ATG Technical Assignment: Self-Healing Classification DAG with Fine-Tuned Model

Machine Learning Intern Deliverables

Task Overview

Build a LangGraph-based classification pipeline that not only performs predictions but also incorporates a self-healing mechanism. The goal is to fine-tune a transformer model and design a fallback strategy in cases of low prediction confidence, ensuring robustness and reliability in human-in-the-loop workflows.

Deadline: 72 hours from task assignment

Objective

Fine-tune a transformer-based text classification model, then integrate it into a LangGraph DAG that uses prediction confidence to decide whether to accept, reject, or request clarification for its outputs. The design should prioritize correctness over blind automation.

Requirements

- Choose any open-access text classification dataset (e.g., sentiment analysis, topic classification, emotion labeling).
- Fine-tune a transformer model (e.g., `DistilBERT`, `TinyBERT`, etc.) using LoRA or full fine-tuning.
- Build a LangGraph workflow composed of the following nodes:
  - **InferenceNode:** Runs classification using the trained model.
  - **ConfidenceCheckNode:** Evaluates the confidence score of the prediction. If below a defined threshold, triggers fallback.
  - **FallbackNode:** Avoids incorrect classification by either:
    - Asking the user for clarification or additional input, or
    - Escalating to an alternative strategy (e.g., backup model).

- Maintain a clean CLI interface to handle:
  - User inputs
  - Clarification questions
  - Final outputs with confidence scores
- Implement structured logging for:
  - Initial predictions and associated confidence
  - Fallback activations and corresponding user interactions
  - Final classification decisions

Expected Output

- End-to-end CLI execution with fallback in action:

```
Input: The movie was painfully slow and boring.

[InferenceNode] Predicted label: Positive | Confidence: 54%
[ConfidenceCheckNode] Confidence too low. Triggering fallback...
[FallbackNode] Could you clarify your intent? Was this a negative review?

User: Yes, it was definitely negative.

Final Label: Negative (Corrected via user clarification)
```

Deliverables

1. **Fine-tuned Model**: Trained on selected dataset (submit model or upload via Hugging Face hub or Drive link).
2. **Source Code**: Python scripts with clearly defined nodes and CLI loop.
3. **README.md**:
   - Instructions for running fine-tuning
   - How to launch and interact with the LangGraph DAG
   - CLI flow explanations
4. **Log File**: Contains timestamps, predictions, fallback invocations, and final decisions.
5. **Demo Video** (2–4 minutes):
   - Show CLI execution
   - Explain DAG design and fallback strategy
   - Walkthrough fine-tuned model logic

Bonus (Optional)

- Integrate a **backup model** (e.g., zero-shot classifier or ensemble) as a fallback.
- Track and display:
  - **Confidence curves** over multiple inputs
  - **Fallback frequency statistics** as a log chart or CLI histogram

Submission Format

- GitHub repository or zipped folder containing:
  - All source code
  - README.md
  - Log files
  - Trained model or download instructions
  - Demo video (or link)

Evaluation Criteria

- Model performance and appropriate use of fine-tuning
- Correct implementation of fallback logic
- Thoughtfulness in interaction design and recovery strategy
- Documentation, logging quality, and CLI usability
- Confidence in demo explanation