Part 1: Theoretical Analysis

Q1: AI-driven code generation tools

Answer: AI-driven code generation tools like GitHub Copilot reduce development time by:

1. Suggesting complete code snippets based on comments or function names
2. Automating boilerplate code creation
3. Offering instant solutions to common programming patterns
4. Reducing context switching between documentation and IDE

Limitations:

1. May suggest insecure or inefficient code
2. Can't understand broader project context
3. Might generate code that violates licensing
4. Limited ability to handle complex business logic

Q2: Supervised vs unsupervised learning for bug detection

Answer:

- Supervised learning: Uses labeled datasets (bug/no-bug) to train models. Effective for known bug patterns but requires extensive labeled data.
- Unsupervised learning: Detects anomalies without labeled data. Can find novel bugs but may have higher false positives.

Q3: Bias mitigation in UX personalization

Answer: Bias mitigation is critical because:

1. Biased recommendations can alienate user groups
2. May reinforce stereotypes or discrimination
3. Could create legal/compliance issues
4. Leads to poor user experience for underrepresented groups

Case Study Analysis: AIOps in DevOps

Answer: AIOps improves deployment efficiency by:

1. Predictive failure analysis: Using historical data to predict deployment failures before they occur
2. Automated rollback: Intelligently deciding when to rollback based on real-time metrics