Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Evaluating AI Coding Assistants and LLM Apps

Class Duration

7 hours of live training delivered over 1-2 days to accommodate your scheduling needs.

Student Prerequisites

- Professional software development experience
- Basic familiarity with testing concepts (unit tests, CI pipelines)

Target Audience

Software engineers, ML engineers, and engineering managers who need to measure the actual quality of AI-assisted output - not just anecdotes - and build systematic evals into their development process. Equally relevant for teams evaluating whether to adopt or switch AI tools, and for developers building LLM-powered features who need to prevent quality regressions. Teams operating those features in production can pair this with *LLM Observability and Cost Engineering*.

Description

This course treats LLM evaluation as a first-class engineering discipline. We cover the design and implementation of eval harnesses for both AI coding assistant workflows and LLM-powered applications: golden dataset construction, automated scoring (LLM-as-judge, unit test pass rate, assertion-based), regression detection in CI, and human evaluation design. Participants build a working eval pipeline for at least one realistic scenario during the labs.

Learning Outcomes

- Describe the dimensions of LLM output quality relevant to code generation and application responses.
- Build a golden dataset for a target task with appropriate input/output pairs and labeling criteria.

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Implement LLM-as-judge scoring with calibration and inter-rater reliability assessment.
- Write assertion-based evals for structured output and functional correctness.
- Integrate an eval suite into a CI pipeline to catch regressions on model or prompt changes.
- Analyze eval results to identify systematic failure modes.
- Design a human evaluation study for tasks that resist automated scoring.
- Build and run evals with DeepEval, including pytest integration and LLM-as-a-Judge metrics, alongside unit tests in CI.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Python 3.12+, API key for at least one frontier model (for LLM-as-judge labs), and Git.

Training Topics

Evaluation Fundamentals

- Why anecdotal assessment fails at scale
- Dimensions of quality: correctness, helpfulness, safety, style
- Automated vs. human evaluation tradeoffs
- The eval pyramid: fast unit evals to slow human evals

Golden Dataset Construction

- Input selection and diversity criteria
- Output labeling: reference answers, rubrics, and criteria
- Labeling tools and workflows
- Dataset versioning and maintenance

LLM-as-Judge Scoring

- Designing LLM judge prompts
- Calibration against human labels
- Pairwise vs. absolute scoring

- Detecting and mitigating judge biases

Assertion-Based Evals

- Exact match, regex, and substring assertions
- Unit test pass rate as an eval metric
- JSON Schema validation for structured outputs
- Combining assertions for composite scores

Eval Harness Implementation

- Eval runner architecture: dataset → model → scorer → report
- Parallelizing eval runs for speed
- Caching model responses during development
- Eval framework options: DeepEval, Braintrust, Promptfoo, and custom harnesses

CI Integration for Regression Detection

- Running evals on prompt or model changes
- Setting pass/fail thresholds
- Delta reporting: regression vs. improvement
- Cost budget for CI evals

Evaluating Coding Assistants Specifically

- Task-based eval: acceptance rate, edit distance, correctness
- Measuring impact on cycle time and review pass rate
- Privacy and data handling for real-codebase evals

Human Evaluation Design

- When automated evals are insufficient
- Study design: sample size, evaluator diversity, instructions
- Inter-rater reliability measurement
- Efficient human-in-the-loop eval workflows

Evals with DeepEval

- DeepEval as a pytest-native evaluation framework
- Metrics: answer relevancy, faithfulness, and tool correctness
- The LLM-as-a-Judge pattern without a hand-labeled dataset
- Running evals alongside unit tests in a CI/CD pipeline

