140509_44.md â€” Comprehensive AI Model Monitoring & Observability Platform

> **Theme:** AI Observability & FinOps for AI
> **Mission:** End-to-end observability of models, data, and infrastructure with drift detection, explainability tracking, alerting, and A/B comparisons.

README (Problem Statement)

Summary: Create a comprehensive monitoring platform that tracks AI model performance, data drift, and operational metrics across the ML lifecycle.
Problem Statement: Production AI models require continuous monitoring to ensure performance, detect issues, and maintain reliability. The platform must monitor predictions, data quality, performance metrics, and system health, detecting drift, degradation, and operational issues while offering insights for improvement.

Steps:
- Real-time model monitoring
- Drift detection
- Explainability tracking
- System metrics integration
- Automated alerts
- Model comparison & A/B testing

Suggested Data: prediction logs, ground truth labels, baselines, resource metrics, model configs.

---

1) Vision, Scope, KPIs

Vision: Provide a unified observability layer for AI in production.
Scope:
- v1: prediction logging, performance metrics, drift alerts.
- v2: explainability, A/B testing, real-time dashboards.
- v3: automated remediation suggestions.

KPIs:
- Drift detection recall â‰¥0.9 @ FPR â‰¤0.05
- Latency overhead â‰¤5%
- Coverage: 100% deployed models monitored

---

2) Personas & User Stories

- **Data Scientist:** â€œI need drift alerts and performance metrics.â€
- **ML Engineer:** â€œI want dashboards with latency and error metrics.â€
- **Product Manager:** â€œI want A/B comparisons for business KPIs.â€
- **Compliance Officer:** â€œI need audit logs of model behavior.â€

User Stories:
- US-01: â€œAs a DS, I want to be notified when input distribution drifts.â€
- US-06: â€œAs an MLE, I want to track latency spikes.â€
- US-10: â€œAs a PM, I want to compare v1 vs v2 models.â€

---

3) PRD

Capabilities:
1. **Prediction Logging:** inputs, outputs, confidences.
2. **Ground Truth Join:** delayed labels for accuracy metrics.
3. **Drift Detection:** PSI, KS tests, embedding drift.

4. **Performance Monitoring:** latency, throughput, error rates.
5. **Explainability Tracking:** SHAP value distributions.
6. **Alerting:** rules, thresholds, anomaly detection.
7. **Model Comparison:** side-by-side metrics, A/B dashboards.

---

4) FRD

- **Agents/SDKs:** embed in inference services.
- **Data Pipeline:** Kafka/Flink ingestion to feature store.
- **Metrics Store:** TSDB (Prometheus/Influx).
- **Drift Detection:** PSI for categorical, KS/ADWIN for numeric, embedding distance.
- **Explainability:** SHAP on sample batch; log distributions.
- **UI:** Grafana dashboards, alerts via Slack/Email.
- **A/B Testing:** bucket requests, compare metrics.

---

5) NFRD

- **Scale:** 10k predictions/s per model.
- **Latency:** â‰¤5% overhead.
- **Availability:** 99.9%.
- **Security:** encrypted logs, PII redaction.
- **Compliance:** GDPR/CCPA.

---

6) Architecture (Logical)

```
[Inference Services] -> [SDK/Agent] -> [Kafka Stream] -> [Processor]
                                  |-> [Metrics Store]
                                  |-> [Drift Detector]
                                  |-> [Explainability Tracker]
                                  |-> [Alert Engine]
                                  |-> [Dashboards/UI]
```

---

7) HLD

- **SDK:** Python/Java wrappers.
- **Processor:** Flink jobs for aggregation.
- **Metrics Store:** Prometheus/Elastic.
- **Drift:** ADWIN for online; PSI nightly batch.
- **Explainability:** periodic SHAP sampling.
- **UI:** Grafana/Kibana.

---

8) LLD Examples

PSI Formula:
PSI = Î£ (actual% - expected%) * ln(actual% / expected%)

Alert Rule:
- If PSI > 0.2 on any key feature â†’ trigger â€œmajor drift.â€

---

9) Pseudocode

```
log_prediction(x, yhat)
if label arrives:
  join(x, y)
  update_accuracy(yhat, y)
if drift_detector.detect(x) > threshold:
  alert("Drift detected")
```

---

10) Data & Evaluation

- **Data:** historical logs, synthetic drifts, benchmarks.
- **Metrics:** drift recall/FPR, alert SLA compliance.
- **Validation:** replay logs with injected drift.

---

11) Security & Governance

- Logs anonymized, PII masked.
- Immutable audit trail.
- Role-based access.

---

12) Observability & Cost

- Metrics: EPS, latency overhead, drift events, false alerts.
- Tracing: OpenTelemetry.
- Cost: autoscaling processors, tiered storage.

---

13) Roadmap

- **M1 (4w):** Logging + metrics.
- **M2 (8w):** Drift + explainability.
- **M3 (12w):** A/B testing.
- **M4 (16w):** Auto-remediation suggestions.

---

14) Risks & Mitigations

- **False drift alerts:** calibrate thresholds, ensembles.
- **Latency overhead:** async logging.
- **Data privacy:** on-prem storage, masking.