In this task, we created a Python function to sort a list of dictionaries by a specific key. Sorting such data structures is a common requirement in many software applications, like organizing user records, filtering product lists, or processing data entries. By using AI-powered code completion tools like GitHub Copilot, we explored how AI can assist in quickly generating efficient and reliable code.

This exercise is significant because it shows how AI tools can save development time while encouraging best practices like handling missing data and adding flexibility. It highlights the practical synergy between human expertise and AI assistance, which is becoming increasingly important in modern software engineering to improve productivity and code quality.

Both the manual and AI-generated versions effectively sort a list of dictionaries by a given key. The **manual version** is clean and efficient, but assumes the key exists in every dictionary. While this works in controlled datasets, it's prone to failure in real-world scenarios with inconsistent data.

The **AI-suggested version** (via GitHub Copilot) is more comprehensive and production-ready. It offers parameters for descending order and a default fallback for missing keys, improving flexibility. For example, when sorting employee records by salary, it avoids crashes if some records don't include a salary field. This reflects how AI tools now anticipate edge cases and promote robust practices.

From a performance standpoint, both use Python's built-in `sorted()` function with O(n log n) complexity. However, the AI-generated version adds error handling and usability enhancements without sacrificing performance.

Ultimately, the AI-suggested solution is **more suitable for real-world use**, as it balances efficiency with robustness. It saves developer time while producing code that is both readable and maintainable. This demonstrates how AI can effectively augment human capabilities, especially in repetitive or logic-driven tasks like sorting or data manipulation.