AI Lab Report Agent: Data Science Report

- **Name:** Yash Pathak
- **University:** Indian Institute of Technology Kanpur (IIT Kanpur)
- **Department:** Civil Engineering

1. Fine-Tuning Setup

This section covers the data, method, and results for the fine-tuned model (Agent 3: The "Writer").

A. Fine-Tuning Target & Justification

The **mistralai/Mistral-7B-Instruct-v0.2** model was chosen as the fine-tuning target. This decision was central to the project's success for three reasons:

1. **Adapted Style:** A base, general-purpose model does not know the specific academic structure required for a CE331 lab report (e.g., ## Objective, ## Methodology, ## Results and Discussion, etc.). Fine-tuning on a small set of high-quality examples was the most effective way to teach the model this rigid **stylistic structure**.
2. **Task Specialization:** The most complex task for this agent is not just writing, but **data transformation**. The agent must take raw, messy experiment data (pasted in by the user) and re-format it into clean, professional Markdown tables. Our fine-tuning dataset specifically trained the model on this "data-in, table-out" skill.
3. **Improved Reliability:** While a base model *might* be prompted to follow a structure, it's not reliable. It may forget sections or merge them. Our fine-tuned model (named checkpoint-6) now **reliably** produces this structure every single time, making it a dependable component for an automated agent.

B. Data Preparation

- **Source:** 5 high-quality, human-written example lab reports from the CE331 course.
- **Format:** A finetune_data.jsonl file was created, with each line containing a single JSON object.
- **Structure:** Each example followed the Mistral "instruction" format: {"text": "<s>[INST] {prompt} [/INST] {report} </s>"}
    - {prompt}: A "Human-in-the-Loop" prompt containing the [TASK] (e.g., "Write a full lab report...") and the [CONTEXT] (the raw data tables and key values from the original report).
    - {report}: The complete, ideal lab report, manually formatted as perfect Markdown.

C. Method

- **Technique:** Parameter-Efficient Fine-Tuning (**PEFT**) using the **LoRA** (Low-Rank Adaptation) method.
- **Environment:** Google Colab with a T4 GPU.
- **Base Model:** mistralai/Mistral-7B-Instruct-v0.2
- **Configuration:**
  - The model was loaded in 4-bit precision using bitsandbytes to fit on the T4 GPU.
  - A LoraConfig was applied with r=16 (rank) and lora_alpha=32, targeting all linear modules (q_proj, v_proj, k_proj, etc.).

D. Results (Training Loss)

The model was trained for 3 epochs. The training loss is the primary quantitative metric showing the model's learning progress. "Loss" represents how "surprised" the model is by the training data; a lower loss means the model's predictions are getting closer to the ideal "style" from our .jsonl file.

The training log shows a clear and successful learning curve:

Epoch	Step	Training Loss
1	1	1.967800
1	2	1.800300
1	3	1.692300
...	...	...
2	4	0.692100
2	5	0.534000
...	...	...
3	6	0.401100
3	7	0.380200
3	8	0.370300

Conclusion: The loss dropped from a high of **1.96** to a low of **0.37**. This is a **quantitative success**, proving that the model successfully learned the specialized structure and style of the target lab reports.

2. Evaluation Methodology and Outcomes

A. Evaluation Methodology

The agent was evaluated qualitatively using a "Human-in-the-Loop" testing method. The goal was to see if the agent could produce a complete, accurate, and correctly formatted report for a new task (Lab 4: Levelling) that was *not* included in its fine-tuning data.

Test 1: (Failure) Generic RAG Query

- **Prompt:** A simple, generic prompt: "Write a report for Lab 4 Levelling"
- **Result:** This test failed due to a **RAG retrieval error**. The RAG agent (Agent 1) searched for "Levelling" and "Collimation" and retrieved irrelevant context about *astronomy telescopes*.
- **Outcome:** The fine-tuned model (Agent 3) correctly built the report structure, but then **hallucinated** content about telescopes, proving that a "smart" writer agent is useless if it's fed "dumb" context.

Test 2: (Success) Upgraded "v3" Agent

- **Prompt:** The final agent script (final_agent_v3.py) was used. This script prompts the user for three specific inputs:
  1. **Lab Title:** Lab 4 Levelling
  2. **Key Concepts:** Rise and Fall Method, Height of Collimation, misclosure, reduced level
  3. **Experiment Data:** The two raw data tables (HI and Rise/Fall) were pasted in.
- **Result:** This test was a **complete success**.
  - The Key Concepts input allowed Agent 1 to retrieve *correct, relevant* definitions from the course books.
  - The Experiment Data was correctly passed to Agent 3.
- **Outcome:** The agent produced a coherent, accurate report that combined the book definitions (from RAG) with the user's data, all formatted in the correct style.

B. Outcomes (Generated vs. Original)

This evaluation compares the final AI-generated PDF (Lab_Report_Lab_4_Levelling...pdf) with a human-written original (lab 3 report final.pdf).

1. Generated Report (The AI's Output)

- **Pros:**
  - **Structure:** Perfectly matched the fine-tuned style (Objective, Equipment, Methodology, Results, Conclusion).
  - **Data Transformation:** This was the biggest success. The agent found the raw, pasted data for *both* tables and perfectly re-formatted them into clean Markdown tables.
  - **Automation:** The agent successfully automated ~95% of the content-generation

work.

- **Cons:**
  - **Hallucination:** The model *still* invented nonsensical formulas (Staff Size (m) = sqrt(...)). This proved that our "simplified" prompt (which removed RAG context to fix the *first* hallucination) was a better approach. The model works best when it is *not* asked to reason about book context and user data *at the same time*.
  - **Layout:** Agent 4 produced a very simple, single-column PDF.

2. Original Report (The Human's Output)

- **Pros:**
  - **Layout:** A professional, two-column layout created in a word processor.
  - **Prose:** The text is complex, human-written, and 100% accurate.
- **Cons:**
  - **Time:** This report took hours of manual work to write, format, and add data to.

Final Evaluation Conclusion:
The AI Agent prototype was a definitive success. It proved that a fine-tuned model (Agent 3) is the correct choice for automating tasks with a specific style and structure. It also demonstrated that the fine-tuned model's most valuable skill was its ability to re-format new user data.
The final, "v3" agent design (where the human provides all context via prompts) is the most reliable, as it avoids the "hallucination" conflicts that arise from a simple RAG system. The agent successfully fulfills its role as an **AI collaborator**, automating 95% of the tedious work and leaving the final 5% of layout and polishing to the human user.