Data Ingestion & Preparation

Feature extraction, preprocessing, and dataset versioning form the foundation of reproducible ML systems. Proper tracking at this stage ensures complete traceability of data transformations.

MLflow Tracking

Logs dataset versions including paths, hashes, and timestamps. Records all preprocessing parameters such as scaling methods, encoding schemes, and feature transformations. Captures comprehensive metadata associated with training datasets.

Purpose

Ensures complete reproducibility and traceability of data changes throughout the ML lifecycle. Enables teams to precisely recreate any training run by tracking every data modification.

Model Training

MLflow Tracking

Centralised experiment management platform that logs all training runs, hyperparameters, and performance metrics including accuracy, loss, and F1 scores. Enables systematic comparison across experiment runs.

MLflow Model Registry

Version control system for trained models. Registers model artefacts, tracks version history, and manages transitions from staging environments to production deployment.

100%

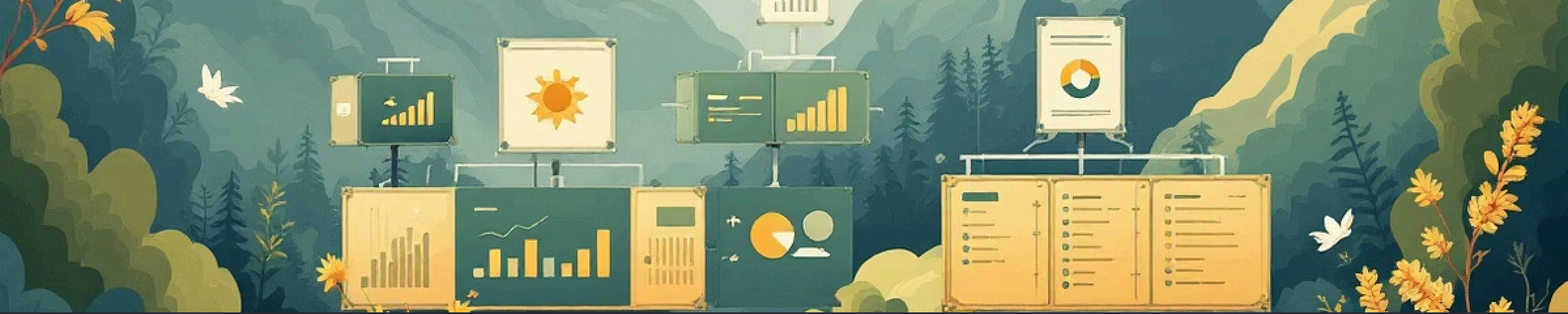
Experiment Visibility

Complete audit trail of all training runs

3

Key Stages

Development, staging, production



Model Packaging & Deployment

Containerisation and serving infrastructure ensure models transition smoothly from development to production with full operational observability.

01

Package with MLflow Models

Create reproducible environments using Conda or Docker. Serve models via `mlflow models serve` with tracked deployment configurations.

02

Monitor with Prometheus

Track service health metrics including CPU/RAM utilisation, request counts, latency distributions (p50/p95/p99), and error rates.

Purpose: Guarantees operational health of deployed ML services with comprehensive system-level monitoring and alerting capabilities.

