Part 1 Theory week 4

# 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 reduce development time by quickly suggesting code snippets, completing functions, and generating boilerplate code based on natural language comments or partial inputs. This helps developers avoid repetitive coding tasks and focus on logic and problem-solving. These tools also assist in learning new syntax or frameworks by offering real-time examples, which speeds up onboarding and prototyping. However, their limitations include a lack of deep understanding of the project’s full context, potential generation of insecure or buggy code, and over-reliance by developers, which may reduce critical thinking. They also sometimes suggest outdated or inefficient solutions, and may replicate licensed or copyrighted code without clarity.

**Q2: Compare supervised and unsupervised learning in the context of automated bug detection.** In automated bug detection, **supervised learning** involves training an AI model using labeled data—examples of code that are marked as “buggy” or “clean.” This allows the model to learn patterns associated with bugs and predict future occurrences in similar contexts. It is effective but depends heavily on the quality and size of the labeled dataset. On the other hand, **unsupervised learning** works without labeled data and instead detects anomalies or unusual code patterns that differ from the norm. This is useful for identifying unknown or emerging bugs, but it may produce more false positives because not all anomalies are actual bugs. In summary, supervised learning is more accurate for known issues, while unsupervised learning is better for discovering new or unexpected bugs.

**Q3: Why is bias mitigation critical when using AI for user experience personalization?** Bias mitigation is crucial in AI-driven user experience personalization because biased algorithms can lead to unfair, discriminatory, or non-inclusive experiences. If the training data reflects stereotypes or lacks diversity, the AI may personalize content, recommendations, or interfaces in ways that favor certain user groups over others, leading to a poor or even harmful user experience. For example, a biased recommendation system might repeatedly show certain products or content only to specific demographics, limiting choice and reinforcing societal biases. Mitigating bias ensures that personalization is ethical, inclusive, and accurately reflects the needs and preferences of all users, which builds trust and improves overall satisfaction.

# 2. Case Study Analysis

**sHow does AIOps improve software deployment efficiency? Provide two examples.**

* AIOps improves software deployment efficiency by using machine learning and data analysis to monitor, optimize, and automate different stages of the deployment pipeline. It helps detect risks early, resolve issues faster, and reduce human intervention, leading to more reliable and faster releases.
* One example is **predictive deployment and anomaly detection**—AIOps tools analyze real-time logs and metrics to identify patterns that often lead to deployment failures. When risky code changes are detected, the system can alert teams or automatically halt the release before it impacts users.
* Another example is **automated self-healing and rollbacks**. If an error occurs during deployment, such as a performance drop or failed tests, AIOps can instantly trigger a rollback to the last stable version or auto-adjust resources to fix the issue. These actions significantly reduce downtime and the need for manual fixes.