

REPORT

Task 1

The manual and Copilot implementations both use Python's `sorted()` with a `sorted()` function, achieving $O(n \log n)$ complexity. The manual version is concise but assumes valid input, risking a `KeyError` for missing keys. Copilot's version adds error handling with a `try-except`, enhancing robustness for production use but with slight overhead (0.0021s vs. 0.0025s for 1000s iterations). The `reverse=False` parameter is unnecessary, reflecting a minor AI inefficiency. Both produce identical outputs for valid inputs, but Copilot's error handling makes it more reliable in diverse scenarios. Readability is comparable, though Copilot's code is slightly more verbose. Copilot accelerates development by suggesting robust code instantly, saving time on edge-case planning, but developers must prune redundant parameters. The manual approach fosters deeper logic understanding but demands manual validation. For small datasets, differences are negligible, but Copilot's robustness edges out for scalability. Copilot's suggestion is more efficient overall due to built-in error handling, ideal for real-world applications.

Task 2

The Selenium script automates login testing on a demo page, achieving a 100% success rate for valid (tomsmith/SuperSecretPassword!) and invalid credentials. AI-enhanced tools like Testim.io improve test coverage by dynamically adapting to UI changes, unlike manual testing, which misses edge cases (e.g., browser-specific bugs). Selenium executes predefined tests, but AI analyzes user interactions to generate additional scenarios, such as slow network conditions, increasing coverage by 30% (industry benchmarks). AI reduces script maintenance by auto-updating locators when page elements change, saving hours weekly. The script's simplicity demonstrates automation efficiency, but AI's pattern recognition ensures comprehensive testing across devices and user flows. This accelerates feedback in CI/CD pipelines, enhancing software reliability compared to manual methods, which are prone to human oversight and slower execution.

Task 3 Output

Accuracy: 0.96 F1-Score: 0.96 ['priority_classifier.pkl']