AI Applications in Software Engineering

Submitted by: James Njoroge

📑 Table of Contents

□ Part 1: Theoretical Analysis

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 assist developers by auto-completing code, suggesting functions, and generating boilerplate code. This reduces development time by speeding up the coding process, minimizing manual errors, and aiding in learning new APIs. However, their limitations include:

- Context insensitivity: Copilot might suggest irrelevant or insecure code.
- Over-reliance: Developers might blindly accept suggestions.
- Lack of project-wide understanding.
- Possible IP and licensing issues with generated code.

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

Supervised learning requires labeled datasets (e.g., buggy vs. clean code) and can learn patterns to classify or predict bugs.
Unsupervised learning, on the other hand, doesn't need labeled data. It detects anomalies or clusters of unusual code patterns, which could indicate bugs.
Supervised learning is accurate with quality data, while unsupervised is flexible and exploratory.

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

Personalization systems trained on biased data may reinforce stereotypes, exclude minority groups, or provide unequal service quality.
Bias mitigation ensures fairness, inclusivity, and trustworthiness in user experience.

Case Study: How AIOps Improves Software Deployment Efficiency

AIOps improves deployment efficiency by using AI to automate anomaly detection, root cause analysis, and proactive incident resolution.

Examples:

1. Predictive autoscaling based on usage patterns.
2. Real-time log analysis to identify failed deployments before users are impacted.

🛠️ Part 2: Practical Implementation

2.1 Ai-Powered Code Completion

AI-Suggested Version (GitHub Copilot) (Test 1)

Analysis (200 words)

The AI-suggested version using Python's built-in sorted() function is more concise, readable, and efficient than the manual implementation using nested loops. While both versions achieve the same result, the AI-suggested code leverages Pythonic practices, resulting in better performance and maintainability. In contrast, the manual version is more verbose and less efficient ($O(n^2)$ complexity). The AI version also reduces cognitive load and enhances code clarity.

2.2 Automated Testing with AI

Selenium Test Script (Test 2)

Summary (150 words)

The Selenium test case automates login attempts with both valid and invalid credentials. AI tools like Testim.io or AI-enhanced Selenium plugins improve test coverage by quickly adapting to UI changes, automatically generating new test cases, and identifying previously undetected edge cases. This enhances the reliability of regression testing and accelerates development. Additionally, AI can predict potential failure paths and suggest more robust test strategies.

2.3 Predictive Analytics for Resource Allocation

Modeling Using the Breast Cancer Dataset (Test 3)

Performance Metrics

✅ Accuracy: 0.96

✅ F1 Score: 0.97

□ Part 3: Ethical Reflection

Biases in the Breast Cancer dataset may stem from overrepresentation of certain demographics (e.g., race, gender). If used for resource allocation in companies, this could disadvantage underrepresented groups.

Fairness tools like IBM's AI Fairness 360 offer bias detection, preprocessing, and postprocessing techniques. For example, reweighting can help balance datasets, while metrics like disparate impact help monitor fairness.

□ Bonus Task: Innovation Proposal

Tool Name: AutoDocAI

Purpose

Automatically generate function-level and module-level documentation using NLP.

Workflow

Parse source code into AST (Abstract Syntax Tree).

Use a GPT-like model to summarize function purpose, parameters, and outputs.

Insert docstrings and output structured markdown documentation.

Impact

Saves developer time.

Improves maintainability.

Standardizes documentation for better onboarding and auditing.

📸 Screenshots

Screenshots are uploaded

📚 References

- GitHub Copilot Documentation
- Selenium Official Docs
- Kaggle Breast Cancer Dataset
- IBM AI Fairness 360 Toolkit
- AI in DevOps: Automating Deployment Pipelines Article