~~AI Evals Pitfalls~~

The Revenge of the Data Scientist

Hamel Husain · PyAI 2026

February 11, 2026 Engineering

Harness engineering: leveraging Codex in an agent-first world

By Ryan Lopopolo, Member of the Technical Staff

Listen to article 18 :04 Share

Over the past five months, our team has been running an experiment: building and shipping an internal beta of a software product with **0 lines of manually-written code**.

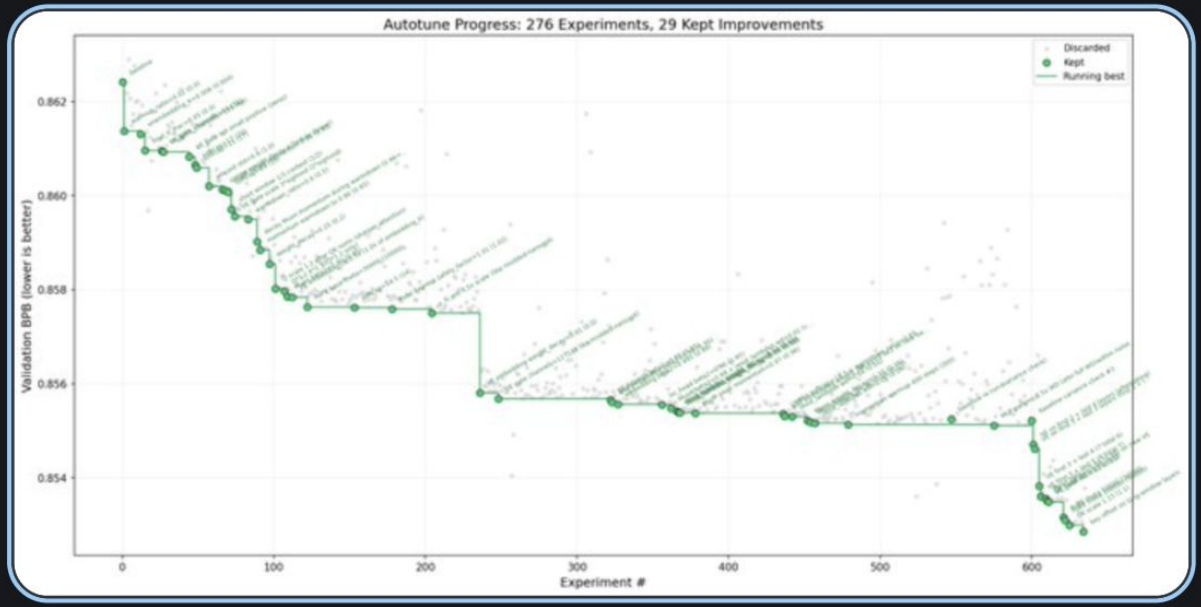
The product has internal daily users and external alpha testers. It ships, deploys, breaks, and gets fixed. What's different is that every line of code—application logic, tests, CI

We did the same for observability tooling. Logs, metrics, and traces are exposed to Codex via a local observability stack that's ephemeral for any given worktree. Codex works on a fully isolated version of that app—including its logs and metrics, which get torn down once that task is complete. Agents can query logs with LogQL and metrics with PromQL. With this context available, prompts like "ensure service startup completes in under 800ms" or "no span in these four critical user journeys exceeds two seconds" become tractable.

Andrej Karpathy @karpathy · 16h

Three days ago I left autoresearch tuning nanochat for ~2 days on depth=12 model. It found ~20 changes that improved the validation loss. I tested these changes yesterday and all of them were additive and transferred to larger (depth=24) models. Stacking up all of these changes,

Show more



702 1.9K 14K 1.6M

The Harness Is Data Science

(A big part of it, anyways)

AI engineers 4 years ago

Examine data w/ stats, visualizations & notebooks

Measure alignment w/human labels

Use a model suited to the task

Designs metrics that align w/ business goals

AI engineers today

It's just vibes yo.