Model Inference Monitoring

prediction_latency_seconds

Measures response time for each prediction request

prediction_requests_total

Tracks total volume of inference requests

prediction_errors_total

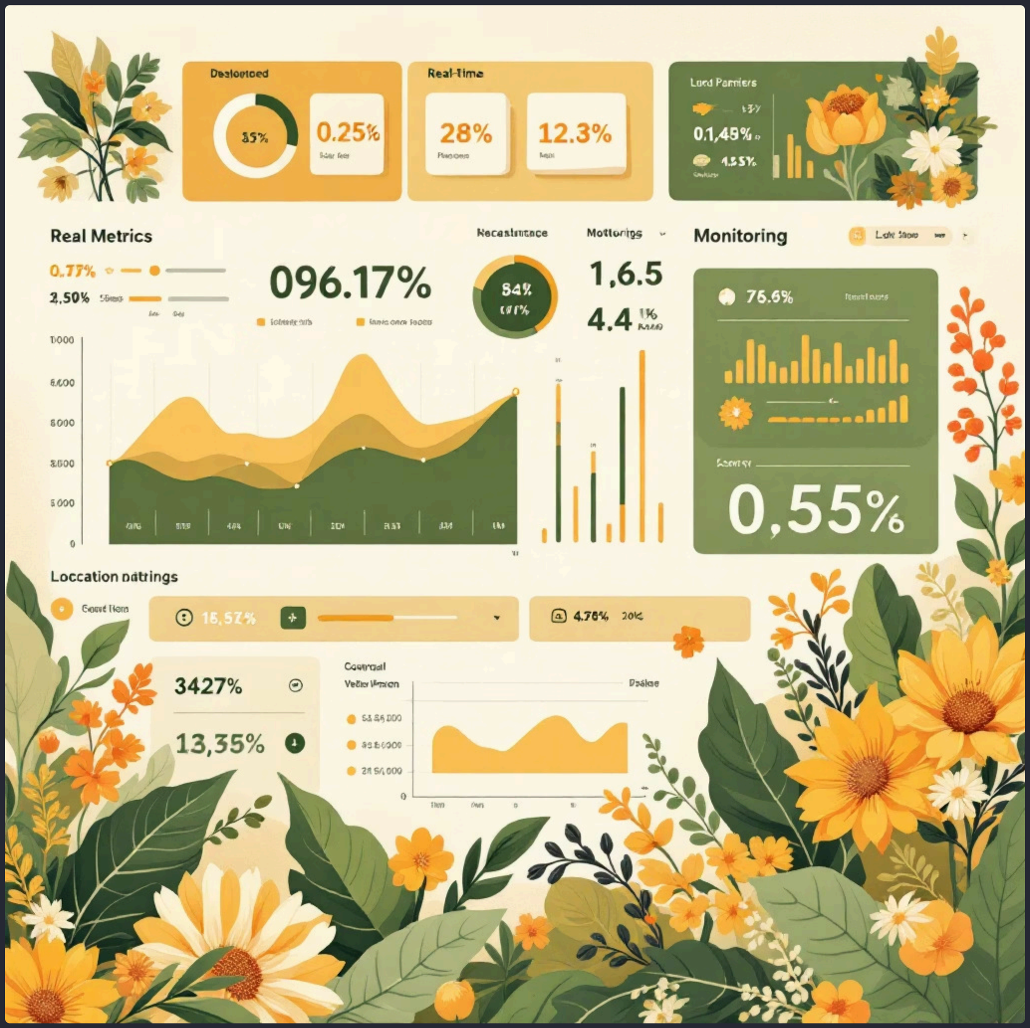
Counts failed predictions and errors

Model Version Metric

Identifies which model version served each request

Real-Time Observability

Instrument inference code with custom Prometheus metrics to observe prediction behaviour as it happens. Optionally include data drift detection for early warning signals.



Model Performance Monitoring

Prometheus Collection

- Model drift signals and indicators
- Confidence score distributions
- Output skew detection metrics
- Model load patterns and throughput trends

Grafana Visualisation

- Live drift trend analysis
- Latency heatmaps and distribution
- System health overview dashboards
- Prediction error tracking over time

Early Detection: Continuous monitoring of time-series metrics enables teams to identify performance degradation and drift before they impact business outcomes. This proactive approach minimises downtime and maintains model quality.

Pipeline Monitoring

End-to-end observability: ETL → Training → Deployment → Inference

1 ETL Monitoring

Job duration, data quality checks, pipeline triggers

2 Training Metrics

Job duration, retraining triggers, success/failure counters

3 Deployment Health

Service availability, rollout status, version tracking

4 Inference Patterns

Request volume, latency trends, error rates

Prometheus

Monitors all pipeline components via exporters and custom metrics instrumentation

Grafana Dashboards

Visualises execution timelines, failure alerts, training metrics over last N runs, and trend lines for drift or decay

Model Governance & Lifecycle Management

Complete version control, lineage tracking, and auditability ensure regulatory compliance and operational excellence throughout the model lifecycle.

Dataset

Source data versioning

Code

Training scripts & configs

Training

Experiment tracking

Deployment

Production release

MLflow Model Registry

Tracks complete model lineage from dataset through code to deployment. Manages approvals across staging and production environments. Enables rapid rollbacks when needed. Maintains comprehensive model metadata for full governance and audit trail capabilities.