Deliverable: Justification for Fine-Tuning Target

Project: Academic Study-Guide Agent
Model: NoteExtractor (Fine-Tuned google/gemma-2b-it)
Author: Rishabh Kumar

1. Primary Goal of Fine-Tuning

The core component of this AI agent is a fine-tuned model, the NoteExtractor. The decision to fine-tune a base model (Gemma 2b-it) rather than using a general-purpose, pre-trained model was a mandatory prerequisite for the agent's success.

A general-purpose model is unsuitable for this task for two primary reasons:

1. **It is unreliable:** It does not guarantee a 100% valid, machine-readable JSON output, which would break the agent's tool pipeline.
2. **It has the wrong style:** It cannot know my personal, specific mental model for how lecture notes should be structured.

My choice of fine-tuning target directly addresses these problems by focusing on **Adapted Style** and **Improved Reliability**, which combine to achieve **Task Specialization**.

2. Explanation of Fine-Tuning Targets

A. Adapted Style (The Primary Justification)

A general-purpose LLM can summarize text, but it does so based on a generic understanding of language. It has no knowledge of my personal, **adapted style** of note-taking.

My manual task is not just "summarizing," but "structuring." I have a specific mental model for what I consider a main_topic, a detail, or a child topic. The **dataset.jsonl** file, which contains 27 examples I created, *is the definition of my style*.

By fine-tuning the model on these examples, I taught it to move beyond generic summarization and adopt my specific, desired output structure. The model is no longer just guessing; it is actively mimicking the hierarchical "style" I provided, a task impossible for a base model.

B. Improved Reliability (The Technical Justification)

The agent's "Reason, Plan, Execute" pipeline is a chain of tools. The markdown_generator.py and mindmap_generator.py tools are not intelligent; they are simple scripts that **require** a perfectly-formatted, valid JSON object as input.

A general-purpose model provides **zero reliability** in its output format.

- It might add chatty text (e.g., "Here is the JSON you requested...").
- It might make a formatting error (e.g., a missing comma or brace).
- It might hallucinate random text after the JSON (as we saw in our initial test with the word Darío).

Any of these errors would break the entire agent pipeline.

By fine-tuning the model *exclusively* on text-to-JSON examples, I trained it to be **reliable**. The model learned that the *only* valid response is a well-formed JSON object. This improved reliability is not a "nice-to-have"; it is the core requirement that makes the agent's automated tool-use possible.

C. Task Specialization (The Final Result)

These two goals, **Adapted Style** and **Improved Reliability**, combine to achieve the final goal: **Task Specialization**.

I did not need a general-purpose chatbot. I needed a specialist. The fine-tuning process transformed the generalist gemma-2b-it model into a highly specialized tool, NoteExtractor. This new model has only one job: to receive unstructured lecture text and output a valid JSON object that matches my personal note-taking hierarchy. This specialization is what makes the final agent prototype functional, reliable, and useful.