Build AI Evals from Scratch for Product Managers and Engineers - Syllabus

The full loop

We take one AI product through the complete evaluation loop, start to finish:

1. Read the production system prompt of a real AI product
2. Build a sample golden dataset with happy cases, edge cases and adversarial cases
3. Run it across multiple models and decide which model to use
4. Run a full experiment on the whole dataset, across multiple models
5. Review and annotate the AI's answers
6. Run error analysis on the annotations to find the failure patterns
7. Pick the right metrics, run the evals and compare models on quality, cost and latency
8. Decide what to improve, how to improve it, and how to check the change with the same evals

This is how you run solution discovery for an AI feature. Each experiment tests one idea: a new prompt, more context, a different model. The evals tell you whether it worked. You keep what improves the results, drop what doesn't, and run the next experiment. In the workshop you go through this cycle yourself, on a real product, with the numbers in front of you.

All the questions the workshop answers

? How do you define quality for an AI product or feature?
? How do you write a rubric to evaluate it?
? How do you create test data, especially when you don't have any data yet?
? Should you write evals before or after shipping, and how do they change as your product grows?
? How do you evaluate a single answer versus a multi-turn conversation or an agent that uses tools?

? Why can't you just let an AI write the evals for your AI product?
? Why shouldn't you rely on template metrics like helpfulness, relevance and correctness?
? How should product, engineering and domain experts work together to write strong evals?
? How do you choose the best model for your AI product?
? How do you use evals to run discovery for an AI feature?
? How do you use AI to scale your evals, and write an LLM judge that matches your product experts' judgment?

By the end, you will have

✓ A clear picture of what a production system prompt looks like
✓ A sample golden dataset with happy cases, edge cases and adversarial cases
✓ Results from a full experiment across multiple models, and a model choice you can explain
✓ A set of AI answers you reviewed yourself, with written annotations
✓ A list of the failure patterns specific to this AI product
✓ Metrics of all types (deterministic checks and LLM judges), applied and read
✓ A comparison of the models on quality, cost and latency, and the trade-offs between them
✓ A plan for what to improve next, and how to check the change by running the same evals again
✓ The full evaluation loop done once, hands on, so you can run it on your own product