1. Detecting Bias in Our Data

In the context of our project, bias refers to a language model being more vulnerable to adversarial prompts from certain categories (e.g., harassment, policy, toxicity) than others.
At this stage, our pipeline only performs data preprocessing, and the dataset contains:

- prompt_text
- category (policy, harassment, toxicity, etc.)
- text_length and length_bucket (short/medium/long)

Since we do not collect demographic attributes such as gender, age, race or location, we cannot perform conventional demographic bias detection. Instead, our focus is on identifying category-level bias, whether a model is disproportionately successful or fails more often for specific types of adversarial prompts. This ensures fairness in safety performance across different risk categories.

2. Data Slicing for Bias Analysis

To detect potential bias, we will use data slicing, which means breaking down model performance across different subgroups within the dataset.
In our project, the relevant slices will be:

- Category-based slices → harassment vs. policy vs. toxicity, etc.
- Text length-based slices → short, medium, and long prompts.

Once LLM evaluation and judge scoring are integrated, we will calculate metrics such as Attack Success Rate (ASR) and Refusal Accuracy for each slice.
If we observe that a model has a higher ASR for certain categories (for example, it gets jailbroken more easily in harassment than policy), this will be considered category-based bias.

Tools such as Fairlearn, TensorFlow Model Analysis (TFMA) or custom evaluation scripts will be used later to automate this slicing and comparison.

3. Mitigation of Bias (Planned for Later Stages)

If we detect that the model is more vulnerable to certain categories, we will consider the following mitigation strategies:

- Data-level mitigation:
  - Collect or generate more prompts for underrepresented categories.
  - Balance the dataset to ensure each category is fairly represented.
- Model-level mitigation:
  - Fine-tune or adversarially train the model specifically on its weak categories.
  - Introduce category-aware safety prompts or guardrails.
- Evaluation-level mitigation:

  - Adjust decision thresholds or refusal triggers based on category risk.

Since we are currently only in the preprocessing phase, these mitigation steps are planned but not yet applied. They will be implemented after we have model responses and judge evaluations.

4. Documentation of Bias Detection and Mitigation

We will carefully document:

- Definition of bias for this project → A model is biased if it is significantly more vulnerable to adversarial prompts in one category than others.
- Method of bias detection → Category-based and text length-based slicing followed by ASR and refusal score comparison.
- Current pipeline status → No demographic bias analysis yet due to the absence of sensitive attributes in the dataset.
- Future mitigation process → If category-level bias is observed, we will apply data balancing, category-based adversarial training, and improved safety prompting.
- Trade-offs → Any reduction in model utility, accuracy or response diversity due to mitigation strategies will be clearly reported.