Galileo Take-home Exercise

Task

Your task is to build any kind of a multi-agentic application using your choice of AI frameworks, and hook in an open source evaluation or free evaluation tool of your choice.

Note: Do not use Galileo's free version for this exercise.

Goal
Your goal is to improve aspects of your application using evals and metrics.
Along the way, take notes on the evaluation and observability process and highlight what an ideal evals and observability platform should look like.

What to Build

- Build a Gen AI (Agentic) application.

- Use Python or TypeScript as the language of choice.

- The app must include at least 2 steps (can be either 2 tools, 2 agents) apart from an LLM call.
  (*e.g. a data lookup + an API lookup (designed as a tool) followed by an LLM call.)*

- Pick a real-world use case you care about — ideally something where LLMs are likely to fail or require reasoning across steps.

Integration with Evaluation SDK

- Log your agent input and output, any metadata to your chosen Evals tool and view it
  (*either in the terminal if it's just a library, or on the eval tool's UI if it's not just a library/SDK*)

- Run a few evaluations and/or experiments - identify issues and improve your system. Take note of issues you've identified.

What to Share

Create a lightweight write-up or report covering (but not limited to) the following:

1. Your Workflow & Developer journey

- What does your app do?

- Your tech stack -- what libraries, data stores and frameworks you used, and why?

- Detail out how you built your application i.e. your journey to getting to a V1 of the app.

- What are the production risks you foresee associated with an agent like yours

2. Evaluation Report

- What kinds of insights or breakdowns did you build using the evals framework

- What Eval workflows helped you the most
  (e.g. doing experimental sweeps was useful in identifying which LLM to use)

3. What makes a "great" Evaluation and Observability tool?

- Jot down (in your own words), your thoughts on what constitutes a solid evals tool. What should the tool enable a developer to do, what should the tool not focus on, what features in an evals tool would really empower the developer