AI in Software Engineering - Assignment Summary

Part 1: Theoretical Analysis

Q1: How AI Code Generation Tools Reduce Dev Time

AI tools like GitHub Copilot assist developers by suggesting code based on context and prompts. They reduce the time spent on repetitive tasks, improve productivity, and help beginners. However, limitations include producing incorrect or insecure code and requiring human review.

Q2: Supervised vs. Unsupervised Learning for Bug Detection

Supervised learning uses labeled data (e.g., bug/no bug) and is good for detecting known issues. Unsupervised learning identifies patterns or anomalies in logs, useful for unknown bugs. Both are valuable in different bug detection scenarios.

Q3: Bias in Personalization AI

Bias in training data leads to unfair or exclusive personalization. AI trained on non-diverse data may fail to serve all users equally. Mitigating bias ensures fairness and a more inclusive experience.

Case Study: AIOps in DevOps

AIOps enhances deployment efficiency by analyzing logs and predicting issues. It can auto-restart failed services and scale resources in real time, reducing downtime and human intervention.

Part 2: Practical Implementation

Task 1: AI Code Completion

Copilot suggested a function to sort dictionaries efficiently. Compared to manual coding, it saved time and provided accurate suggestions. Human review was necessary to ensure correctness.

Task 2: AI Testing with Selenium in Colab

Using Selenium in headless mode, I tested login functionality with valid and invalid credentials. Selenium adapted to the page structure and performed automated checks. This improved testing efficiency and reliability.

Task 3: Predictive Analytics

I trained a Random Forest model on the Breast Cancer dataset. After preprocessing and evaluation, the model achieved over 95% accuracy and high F1-score, showing effective classification of the target variable.

Part 3: Ethical Reflection

The dataset used may be biased toward specific groups (e.g., older women). This can result in unfair or inaccurate predictions. Fairness tools like IBM AI Fairness 360 can help detect and correct these biases, improving equity in AI applications.

Bonus Task: AI Tool Proposal

Tool Name: AutoDocAI

Purpose: Automatically generates human-readable documentation from Python source code using NLP.

Workflow: Scans functions and comments -> Converts to Markdown docs.

Impact: Saves developer time, improves understanding, and enhances team collaboration.