Ask the model if it did a good job

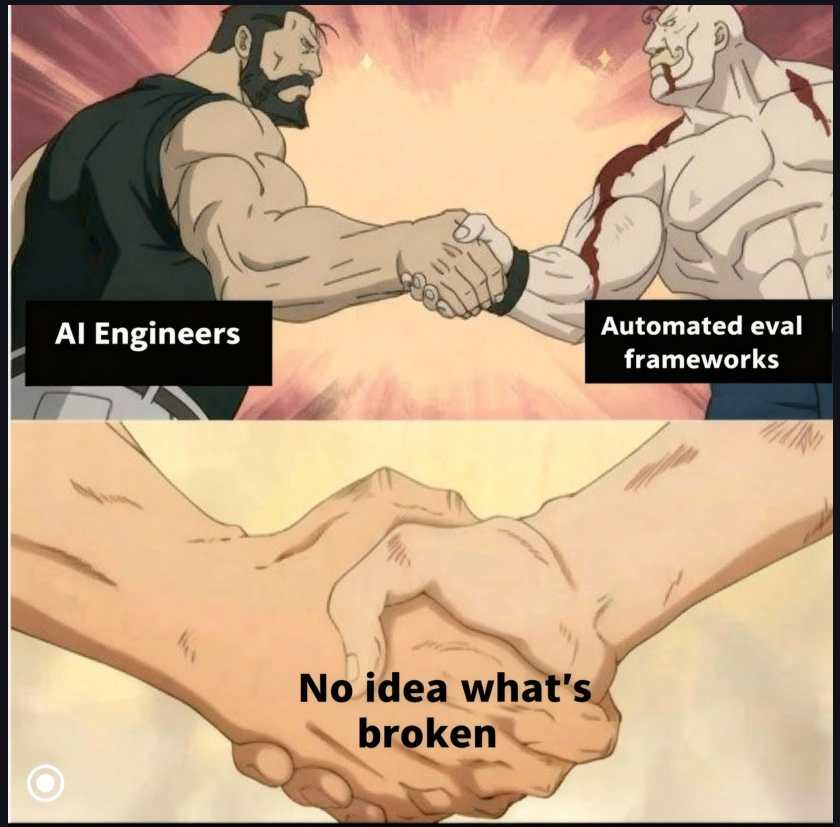
Use LLM API to generate scores on a scale of 1-100

npm install metrics


I don't understand retrieval or evals
Then we'll just say RAG & Evals are dead
imgflip.com

Ok, here’s the pitfalls

Pitfall #1: Generic Metrics



What To Measure?

What most teams do:

Helpfulness Score

Coherence Score

Hallucination Score

What To Measure?

What most teams do:

Helpfulness Score

Coherence Score

Hallucination Score

What a data scientist would do:

Explore the data.

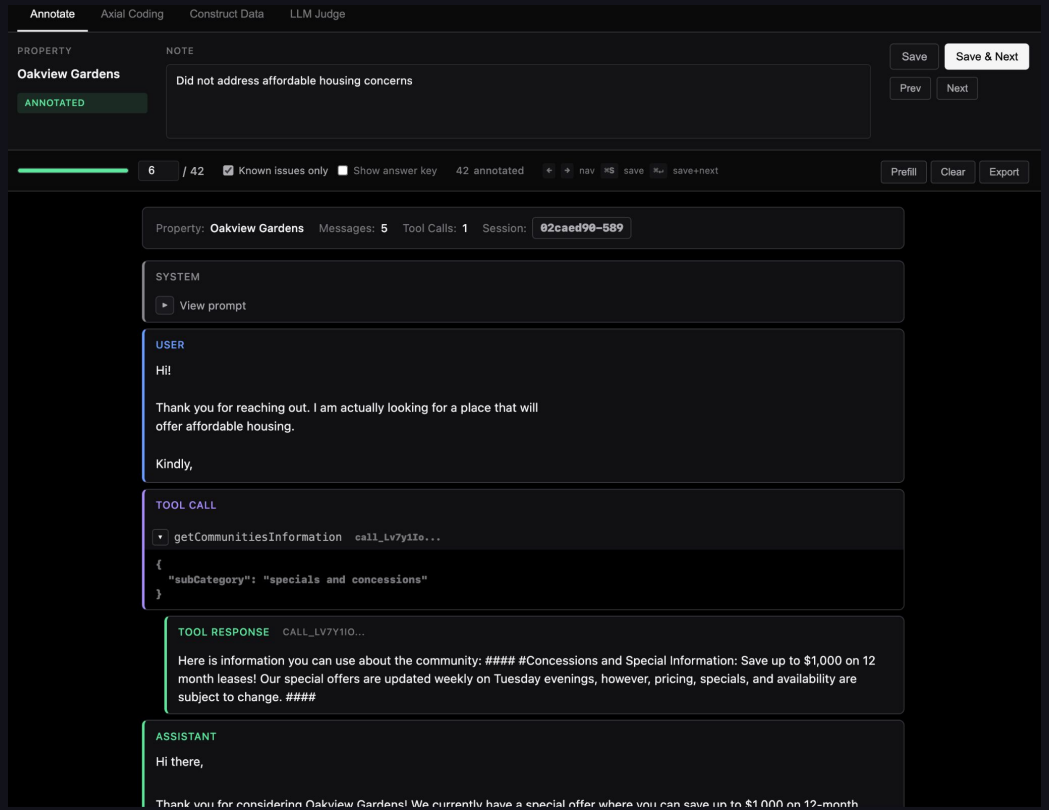
What’s actually breaking?

