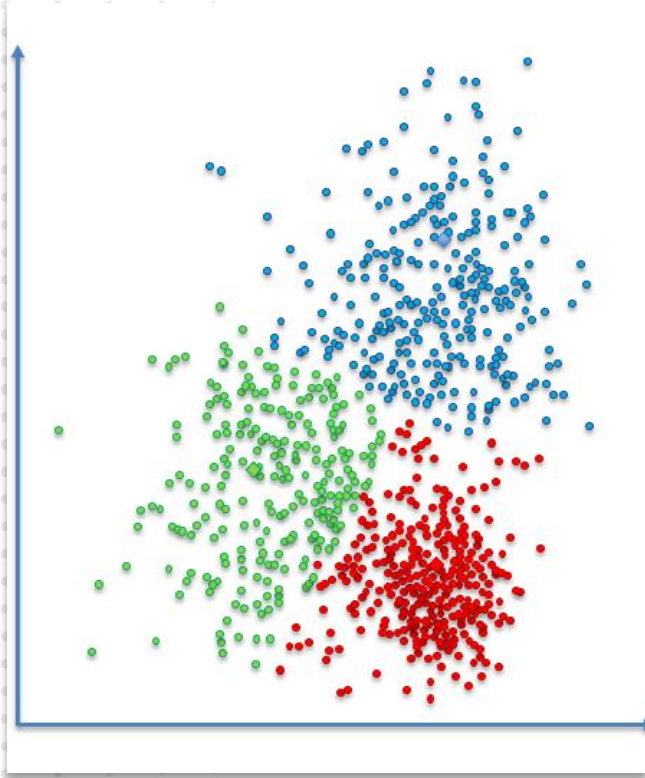


# Lesson #6: LLM Monitoring and Observability

Objectives:

- Monitoring and Observability Overview
- LangSmith



# Monitoring and Observability Platforms

## Models and Transformers

Llama

OpenAI,  
Claude, etc

DeepSeek

## Development Environments

Colab

VS Code

Cursor /  
Windsurf

## Local LLM Management Platforms

Ollama

LM Studio

Jan

## RAG & Agentic Frameworks

LlamaIndex

LangGraph

Haystack

## Agent Orchestration Platforms

LangGraph

AutoGen

CrewAI

## Agentic Protocols

APIs

MCP

A2A

## Cloud Development Platforms

AWS Bedrock

Azure  
AI Studio

Google  
Vertex AI

## Monitoring and Observability

LangSmith

Weights &  
Biases

Helicone

# Seeing Inside the Black Box: Monitoring AI Agents in Production

Your AI agent just made 47 API calls, reasoned through 12 steps, and gave a wrong answer.  
How do you debug that????

Quick statistic: "73% of AI projects fail due to lack of proper monitoring and observability"

# The AI Agent Monitoring Challenge

- **Complex reasoning chains:** Agents make multi-step decisions we can't see
- **Non-deterministic behavior:** Same input  $\neq$  same output
- **Tool orchestration:** Agents use multiple tools, APIs, and data sources
- **Emergent failures:** Issues arise from interaction patterns (e.g. workflows), not individual components

# What Makes AI Agent Observability Different

## **Reasoning is not transparent:**

- Need to see the "thought process" not just inputs/outputs
- Understanding decision trees and branching logic (why was a certain tool selected?)
- Tracking confidence levels and uncertainty

## **AI uses multi-modal interactions**

## **Behaviour is dynamic (agents adapt based on context)**


# LangSmith - Purpose-Built for AI Observability

## What is LangSmith?

- Observability platform specifically designed for LLM applications and AI agents
- Created by LangChain team - understands the AI agent ecosystem
- End-to-end visibility from user query to final response
  - Allows tracing of every step of the agentic workflow

# LangSmith – How it Works

- LangChain has a built-in callback architecture that all its components use.
- Whenever something meaningful happens (e.g. like calling an LLM, executing a tool, running a chain) it triggers callback events.
- The Callback Manager is like an event hub. It sends those events to all registered handlers.



```
LANGSMITH_TRACING=true  
LANGSMITH_ENDPOINT="https://api.smith.langchain.com"  
LANGSMITH_API_KEY "<your-api-key>"  
LANGSMITH_PROJECT="Climate Research"
```

- Also works without a LangChain framework, using “wrappers”

# LangSmith in Action

**DEMO**

The screenshot displays the LangSmith web interface. On the left is a sidebar with navigation links: Home, Observability, Tracing Projects (1), Monitoring (0), Evaluation, Datasets & Experiments (0), Annotation Queues (0), Prompt Engineering, Prompts (0), Playground, LangGraph Platform, and Deployments (0). At the bottom of the sidebar are links for Settings, Documentation, What's New, Contact Sales, and Invitations.

The main area shows a tracing project named 'Climate Research' with the description 'A test to see how Langchain wor...'. It has tabs for Runs, Threads, Alerts, and Setup. A filter for '1 filter' and a time range of 'Last 1 hour' are visible. A list of runs is shown, with 'AgentExecutor' selected. The 'AgentExecutor' run is highlighted in green, indicating success, and shows a duration of 47.70s and 2,081 tokens.

The 'TRACE' section shows a waterfall view of the execution steps:

- RunnableSequence (1.35s)
- RunnableAssign... (0.00s)
- RunnablePara... (0.00s)
- RunnableLamb... (0.00s)
- PromptTem... (0.00s)
- OllamaLLM (1.33s)
- ReActSingleInputOut... (0.00s)
- LocalDocQA (1.26s)
- RetrievalQA (1.26s)
- VectorStoreRetrie... (0.07s)
- StuffDocumen... (1.19s)
- LLMChain (1.19s)
- OllamaLLM (1.19s)
- RunnableSequence (1.31s)
- RunnableAssign... (0.00s)
- RunnablePara... (0.00s)
- RunnableLamb... (0.00s)
- PromptTem... (0.00s)

The 'AgentExecutor' details panel on the right shows the 'Run' tab selected. It includes a 'Compare' button and icons for refresh, add, share, and edit. The 'Input' section shows the prompt: 'What causes the most CO2 emissions?'. The 'Output' section shows the response: 'Agent stopped due to iteration limit or time limit.'.

Metadata on the right includes:

- START TIME: 06/24/2025, 08:44:04 AM
- END TIME: 06/24/2025, 08:44:52 AM
- TIME TO FIRST TOKEN: 637 ms
- STATUS: Success
- TOTAL TOKENS: 2,081 tokens
- LATENCY: 47.70s
- TYPE: Chain