



Neo Take-Home

The Challenge

Enterprise teams rely heavily on spreadsheets but face significant workflow challenges including data inconsistency, collaboration difficulties, and time-consuming manual processes.



Your task is to build an AI tool that understands and works with complex spreadsheets.

This task is intentionally unbounded and vague to encourage creativity, thoroughness, and fun! Use this challenge to demonstrate your curiosity and abilities.

Over the course of 24-72 hours, please let your creativity run wild on either (or both!) of the following tracks.

Track A: Excel Agent & Automation

Build an AI agent that can make programmatic modifications to Excel files:

- Parse natural language requests ("add a total row", "what's the estimated cost?")
- Execute simple but useful spreadsheet operations
- Handle real Excel files with formatting, formulas, multiple sheets
- Focus on agent design, tool selection, and user interaction patterns

 **Example deliverable:** Excel plugin using LangGraph to query data from an open spreadsheet.

Track B: Spreadsheet Understanding

Build a system that extracts meaningful structure and relationships from Excel files. Some ideas to consider:

- Formula dependency mapping - parse formulas and create relationship graphs
- Table detection and boundary identification in multi-table sheets
- Extract and categorize different data types (inputs, calculations, outputs)
- Focus on parsing, analysis algorithms, and data extraction accuracy.



Example deliverable: Jupyter notebook reproducing approaches of [SpreadsheetLLM](#).

Test Cases

For your convenience, I have downloaded and attached an example commercial real-estate model for you to use in your testing (attached in email).

Submission

1. Once you are content, please forward me your submission over email along with instructions for how I may run/use it before we discuss live.
2. After submission, please schedule a live interview with me through the [calendly link](#).

Success Criteria

We're more interested in your **technical approach, problem-solving process, and depth technical ability** than achieving a perfect solution. Show us:

- How you break down complex problems
- Your technical architecture decisions and trade-offs
- Quality of code and system design
- How you handle edge cases and real-world messiness

Rules

- AI use as a propellant is encouraged. We will discuss your approach during the live interview.
- Feel free to ask around to brainstorm. Similarly to AI-use, we will discuss your approach and thinking during the live interview.
- Use APIs where helpful and efficient. Don't reinvent the wheel for the purposes of discussion.