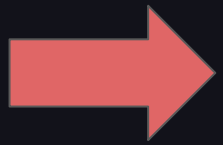
What’s the highest value thing to start measuring?

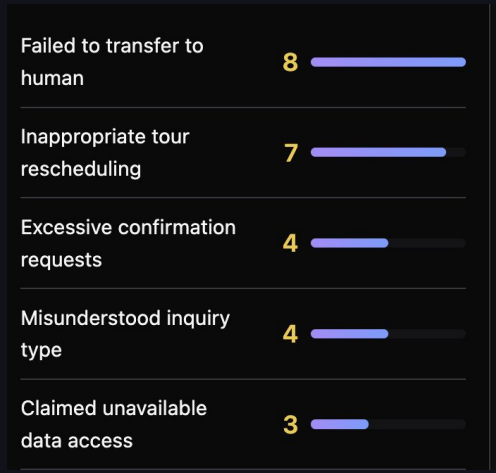
Form new hypotheses.

Analyze, repeat.

Looking At The Data







Don't do this!

- Rouge
- Bleu
- Faithfulness
- Helpfulness
- Tone

Generic Scores

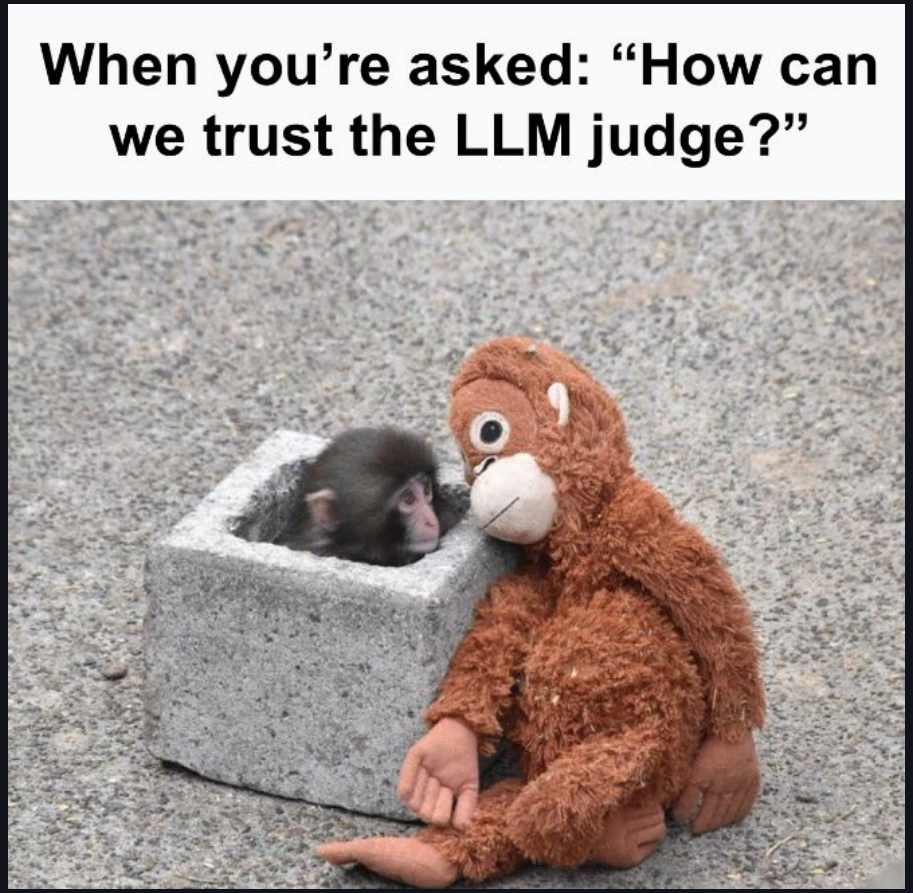
- Calendar Scheduling Failure
- Interrupted Conversation Flow
- Widget Rendering Issue
- Email recipient incorrect
- Failure to Escalate To Human

Application Specific Metrics

Really. Do It.


