Part 1: Theoretical Analysis

Q1: How do tools like GitHub Copilot help save time, and what can't they do well?

AI tools like GitHub Copilot help you code faster by suggesting ready-to-use lines of code while you're typing. They save time by handling repetitive tasks and even help you discover new ways to solve a problem.

But they're not perfect. Sometimes they suggest code that doesn't fit your project or even introduce bugs. They don't always understand your goals, and they might repeat bad habits from the code they were trained on. It's still up to you to check and improve what they suggest.

Q2: What's the difference between supervised and unsupervised learning for bug detection?

Supervised learning means the model learns from labeled data—like past examples where we already know which code had bugs and which didn't. This helps it spot similar issues in the future.

Unsupervised learning doesn't use labels. Instead, it tries to find patterns or strange behaviour in the code, like an unusual change in a file. This can help find bugs we didn't know to look for.

Supervised is great when we have good data. Unsupervised is helpful when we don't.

Q3: Why is it important to fix bias when using AI to personalize user experiences?

Personalized AI tries to recommend things based on what users like. But if the data used to train it is biased, like showing mostly one type of content or favoring one group, it can leave others out or treat them unfairly.

Fixing that bias makes the AI more fair and inclusive. Everyone gets a better experience, and it helps build trust with your users.

Answer: How does AIOps improve software deployment efficiency? Provide two examples.

AIOps (Artificial Intelligence for IT Operations) improves software deployment efficiency by using machine learning and real-time analytics to automate repetitive tasks, predict failures, and optimize performance without human intervention. It speeds up delivery pipelines, reduces downtime, and helps teams catch issues before they impact users.

Example 1: Harness – Automatic Rollbacks Harness uses AI to monitor deployments in real time. If something goes wrong—like a sudden drop in system performance—it can automatically roll back the changes, preventing user impact. This means less time spent debugging failures and faster recovery from issues.

Example 2: CircleCI – Smart Test Optimization CircleCI uses AI to analyze past test results and figures out which test cases are most likely to fail. It runs those first, so developers get faster feedback. This saves time during testing and helps developers fix bugs earlier in the pipeline.

Part 3: Ethical Reflection (10%)

- **Prompt**: Your predictive model from Task 3 is deployed in a company. Discuss:
    - Potential biases in the dataset (e.g., underrepresented teams).
    - How fairness tools like IBM AI Fairness 360 could address these biases.

Ethical Reflection

When this model is used in a real company, we have to be careful about fairness. One risk is that the data may already be biased. For example, there are fewer records labeled as "High Priority" in the dataset. If this group includes certain races, age groups, or clinical backgrounds that are underrepresented, the model could end up ignoring people who really need urgent care. That wouldn’t be fair, especially if those patients don’t match the majority in the data.

Also, models are only as fair as the data we give them. If the data reflects old patterns of inequality, the model might unknowingly continue that pattern.

To help fix this, tools like IBM AI Fairness 360 can check if the model treats different groups fairly. It can look at how the model performs across race, gender, or other features and spot where it's making unfair decisions. It can also suggest changes to the data or the model to make things more balanced. That helps make sure the model’s decisions are not just accurate, but also fair and responsible.