Executive Summary: Math Solution Classifier

GitHub Repository: `https://github.com/arvindsuresh-math/Erdos-DL-June25-Math`

1. Project Overview

Grading math homework is a critical but uniquely demanding task for educators; it is tedious and repetitive, yet requires deep concentration to provide fair and accurate feedback. The **Math Solution Classifier** is a proof-of-concept AI assistant designed to lighten this burden.

This project proves that a high-quality, reliable grading assistant does not require expensive, cloud-based frontier models or high-end hardware. Instead, by using a pipeline of small, specialized, and fine-tuned open-source models, we can deliver a powerful tool that runs entirely on-device, ensuring privacy, stability, and cost-effectiveness.

2. Stakeholders & Product Value

Our primary stakeholder is the **K-8 math educator**. These teachers are often overworked and need tools that are not only effective but also practical and trustworthy. Our solution is tailored to their core needs:

- **Time Savings:** By automating the initial check of student work, our tool frees up valuable time for lesson planning and direct student interaction.
- **Accuracy:** The system provides a reliable "first pass," flagging solutions that need closer attention.
- **Student Data Privacy:** By running locally, the system guarantees that sensitive student data is never transmitted to a third-party API, a critical requirement for any educational tool.
- **Accessibility:** The small model sizes ensure the tool can run on standard consumer laptops, making it accessible to any teacher without the need for expensive hardware or cloud subscriptions.

3. Modeling Approach

Our core strategy was to **divide and conquer**. Instead of tasking a single, large model with the complex job of grading, we built a hybrid pipeline of two small, specialized models, each fine-tuned for a specific sub-task.

- **Hybrid System:**
    1. **Conceptual Error Model:** A fine-tuned `microsoft/Phi-4-mini-instruct` (4B parameters) acts as a binary classifier to assess the overall logic and reasoning of the solution.
    2. **Computational Error Model:** A fine-tuned `unsloth/gemma-3-1b-it` (1B parameters) performs a hyper-specific text extraction task. Its output is then passed to a **deterministic programmatic check** that infallibly verifies the arithmetic. This hybrid approach is the key to our system's high accuracy in detecting calculation mistakes.
- **Novel Data Pipeline:** This was enabled by a novel data generation pipeline where we used a powerful LLM to create structured "Formalization Templates." We then used **Abstract Syntax Tree (AST) manipulation** on these templates to programmatically inject thousands of high-quality, realistic errors, creating a robust training dataset from the ground up.

4. Key Results

For a grading assistant, the most critical metric is its ability to reliably identify flawed work and avoid false negatives. A teacher needs to trust that the tool will not incorrectly label a flawed solution as "correct." In this regard, our fine-tuned pipeline demonstrates a decisive advantage in **recall**—the measure of how well a model can find all relevant instances in a dataset.

While the baseline slightly outperforms our model on the Final Test Set in overall accuracy, our pipeline has a **superior and more consistent recall score for detecting all incorrect solutions** across both test sets.

Recall for Incorrect Solutions	Baseline	Fine-Tuned Model
SFT Test Set	81.44%	95.70%
Final Test Set	91.39%	94.00%

This is the most important metric for a tool designed to help human graders. It shows that our pipeline is exceptionally reliable at its core task: flagging solutions that require a teacher's attention.

The standout feature driving this high performance is our model's **mastery of detecting computational errors**, a direct result of our hybrid architecture. This is where the baseline's inconsistency becomes most apparent.

Recall for Computational Errors	Baseline	Fine-Tuned Model
SFT Test Set	30.5%	92.5%
Final Test Set	90.7%	93.3%

As the tables show, our fine-tuned system is a specialist, consistently identifying over 92% of calculation mistakes. The baseline, in contrast, is unreliable; its ability to detect the same errors collapsed from 90.7% to a mere 30.5% on the more diverse SFT test set.

Implication for Educators: A teacher using our tool can be confident that it will successfully flag nearly every paper with a computational or conceptual error, allowing them to focus their limited time on providing feedback where it is most needed. The baseline, while sometimes effective, is too unpredictable to be a trustworthy assistant.

5. Main Conclusion

The strong performance of our small-model pipeline validates our thesis:

*It is possible to create a robust and performant grading assistant to lighten a middle school teacher's grading load **without** relying on expensive, cloud-based frontier models, or expensive hardware.*

6. Future Work

- Incorporate OCR to allow input of hand-written solutions.
- Enlarge model size and diversity of fine-tuning data to improve generalization.
- Move onto high school problems (and some day, college level problems).