SCORED.TOOLS · ROUNDUP

Best Open Source ML Monitoring & Observability Tools 2026

Eight open source ML monitoring and observability tools ranked by what actually catches drift, broken traces, and silent regressions in production.

ML monitoring in 2026 splits cleanly into two problems that used to be one. Classical ML still needs drift detection, feature statistics, and post-deployment performance tracking on tabular data. LLM systems need trace-level visibility into prompts, tool calls, retrieval hits, latency, cost, and eval scores — problems that most 2023-era monitoring stacks were never designed to solve. The tools below cover both sides, ranked by what we'd actually install on a fresh project this quarter.

Rankings are based on running these in production against real workloads: LLM apps serving thousands of daily requests, classical models scoring millions of rows a day, and RAG pipelines where a broken embedding column silently tanks recall. We penalized tools that look great in a demo notebook and fall apart the moment you try to self-host them at scale.

1. Arize Phoenix — Score: 9.4/10

Arize Phoenix is the LLM observability tool we reach for first in 2026. It's built on OpenTelemetry from the ground up, ships as a pip install for local dev and a Docker image for self-hosted prod, and instruments LangChain, LlamaIndex, the OpenAI SDK, Anthropic, LiteLLM, and Vercel AI SDK out of the box. You get trace waterfalls, span-level cost/latency, prompt playback, retrieval evaluation, and an eval harness that runs LLM-as-judge or code-based checks over your traces. The UI is fast, the query language is sensible, and the OTel foundation means you can pipe the same traces into Grafana Tempo or Jaeger if you outgrow the built-in UI.

Best for

Teams building LLM or RAG applications who want production-grade tracing without renting a SaaS.

Pricing

Free, Elastic License 2.0. Fully self-hostable; Arize sells a managed cloud version if you don't want to operate it.

2. Langfuse — Score: 9.2/10

Langfuse is the other serious contender for LLM observability and honestly the choice comes down to team taste. Where Phoenix is OTel-native and eval-heavy, Langfuse is opinionated about the full LLMOps loop: tracing, prompt management with versioning, datasets, evals, and a scoring UI that non-engineers will actually use. Self-hosting is a docker-compose away and the schema is stable enough that we've run the same Postgres through three Langfuse upgrades without a migration hiccup. The SDKs (Python, JS, and a decorator-based API) are the least intrusive of any tracer we've tried.

Best for

Product teams that want engineers, PMs, and domain experts looking at the same traces and scoring the same outputs.

Pricing

Free, MIT license, self-hosted. Managed cloud starts at $0 with usage-based pricing above the free tier.

3. MLflow — Score: 9.0/10

Mlflow is still the gravitational center of classical MLOps and in 2026 it has grown a credible LLM tracing story too. The experiment tracking, model registry, and deployment scaffolding are boring in the best possible way — they just work, they scale, and every ML engineer already knows the API. MLflow 3.x added `mlflow.trace` for LLM spans, prompt versioning, and evaluation harnesses that plug into the same UI as your classical runs. The single-server backend can be swapped for Postgres + S3 the moment you outgrow the SQLite default, and the OSS project keeps shipping without the enterprise-only gates that plague some competitors.

Best for

Any team running both classical ML and LLM workloads that wants one artifact store, one registry, and one UI for both.

Pricing

Free, Apache 2.0. Databricks sells a managed version integrated with their platform.

4. Evidently AI — Score: 8.8/10

Evidently Ai is the fastest way to get drift detection, data quality reports, and model performance dashboards on a classical ML pipeline. Point it at a reference dataset and a current batch, and it

generates an HTML report or JSON payload covering feature drift (KS, Wasserstein, PSI), target drift, missing value patterns, and prediction quality — all with sensible defaults you can override. In 2026 the LLM eval side has also matured: hallucination detection, response quality scoring, and RAG-specific metrics ship in the same package. It runs equally well as a one-shot report in a notebook, a scheduled Airflow job, or a live monitoring service via their `evidently ui` dashboard.

Best for

Data science teams that need drift and quality monitoring on tabular models today, not next quarter.

Pricing

Free, Apache 2.0. Evidently Cloud offers hosted monitoring on a usage-based tier.

5. Prometheus + Grafana — Score: 8.7/10

Not ML-specific, but every serious ML platform we've seen in production still runs on Prometheus for metrics and Grafana for dashboards. GPU utilization, model server latency, request rates, error budgets, queue depth — this is where they live. In 2026 the ecosystem around it (Grafana Tempo for traces, Loki for logs, Mimir for long-term metric storage, Alloy as the unified agent) makes it a legitimate one-stop observability stack that ML workloads can share with the rest of the infra. Pair it with Phoenix or Langfuse for the LLM-specific views and you've covered the full pyramid.

