CS 264: Homework 4 Report

1. Accuracy Result & Observations

I evaluated my ReAct-style SWE-agent on the lynnliu030/swebench-eval-subset benchmark using the model **gpt-5-mini**, with **max_steps = 100**.

Final Accuracy

- **Resolved instances:** 9 / 20
- **Accuracy: 45%**

Observations

- The agent successfully solved bugs requiring:
    - Reading error messages
    - Locating the failing function
    - Making small, localized edits
    - Re-running tests to verify
- Most resolved cases were from Django, SymPy, and scikit-learn, which generally have predictable error signatures.
- Failures usually happened due to:
    - Missing test-driven iteration (model forgot to re-run tests)
    - Over-editing or hallucinating fix locations
    - Context overflow (too many tool outputs accumulated)
    - Patch generation succeeded, but change didn't fully fix behavior

2. Custom Tools Implemented

2.1 run_bash_cmd

Purpose:
Universal execution tool, i.e. shell inspection, running tests, printing files, grepping, etc.

Why:
SWEBench repos vary widely, so this avoids needing many narrow tools.

2.2 generate_patch

Purpose:
Returns the final git diff to submit to SWEBench.

Why:
Required to produce benchmark-compatible output. Prevents hallucinated patch formatting.

2.3 replace_in_file

Purpose:
Precise, line-based editing without rewriting full files.

Why:
LLMs struggle with copying entire files or preserving indentation. This tool constrains modifications and reduces formatting errors.

2.4 show_file

Purpose:
Returns entire file contents.

Why:
Models often need broader context beyond a snippet, especially for import dependencies and class definitions.