

Part 1: Theoretical Analysis

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

Modern code completion tools such as GitHub Copilot enable faster developmental cycles, better productivity, and automation abilities. As a result, they reduce development time through improve productivity, automating repetitive tasks, and improving code efficiency by addressing challenges such as code quality concerns, security risks, and ethical considerations (Patel, 2025). However, these tools have their downsides linked to code reliability, security vulnerabilities, loss of developer control, and technical debts risks. AI-generated code has errors, bugs, and vulnerabilities and needs adequate testing and code reviews. Additionally, AI might introduce vulnerabilities to developer's codes unintentionally due to poorly implementation authentication or exposure to common exploits. Over-reliance on AI tools minimizes hands-on experiences and understanding. Lastly, high reliance on AI for quick fixes or repetitive tasks might results in codes that work short-term, but presents long-term maintenance challenges. Overtime, this can slow down the developmental process and increase costs

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

Supervised machine learning relies on labelled data sets, while unsupervised learning uses machine learning algorithms to cluster and analyze unlabeled data sets to provide data insights. Within the context of automated bug detection, in supervised learning, the AI model is provided a dataset of code examples such as "bug" or "no bug". The model then learns common patterns in the buggy code and later scans new code and predicts whether it is a big or not. On the other hand, in unsupervised learning, the AI model learns from unlabeled data by looking for

anomalies or unusual patterns in the code. The model then flags the code that looks different from the rest as one which can be a bug.

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

According to Para (2024), when it comes to user experience personalization, personalization algorithms have become increasingly complex and intricate. Without bias mitigation, personalization algorithms may inadvertently shape the hyper-personalized content consumers receive, potentially reinforcing stereotypes as a result limiting exposure to diverse content and entrenching social inequalities (Para, 2024). Therefore, bias mitigation is critical in user experience personalization to:

- Prevent reinforcing existing biases and stereotypes
- Preventing unequal access to information or opportunities
- Maintaining user trust and engagement
- Promoting fairness and inclusivity
- Enhancing AI performance

References

- Para, R. K. (2024). The Role of Explainable AI in Bias Mitigation for Hyper-personalization. *Journal of Artificial Intelligence General science (JAIGS)* ISSN: 3006-4023, 6(1), 625-635 <http://newjaigs.com/index.php/JAIGS/article/view/289>
- Patel, A. (2025). *The Impact of AI on Software Development: A Case Study on Copilot & ChatGPT*.
<https://www.researchgate.net/publication/390299524> The Impact of AI on Software Development A Case Study on Copilot ChatGPT