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

Answer:

AI code generation tools reduce developmen time by:

Auto-completing code based on context

Generating boilerplate or repetitive code automatically.

Suggesting function logic from comments or natural language.

Reducing time spent switching between documentation and coding.

Limitations:

May produce incorrect, inefficient, or insecure code.

Lacks deep understanding of business rules or project structure.

Can lead to over-reliance by developers.

Potential legal or ethical concerns from generated code resembling licensed material.

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

Answer:

Supervised learning:

Requires labeled data (e.g., code marked as "buggy" or "clean").

Learns from historical bug data to detect similar issues in future code.

Provides higher accuracy for known bugs.

Unsupervised learning:

Works with unlabeled code.

Detects unusual or anomalous patterns that may indicate new bugs.

Useful when labeled data is unavailable but may produce false positives.

Comparison Summary:

Supervised is best for detecting familiar bugs with labeled datasets.

Unsupervised is better for uncovering unexpected or rare issues.

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

Answer

Bias mitigation ensures fairness and inclusivity in AI-driven experiences.

Unmitigated bias can:

Reinforce harmful stereotypes.

Exclude or misrepresent certain user groups.

Create unfair experiences (e.g., biased recommendations or access restrictions).

Bias-aware systems promote user trust, ethical design, and equitable access.

Techniques like diverse training data and fairness audits are essential to minimize harm.

2. Case Study Analysis

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

Answer:
AIOps (Artificial Intelligence for IT Operations) improves deployment efficiency by automating complex DevOps tasks, reducing human error, and enabling real-time insights.

Examples:

Automated Root Cause Analysis

AIOps tools analyze logs and metrics to quickly pinpoint the source of a failed deployment or performance degradation, reducing downtime and debugging effort.

Predictive Resource Scaling

AI models predict usage spikes and auto-scale resources in deployment environments, ensuring smooth rollouts and minimizing resource waste.