Ethical Reflection

Prompt: Your predictive model from Task 3 is deployed in a company. Discuss potential biases and how to address them.

Answer:

If the predictive model from Task 3 were deployed to prioritize bug-fixing resources, several biases could arise, leading to unfair outcomes.

Potential Biases:

1. **Underrepresented Teams/Features:** The dataset's "priority" labels are likely based on past decisions. If certain teams (e.g., a new team working on a less-established product) had their bugs historically under-prioritized, the model will learn and perpetuate this pattern. Their future bugs may always be predicted as "Low" priority, hindering their project's development.
2. **Reporter Bias:** Bugs reported by senior engineers or managers might have been labeled as "High" priority more often than identical bugs reported by junior engineers. The model could then bias its predictions based on the reporter's identity rather than the bug's actual technical severity.
3. **Data Drift:** The model is trained on data from a specific time. If the company's strategic focus shifts (e.g., from desktop to mobile), the old "priority" definitions become obsolete and biased against new initiatives.

Addressing Biases with IBM AI Fairness 360 (AIF360):

The IBM AI Fairness 360 toolkit provides a comprehensive set of algorithms to detect and mitigate bias throughout the ML pipeline.

- **Detection:** I would use AIF360's metrics, such as **Disparate Impact Ratio** and **Average Odds Difference**, to check if the model's predictions are fair across different subgroups (e.g., different development teams). This would quantify any bias present.
- **Mitigation:** After detecting bias, I could apply a mitigation technique from the toolkit. For example, **Reweighing** is a pre-processing algorithm that would assign weights to the training examples (the historical bugs) to ensure that protected groups (like the new team) are fairly represented before the model is even trained. This helps create a fairer model from the outset.