

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

Answer:

AI-driven code generation tools like GitHub Copilot reduce development time by providing real-time code suggestions, completing boilerplate code, and offering intelligent auto-completions. They accelerate coding tasks, reduce context-switching, and help developers focus on logic rather than syntax.

Limitations:

- Inaccurate suggestions that can introduce bugs if not reviewed
- Lack of context awareness for complex business logic
- Security risks due to potential inclusion of vulnerable or copyrighted code
- Over-reliance which may weaken core programming skills over time

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

Answer:

Supervised learning uses labeled data to train models to classify or predict bugs. It works well when historical bug datasets are available.

Unsupervised learning, on the other hand, doesn't rely on labels. It can detect anomalies or unusual patterns in code, logs, or runtime behavior — useful for identifying previously unseen or zero-day bugs.

Key difference: Supervised is good for known patterns, unsupervised is better for discovering unknown issues.

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

Answer:

Bias mitigation is critical because AI-driven personalization systems often use demographic, behavioral, or preference data that may reinforce stereotypes or exclude certain user groups. This can lead to:

Unfair treatment of users e.g gender or racial bias

Homogenized content that narrows the user's experience

Loss of trust in the platform

Mitigating bias ensures personalization is inclusive, diverse, and aligned with ethical AI principles.

2. Case Study Analysis

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

AIOps enhances deployment efficiency by automating routine tasks, reducing human error, and enabling faster response to issues through intelligent insights.

examples:

Automated Rollbacks: AIOps systems can detect anomalies in new deployments (e.g., performance drops or error spikes) and automatically trigger rollbacks, preventing user-facing outages.

Predictive Resource Management: AIOps tools analyze historical usage data to forecast resource needs, automatically scaling infrastructure during deployments — ensuring optimal performance without overprovisioning.