

Hackathon Problem Proposal: AI in Education - Text Classification

Theme:

"Leveraging AI to Enhance Education through Text Classification"

Objective:

Your task is to build a text-based classification model that addresses a real-world problem in education. You will:

- Find a relevant dataset related to education.
- Train a classification model (e.g., sentiment analysis, topic categorization, spam detection, essay grading, etc.).
- Export the model for inference.
- Build a Node.js backend to serve the model via an API.
- Implement JWT-based authentication for user management. (Bonus)

Problem Statement:

Education generates vast amounts of text data—student essays, discussion forums, feedback forms, and more. AI can help automate and improve processes like grading, moderation, and content organization.

Your challenge is to identify a specific problem in education that can be solved with text classification, build a model, and deploy it as a secure API.

Example Use Cases:

Automated Essay Scoring (Classify essays into grade levels)

Student Feedback Sentiment Analysis (Positive/Negative/Neutral)

Question Topic Categorization (Math, Science, History, etc.)

Plagiarism Detection (Classify text as original or plagiarized)

Discussion Forum Moderation (Detect toxic/inappropriate comments)

Requirements:

1. Dataset & Model Development

- Find a labeled dataset related to education.
- Pre-process the data (cleaning, tokenization, etc.).
- Train a classification model
- Evaluate model performance (accuracy, precision, recall, F1-score).
- Export the model

2. Backend Development (Node.js)

Create a REST API that loads the model and performs predictions.

Endpoints:

- /predict (Accepts text input, returns classification result)
- Bonus: Add model explainability (e.g., SHAP/LIME for feature importance).

3. Authentication (JWT) (Bonus)

- Implement user registration/login with JWT.
- Secure the /predict endpoint so only authenticated users can access it.
- Store user credentials securely (use password hashing like bcrypt).

4. Frontend (Bonus)

Build a simple UI (React/HTML) to interact with the API.

Evaluation Criteria:

- Creativity – How well does the solution address an educational problem?
- Model Performance – Accuracy, efficiency, and robustness.
- Backend Implementation – Clean, scalable, and secure API.
- Authentication – Proper JWT implementation with user management.
- Presentation – Clear explanation of approach, results, teamwork.
- Prize for the top team

Submission Guidelines:

- Code (GitHub repo with clear README)
- Dataset (Include source or generation method)
- Demo (Live API endpoint or video walkthrough)
- Slides