Part 1: Theoretical Analysis (30%)

1. Short Answer Questions

Q1: Explain how AI-driven code generation tools (e.g., GitHub Copilot) reduce development time. What are their limitations?

AI-driven code generation tools, like GitHub Copilot, act as a "robot assistant" by automating repetitive tasks, such as writing boilerplate code. This automation frees up time for creativity and accelerates development.

Limitations
Models may not be inclusive, and over-reliance on AI is a challenge, emphasizing that the solution involves AI + human collaboration ("Botho").

Q2: Compare supervised and unsupervised learning in the context of automated bug detection.

In automated bug detection, supervised learning is ideal for known bug types, as it relies on labelled datasets where examples of buggy and clean code are provided. The model learns to identify patterns associated with bugs and can predict issues in new code with high accuracy, provided there's enough labelled data.

While unsupervised learning helps uncover hidden or unexpected issues. It doesn't require labelled examples, as it analyses code to find anomalies or unusual patterns instead that might indicate bugs, especially when labelled data is scarce,

Q3: Why is bias mitigation critical when using AI for user experience personalization?

Bias mitigation is crucial because AI is used as the "empathetic friend" for UX to build software people love through features like personalization and the models need to be trained well to ensure the system embodies the user experience personalization.

2. Case Study Analysis

Read the article: AI in DevOps: Automating Deployment Pipelines.

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

AIOps improves software deployment efficiency by transitioning operations from reactive troubleshooting to proactive prediction and automated action. By applying machine learning to analyse massive volumes of operational data (logs, metrics, events), AIOps ensures faster delivery and greater system stability.

Examples of software deployment efficiency:

Intelligent Release Risk Management: AIOps analyses current changes against historical patterns to predict the risk of failure before a full deployment, automatically halting or rolling back risky releases to prevent costly production outages.

Automated Cloud Resource Optimization: During and after deployment, AIOps continuously monitors resource needs and uses predictive analytics to automatically scale cloud resources up or down, which ensures optimal performance without manual intervention.

Task 1: AI-Powered Code Completion

1.Compare the AI-suggested code with your manual implementation.

Both versions sort the list correctly however, the manual code assumes all dictionaries have the key. The AI code uses .get() to handle missing keys safely and the AI adds robustness by preventing errors when data is incomplete.

2.Document which version is more efficient and why.

The AI version is more efficient in real-world use because it avoids crashes and handles edge cases better.

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).

Biased models can undervalue issues from underrepresented teams, reinforce workplace inequalities and reduce trust in AI systems. From the task 3 Kaggle Breast Cancer Dataset, we can see that there is a subtle unfair prioritization of issues from certain teams or demographics from the fairness check done.

Bias Detection Results

- Mean difference: 0.0319 → Indicates a small but measurable gap in priority predictions between simulated gender groups.

- Disparate impact: 1.0895 → Slightly above the ideal range (0.8–1.25), suggesting mild bias but not extreme.

How fairness tools like IBM AI Fairness 360 could address these biases.

Tools like IBM AI Fairness 360 can audit datasets before training, apply mitigation techniques like reweighing or adversarial debiasing and monitor fairness metrics alongside accuracy and F1-score.
Bias Mitigation with Reweighing

After reweighing:

- Mean difference: 0.0

- Disparate impact: 1.0

Reweighing successfully balanced the dataset, ensuring equal treatment across protected groups. This shows how fairness tools can correct bias before model training.