 teacher experience through an **AI-Powered Personalized Insights and Feedback System**. This system meticulously analyzes various performance metrics, such as time taken per question, number of attempts, and strengths across problem categories, to deliver tailored feedback that mirrors the guidance of an experienced educator. By leveraging advanced AI and machine learning techniques, the platform goes beyond simple correctness tracking, offering adaptive learning paths, real-time performance insights, and personalized recommendations to enhance problem-solving skills and accuracy.

This document outlines the technical design, implementation strategy, and innovative features of the solution. The ultimate goal is to transform raw performance data into actionable insights, enabling users to pinpoint weaknesses, build on strengths, and achieve mastery in mathematics.

Challenges in Math Education

1. **Lack of Personalization:** Traditional classroom settings and generic online platforms fail to cater to individual learning paces and styles.
2. **Limited Feedback:** Most tools provide binary feedback (right or wrong) without addressing "why" a mistake occurred or "how" to improve.
3. **Static Content:** Existing platforms often present fixed difficulty levels, offering limited progression for advanced learners or suitable challenges for beginners.

The Need for a Solution

To address these gaps, there is a critical need for a system that:

- **Real-time adaptation:** Dynamically adjusts content difficulty and focus areas based on user performance trends.
- **Actionable Insights:** Offers detailed feedback on performance metrics, enabling targeted improvement.
- **Visual Feedback:** Presents performance analytics through intuitive visualizations, fostering user engagement and clarity.

Key Features

- **Performance Analytics:** Track granular metrics such as time per question, accuracy rates, and category-specific performance trends.
- **Adaptive Learning:** Implement a reinforcement learning algorithm to adjust problem difficulty dynamically.
- **Visual Dashboards:** Present insights via engaging graphs, charts, and heatmaps for easy interpretation.
- **Doubt Solver:** Provide an AI-powered interface for step-by-step problem explanations and on-demand topic clarifications.

Metrics to Analyze

The system analyzes various metrics to gain insights into user performance, including question details such as **content type, problem category (e.g., arithmetic, algebra), and difficulty level**. Time metrics are evaluated by assessing both **individual and average time spent** on each question, aiming to identify patterns and inefficiencies. The number of **attempts** before achieving correctness is tracked, which helps highlight user perseverance or areas of potential confusion. Additionally, **category trends** are analyzed to assess **strengths** and **weaknesses** by problem type, enabling the provision of targeted practice recommendations. The data collection workflow involves recording user interactions with problems, including time, attempts, and solution accuracy, organizing this data into structured formats suitable for analysis, and using aggregated information to generate insights and adaptive feedback through machine learning models.

Core Algorithms and Models

- **Regression Models (Linear, Ridge, Lasso):** Used to predict user improvement trends by analyzing historical performance metrics. These models are fast to train and highly interpretable, making them suitable for tracking progress over time.
- **Clustering Algorithms (K-Means, DBSCAN):** These methods segment users into groups with similar performance patterns. For example, grouping users struggling with algebra versus those excelling in geometry allows for targeted intervention.
- **Decision Trees and Random Forests:** Provide interpretable feedback by analyzing error patterns and suggesting specific areas for improvement based on decision paths.
- **Reinforcement Learning (Deep Q-Learning):** Dynamically adapts problem difficulty based on a user's success rate. This approach rewards successful completion and penalizes repeated errors, optimizing for user improvement.
- **Natural Language Processing (NLP):** Deployed in doubt-solving interfaces to parse user queries and generate contextual hints or step-by-step explanations.
- **Transformer Models (BERT/GPT):** Fine-tuned for mathematical content, these models can generate detailed explanations, solve complex user queries, and even create custom problem sets tailored to the user's skill level.
- **Graph Neural Networks (GNN):** Used to model user knowledge as a graph where nodes represent concepts, allowing the system to identify how mastery in one area influences others.

Visual Dashboards

- **Graphs:** Line charts and bar graphs to display user progress over time, categorized by problem type and difficulty.
- **Heatmaps:** Highlight areas of struggle (e.g., "Takes longer in Profit Loss Problems").
- **Progress Radars:** Visualize improvement across key skill areas.
- **Custom Animations:** Use animated visuals to explain statistical concepts or track learning milestones.

Realtime Doubt-Solving and feedback Interface

1. **NLP-Powered Search Engine:** Employ advanced NLP models like GPT or BERT to understand and interpret complex student queries in natural language, enabling the system to grasp the context and intent behind queries. This ensures precise and relevant answers while providing well-structured, clear, and step-by-step explanations, which can be enhanced with visual aids and examples.

2. **Interactive Step-by-Step Guides:** AI adapts the guides based on user performance, tailoring complexity and pace to individual needs. Dynamic hint generation powered by AI analyzes student progress to offer contextual tips, and gamified elements like badges or points for completing guides enhance engagement and motivation.

3. **Real-Time AI-Powered Chatbot:** Develop an AI assistant using technologies like OpenAI's ChatGPT or Open Source LLMs like llama, for instant doubt resolution, incorporating sentiment analysis to adjust responses based on the student's emotional state. This ensures a supportive interaction and offers 24/7 availability for continuous support and learning opportunities.

4. **Advanced Data Analytics and Feedback System:** Use AI to analyze metrics such as problem-solving time, accuracy, and category-specific performance. Implement machine learning models to predict future trends and recommend personalized strategies. Interactive dashboards provide real-time insights through visually engaging graphs and charts.

