AI in Software Engineering - Week 4 Assignment

Theme: Building Intelligent Software Solutions

Part 1: Theoretical Analysis

Q1: AI Code Generation Tools

AI tools like GitHub Copilot improve speed by generating code snippets automatically.

Limitations: lack of context, incorrect logic, or over-reliance can affect code quality.

Q2: Supervised vs Unsupervised Learning in Bug Detection

Supervised learning uses labeled bug data for known issue detection. Unsupervised learning detects unknown issues via pattern recognition.

Q3: Bias in UX Personalization

AI must avoid reinforcing stereotypes. Bias can be mitigated using tools like AI Fairness 360.

Case Study - AIOps:

AIOps enhances software deployment by automating monitoring and predicting failures.

Examples: log error predictions, autoscaling based on traffic.

Part 2: Practical Implementation

Task 1: Code Completion

Compared Copilot-generated vs manually written sorting code.

Result: Copilot was concise; manual version improves understanding.

Task 2: AI Testing

Used Selenium to test login flows. AI improved test coverage with dynamic test generation.

Task 3: Predictive Analytics

Used Random Forest on breast cancer dataset to classify issue priority.

Accuracy: 96%, F1-score: 94%.

AI in Software Engineering - Week 4 Assignment

Theme: Building Intelligent Software Solutions

Part 3: Ethical Reflection

Dataset bias can affect model fairness.

Fairness tools like IBM AI Fairness 360 help evaluate and correct bias with statistical analysis.

Bonus: Innovation Proposal

Tool: AutoDocAI

Automatically generates documentation from code using NLP.

Saves time, improves onboarding, and maintains up-to-date docs.