What are your superpowers again?
I read traces logged by
AI products

Pitfall #2: Unverified Judges



Judge Validation

What most teams do:

Ask an LLM to rate outputs 1–5.
Deploy it. Trust the numbers.

Judge Validation

What most teams do:

Ask an LLM to rate outputs 1–5. Deploy it. Trust the numbers.

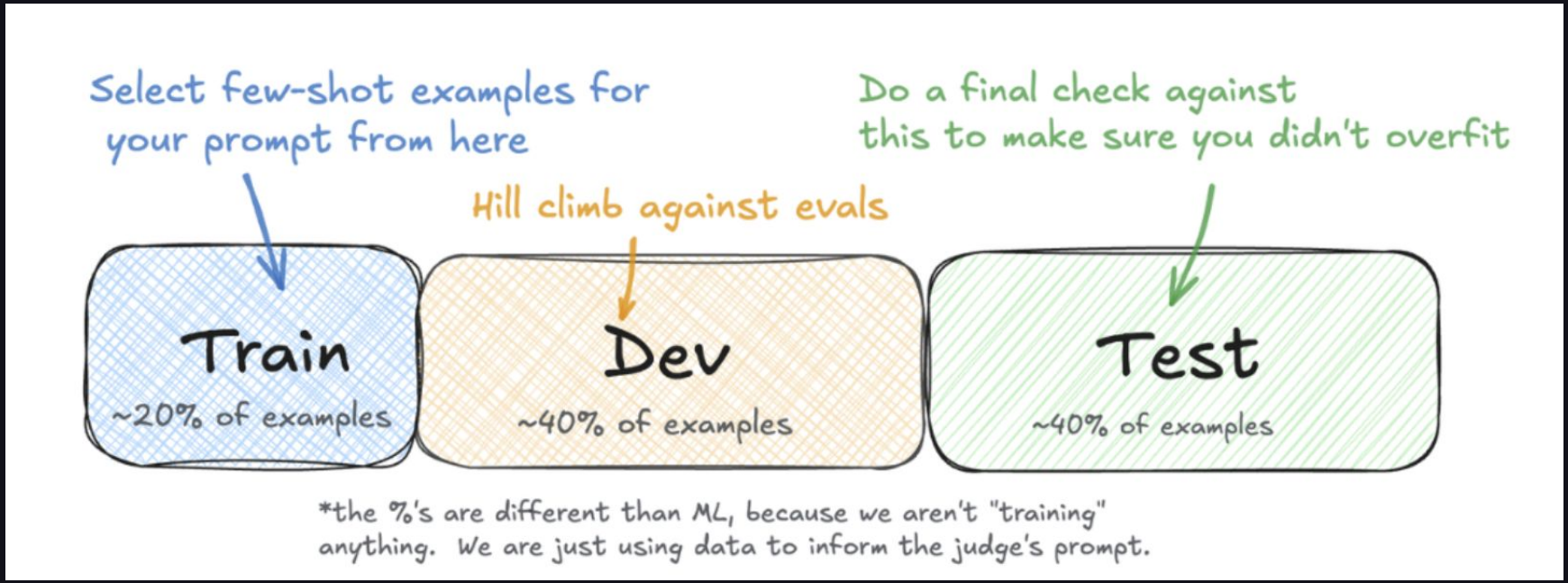
What a data scientist would do:

Treat it like a classifier.