Best for

Teams that want infrastructure-level observability for model servers, GPU clusters, and inference APIs.

Pricing

Free, Apache 2.0 (Prometheus) and AGPLv3 (Grafana OSS). Grafana Cloud has a generous free tier.

6. WhyLogs — Score: 8.3/10

WhyLogs is the open source data profiling library from WhyLabs and it fills a niche the heavier tools miss: statistical fingerprints of your data that are small enough to log continuously. A profile of a million-row batch is a few kilobytes, mergeable across shards, and comparable across time windows without ever moving the raw data. That makes it the right tool for privacy-sensitive pipelines where you can't ship raw features to a monitoring service, and for high-volume streams where sampling would miss the tail. The visualization story is thinner than Evidently's — you either use the WhyLabs SaaS or roll your own dashboard on the profiles.

Best for

Pipelines processing millions of rows a day where continuous profiling has to be cheap and privacy-preserving.

Pricing

Free, Apache 2.0. WhyLabs sells a hosted observability platform that consumes the profiles.

7. NannyML — Score: 8.1/10

NannyML solves a specific and painful problem: estimating model performance in production when ground-truth labels are delayed or missing entirely. Its CBPE (Confidence-Based Performance Estimation) and DLE (Direct Loss Estimation) methods actually work on the fraud, churn, and recommendation problems where you don't know if a prediction was right for weeks. The library is small, the docs are excellent, and the visualization output is honest about uncertainty. It's not a full monitoring platform — you still want Evidently or Grafana on top — but for the label-delay problem there's nothing else in the OSS world that comes close.

Best for

Classical ML teams with delayed-label problems (fraud, churn, credit, recommendations) who need to know if the model degraded before the labels arrive.

Pricing

Free, Apache 2.0. NannyML Cloud is the managed offering.

8. Deepchecks — Score: 7.9/10

Deepchecks is a validation-suite library in the pytest tradition: you write a suite of checks (feature drift, label drift, prediction distribution, class imbalance, integrity issues) and run it against a dataset and model on a schedule or in CI. It overlaps with Evidently, but the mental model is different — Evidently generates reports for humans to read, Deepchecks generates pass/fail results for pipelines to act on. Their LLM eval product (Deepchecks LLM Evaluation) is newer and less battle-tested than Phoenix or Langfuse but is worth watching if you like the suite-based approach.

Best for

Teams that want ML validation to feel like a test suite that gates a deployment.

Pricing

Free, AGPLv3 (with a commercial license option). Deepchecks Cloud handles hosted runs.

Comparison Table

Tool	Score	Primary Use	Self-Host	License
Arize Phoenix	9.4	LLM tracing & eval	Yes	Elastic v2
Langfuse	9.2	LLM ops loop	Yes	MIT
MLflow	9.0	Experiment tracking + registry	Yes	Apache 2.0
Evidently AI	8.8	Drift & data quality	Yes	Apache 2.0
Prometheus + Grafana	8.7	Infra metrics & dashboards	Yes	Apache 2.0 / AGPLv3
WhyLogs	8.3	Lightweight data profiling	Yes	Apache 2.0
NannyML	8.1	Performance estimation w/o labels	Yes	Apache 2.0
Deepchecks	7.9	Validation test suites	Yes	AGPLv3

Final Picks

If you're building an LLM or RAG application: start with Arize Phoenix or Langfuse. Both are excellent; pick Phoenix if you value the OpenTelemetry foundation and the eval library, Langfuse if you want prompt management and a scoring UI built in.

If you're running classical ML in production: Mlflow for tracking and the registry, Evidently Ai for drift and data quality reports, and NannyML if you have delayed labels. This is the boring, correct answer.

If you're doing both: Mlflow as the artifact backbone, Phoenix or Langfuse for LLM traces, Evidently Ai for tabular drift, and Prometheus + Grafana underneath the whole thing for infra. The pieces compose better than any single "unified" platform we've tried — and every piece is free to run on your own hardware.

The pattern that keeps holding up in 2026: pick the smallest, most focused tool for each layer, wire them together with OpenTelemetry where possible, and resist the pull toward a single vendor promising to do all of it. The single-vendor pitch always sounds cleaner in the demo and always leaves gaps once real traffic hits.

Generated 2026-07-29 · Read the live article at scored-tools.netlify.app/blog/best-open-source-ml-monitoring-and-observability-tools-2026