5. **Adaptive Learning and Feedback:** Continuous monitoring tracks user responses in real time, including accuracy, time taken, and retries. The system leverages reinforcement learning to dynamically adjust question difficulty and suggest targeted categories for practice. Personalized advice is offered based on performance trends, such as, "Focus on geometry for 20 minutes daily to improve speed" or "Switch to easier problems in algebra to build confidence." This real-time adjustment mimics a personal teacher's guidance, refining recommendations and insights as the student progresses.

6. **AI-Driven Content Creation:** Leverage AI to create new, personalized questions and exercises tailored to the user's needs, ensuring content diversification across various types of problems and scenarios to cover a broad range of topics and skill levels.

Technical Implementation

1. **Feedback Engine:** The feedback engine is designed to combine rule-based heuristics with machine learning models to provide contextual and personalized suggestions. The system relies on:

- **Rule-Based Heuristics:** These are predefined rules that trigger specific feedback based on the user's performance metrics, such as accuracy, speed, and the number of retries. For example, if a student repeatedly struggles with geometry questions, a rule might suggest additional practice in that area.
- **Machine Learning Predictions:** Advanced models, particularly those trained on historical data, predict future performance and areas needing improvement. These predictions help in crafting feedback that is both timely and relevant.

2. **Libraries and Frameworks**

- **TensorFlow:** This is used to implement and train reinforcement learning (RL) models that dynamically adjust the difficulty of questions and suggest categories for targeted practice. The RL models learn optimal strategies for enhancing learning outcomes based on real-time student interactions.
- **LLMs (e.g., OpenAI's GPT, or open-source models like LLaMA):**
Leveraging state-of-the-art LLMs for all NLP tasks, providing robust understanding and generation capabilities. These models are fine-tuned for mathematical content and are integrated for tasks such as parsing user queries, generating personalized feedback, and creating adaptive problem sets.

Key advantages of using LLMs:

- **Contextual Understanding:** LLMs provide superior comprehension of user intents, allowing them to interpret complex queries and provide detailed, human-like explanations.
- **Dynamic Query Handling:** Fine-tuning enables these models to adapt to math-specific terminology, generating explanations and personalized questions that align with the user's proficiency.
- **Step-by-Step Feedback:** The models can generate detailed step-by-step guides tailored to a student's learning pace and preferred explanation style.
- **Query Parsing and Intent Recognition:** LLMs will process natural language queries for the doubt-solving feature, identifying relevant concepts and generating tailored responses.
- **Adaptive Content Generation:** Use LLMs to create new problem sets and offer stepwise solutions dynamically based on user performance metrics.
- **Real-time Recommendations:** Generate instant, personalized tips for improvement, leveraging LLM capabilities for analyzing user progress and providing actionable advice.

3. **Knowledge Tracing Algorithms**

- **Bayesian Knowledge Tracing (BKT):** This algorithm predicts a student's mastery over specific concepts by updating probabilities as the student answers questions. BKT is essential for providing hyper-personalized feedback, as it continuously adapts its understanding of the student's knowledge state.
  - **Model Structure:** BKT consists of a series of hidden states representing whether a student has mastered a concept. It updates these states based on the correctness of each answer.
  - **Parameter Estimation:** Parameters such as the probability of guessing, slipping, and the initial mastery level are estimated from the data, which helps in accurately predicting the student's current state of knowledge.

4. **Doubt-Solving Engine:** The Doubt-Solving Engine provides instant, adaptive assistance through AI-powered tools, offering step-by-step solutions, interactive hints, and personalized feedback.

- **Query Understanding:**
  - NLP models (e.g., GPT, BERT) process user queries, extracting context and intent.
  - Detect errors or ambiguities in user input for better clarification.
- **Dynamic Problem Resolution:**
  - Step-by-step solutions tailored to user proficiency.
  - Contextual hints and alternative solution paths to foster learning.
  - Follow-up questions to reinforce understanding.
- **Knowledge Base Integration:**
  - Link queries to a structured knowledge base of mathematical concepts.
  - Provide relevant content and examples for accurate responses.
- **Personalized Feedback:**
  - Highlight weaknesses and suggest targeted practice.
  - Use sentiment analysis to ensure supportive, motivating interactions.
- **Query Handling:**
  - Process user input with NLP models for intent recognition and error localization.
  - Map queries to relevant concepts and solution templates.
- **Solution Generation:**
  - Combine rule-based logic with AI models to produce detailed solutions.

- Use GNNs to trace interrelated concepts and reinforcement learning to refine hints.
  - **Example Workflow**
    - Query: "How do I solve 3x^2 -7x -6 = 0"
    - Response:
      - Step-by-step solution using the quadratic formula.
      - Provide interactive hints and suggest related practice problems.
    - Feedback: Analyze user input to refine solutions and offer personalized advice.

Workflow Integration

- The REST APIs will handle essential functions such as user authentication, synchronization of performance data, and retrieval of insights and feedback. Secure authentication mechanisms like JWT or OAuth 2.0 will manage user sessions, while endpoints such as /sync-data and /get-feedback will facilitate the flow of user metrics to the backend and deliver personalized recommendations in real-time.

- For backend orchestration, Celery will manage asynchronous tasks like updating user statistics, training models, and generating feedback reports. This ensures that computationally intensive operations run in the background without disrupting the main application flow, maintaining a responsive user experience. Celery will work alongside message brokers like Redis and databases such as PostgreSQL for efficient task handling and result retrieval.

The implementation of these ideas as a proof of concept are available on my github repo: https://github.com/mayank-jangid-moon/matiks-ideathon-2