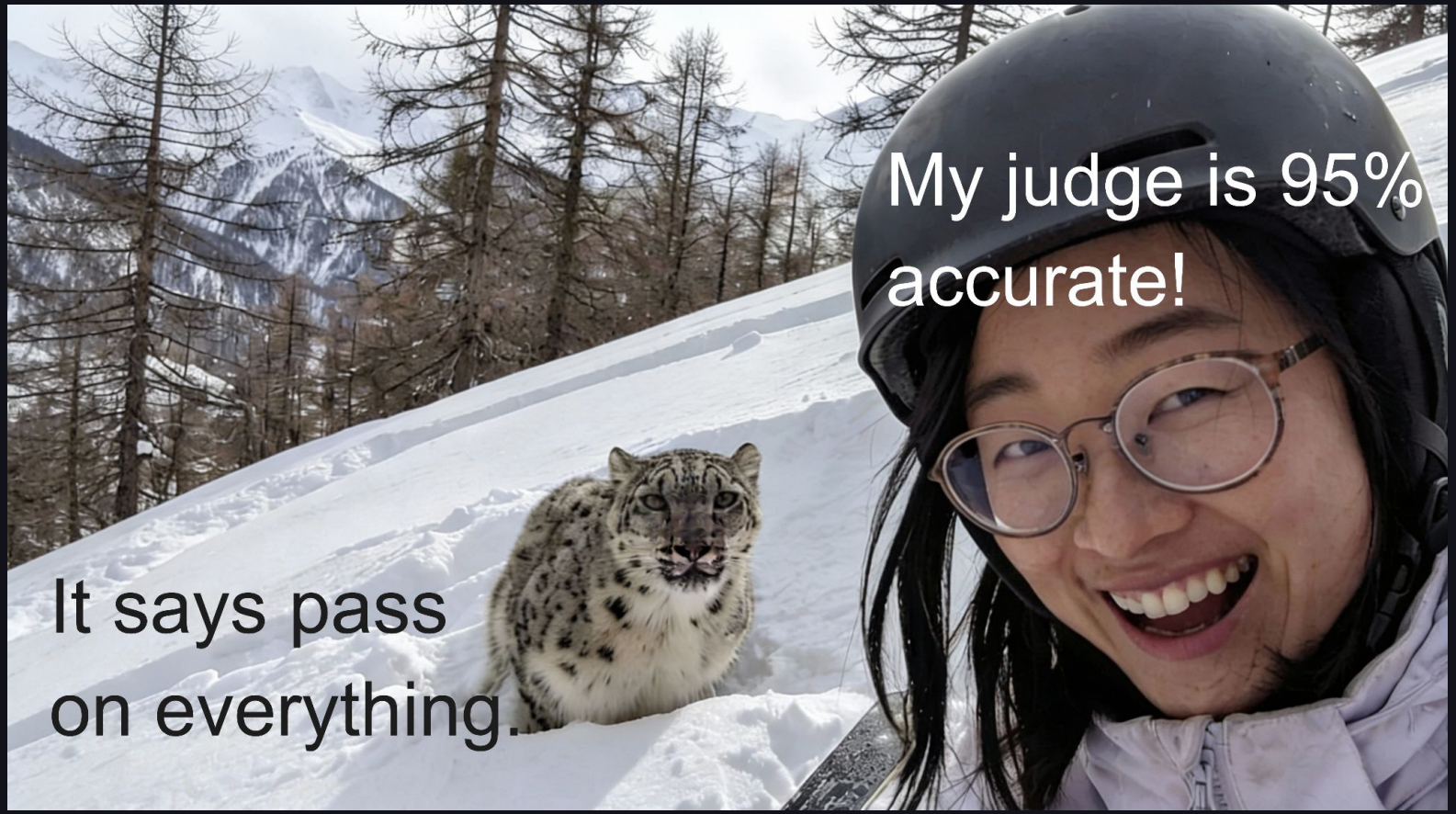
Label 100 outputs pass/fail. Partition the data.

Measure precision & recall on the held-out set.

Treat Like A Classifier



Example: Don’t use “accuracy”



Pitfall #3: Bad Experimental Design

Synthetic eval dataset from prompting 'generate 100 diverse user queries'

Synthetic Data Generation

What most teams do:

Prompt an LLM: ‘Give me 50 test queries.’ Get 50 generic questions.

Synthetic Data Generation

What most teams do:

Prompt an LLM: ‘Give me 50 test queries.’ Get 50 generic questions.

What a data scientist would do:

Use hypothesis to inform which dimensions should be varied.

Generate combinations. Convert to realistic queries.

Review for quality.

Look at the data!

Synthetic Data Generation

Use structured input for diversity. Define key dimensions (e.g., Feature, Persona, Scenario) and use them as variables in your prompt.

When possible, **seed your generation with real logs or traces**. Then, ask the model to explicitly inject changes, like a new constraint or a modified variable, to create realistic edge cases.

