POWER LEARN PROJECT AFRICA
AI FOR SOFTWARE ENGINEERING

NAME : ERICK WAMBUGU
WEEK 4 ASSIGNMENT

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

- AI-driven code generation tools, uses machine learning models trained on large code-bases to predict and suggest codes as developers type. They reduce development time by:
  - Auto- completing functions and logic structure codes
  - Reducing boilerplate coding time
  - Accelerating learning
  - Improving productivity
- LIMITATIONS
  - Context misunderstanding – may generate incorrect or insecure codes
  - over-reliance risk – developers accepting suggestions without verification
  - limited domain understanding – lacks business logic
  - possible copyright and licensing issues

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

Supervised learning is precise and has high accuracy but data-dependent since it requires extensive labeled data for training which is costly

unsupervised learning is adaptive but less accurate since it works with unlabeled data to find anomalies and unusual patterns detecting code blocks that deviates statistically from normal patterns and it may produce false positives since not all anomalies are bugs.

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

Bias mitigation ensures AI-driven personalization systems treats users fairly and inclusively in order to mitigate:

- unfair recommendations

PART 2: CASE STUDY ANALYSIS:

How does Ai Ops improve software development efficiency? Provide two examples.

- Predicting build failures and optimizing CI/CD workflows - Using historical data from previous builds and tests, AI models can predict which builds are likely to fail and can prioritize test cases most likely to yield value. AI-driven pipelines can automatically roll back failed deployments, minimizing human intervention and reducing downtime or waste from faulty releases. These capabilities reduce manual oversight, shorten feedback cycles, and help get stable deployments out faster.

POWER LEARN PROJECT AFRICA
AI FOR SOFTWARE ENGINEERING

NAME : ERICK WAMBUGU
WEEK 4 ASSIGNMENT

- Automated monitoring, anomaly detection & self-healing infrastructure - AI-based monitoring tools analyses metrics, traces and logs in real time to detect anomalies or degradation before users are impacted. Infrastructure as Code (IaC) combined with AI can recommend optimized configurations, detect mis-configurations or security vulnerabilities in infrastructure, and dynamically scale or adjust resources for optimal performance (and cost) in the deployment environment. These reduce manual monitoring, lessen hidden downtime or performance issues post-deployment, and allow the deployment pipeline to succeed reliably and faster.

Examples:
1. Azure/Netflix-style canary deployment with AI
2. AI-driven resource allocation and IaC optimization