Enforce output structure & filter. Define a schema for the output. Generate many candidates, then filter to retain the highest-quality, challenging examples.

Designing Metrics

What most teams do:

Generic Metrics

Bundle entire rubric into one eval

Likert scale: 1-5

Scale of 1-100

Designing Metrics

What most teams do:

Generic Metrics

Bundle entire rubric into one eval

Likert scale: 1-5

Scale of 1-100

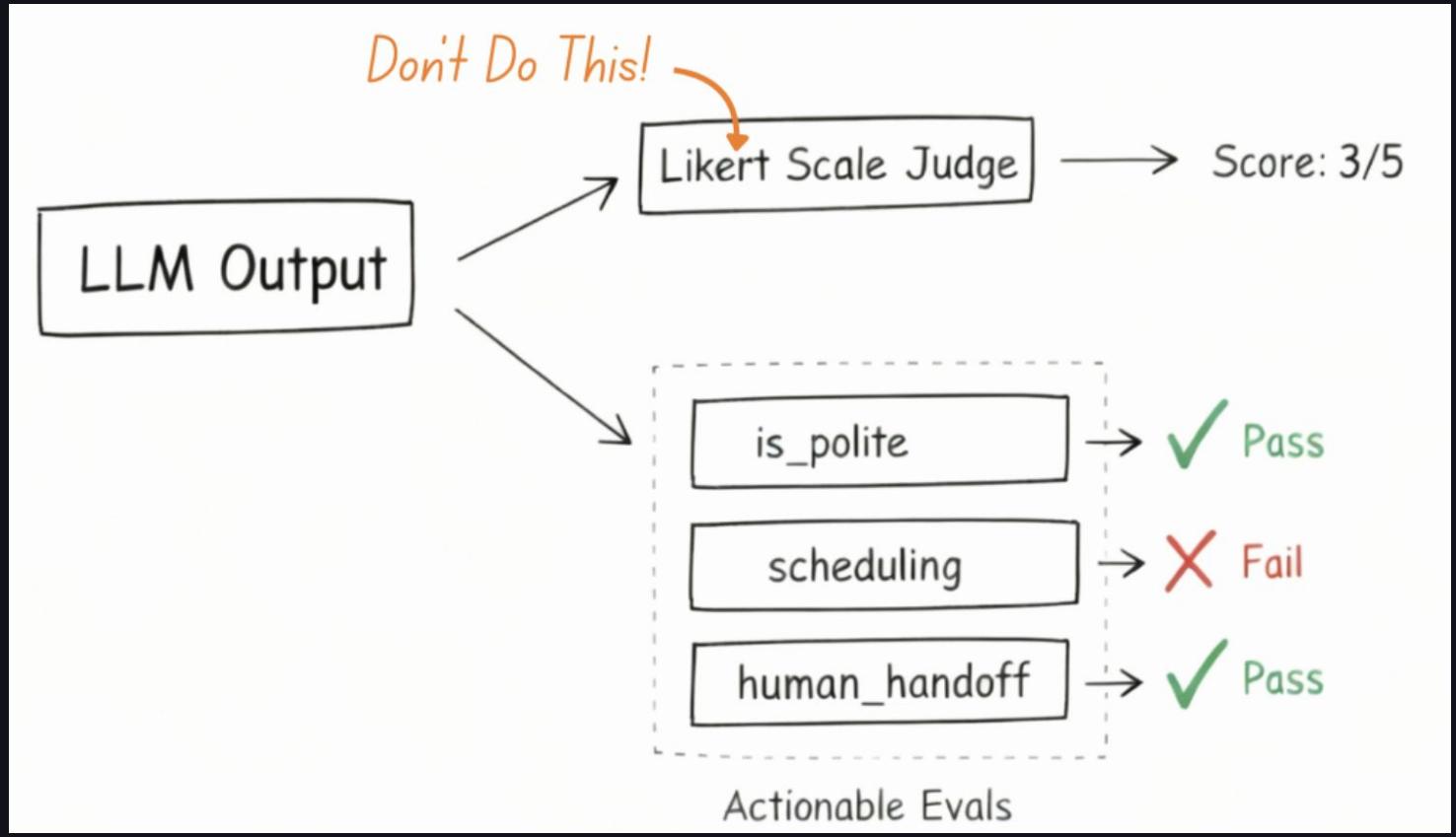
What a data scientist would do:

Reduce complexity

Make metrics actionable

Align with business outcomes

Try to use binary scores



Pitfall #4: Bad Data/Labels

Labeling

What most teams do:

Make it someone else’s problem

Labeling

What most teams do:

Make it someone else's problem

What a data scientist would do:

Insist that domain experts label

Be very skeptical

Look at the data!

Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences

Shreya Shankar
UC Berkeley
Berkeley, California, USA
shreyashankar@berkeley.edu

J.D. Zamfirescu-Pereira
UC Berkeley
Berkeley, California, USA
zamfi@berkeley.edu

Björn Hartmann
UC Berkeley
Berkeley, California, USA
bjoern@eecs.berkeley.edu

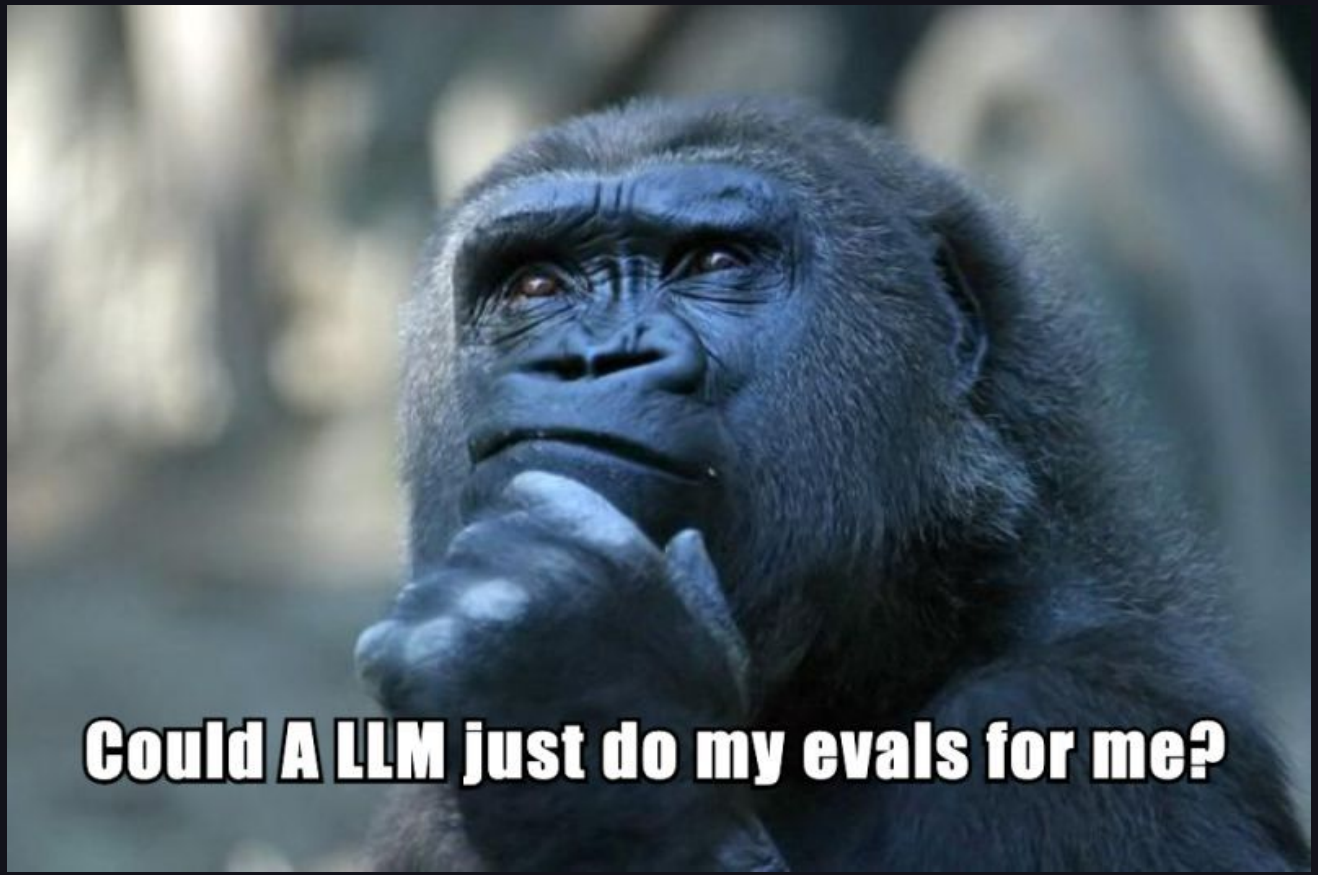
Aditya G. Parameswaran
UC Berkeley
Berkeley, California, USA
adityagp@berkeley.edu

Ian Arawjo
Université de Montréal
Montréal, Québec, Canada
ian.arawjo@umontreal.ca

the subjectivity and iterative process of alignment. In particular, we identify a phenomenon we dub *criteria drift*: users need criteria to grade outputs, but grading outputs helps users define criteria. What is more, some criteria appears *dependent* on the specific LLM outputs observed (rather than independent criteria that can be defined *a priori*), raising serious questions for approaches that assume the independence of evaluation from observation of model outputs.

Forcing your PMs to look at traces instead of reporting "Coherence Score"

Pitfall #5: Automating Too Much



Not Yet

Other Pitfalls

1. Misusing similarity scores
2. Asking the judge "is this helpful?"
3. Making annotators read raw JSON
4. Reporting uncalibrated scores without confidence intervals
5. Ignoring data & criteria drift
6. Overfitting judges to data
7. Not sampling data effectively
8. Dashboards with low signal
9. Logging traces and saying “that’s evals”

The Harness Is Data Science

1. **Error analysis** — Read outputs, find patterns (EDA)
2. **Metric design** — evals aligned to what matters
3. **Validation** — Prove evaluators match human judgment (Model evaluation)
4. **Test data** — Generate diverse inputs (Experimental design)
5. **Monitoring** — Detect drift (Production ML)
6. **Iteration** — Measure, improve, experiment (The scientific method)

Why Python

The harness runs on your tools

hamelsmu/evals-skills

Eval Skills for AI Coding Agents

Skills that guide AI coding agents to help you build LLM evaluations.

These skills guard against common mistakes I've seen helping 50+ companies and teaching students in our AI Evals course. If you're new to evals, see questions.md for free resources on the fundamentals.

New to Evals? Start Here

If you are new to evals, start with the `eval-audit` skill. Give your coding agent these instructions:

> Install the eval skills plugin from https://github.com/hamelsmu/evals-skills, then run /evals-skills:eval-audit on my eval pipeline. Investigate each diagnostic area using a separate subagent in parallel, then synthesize the findings into a single report. Use other skills in the plugin as recommended by the audit.

The audit isn't a complete solution, but it will catch common problems we've seen in evals. It will also recommend other skills to use to fix the problems.

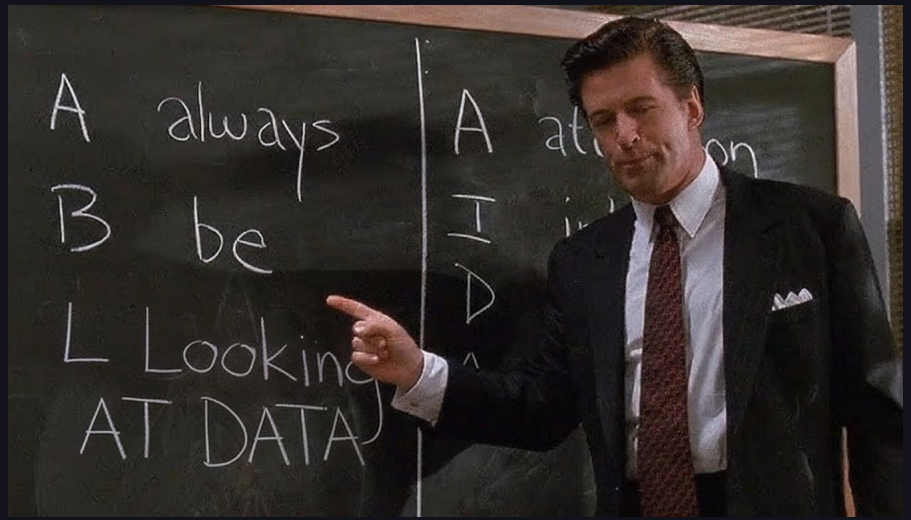
Installation

In Claude Code, run these two commands:

```
# Step 1: Register the plugin repository
/plugin marketplace add hamelsmu/evals-skills

# Step 2: Install the plugin
/plugin install evals-skills@hamelsmu-evals-skills
```

Most Importantly



Find the memes @ hamel.dev