Assurer la supervision et la fiabilité des modèles d'IA

Objectif : Garantir la robustesse, la performance, la sécurité et la conformité des modèles déployés (ML classiques & LLM), du développement à l'exploitation.

---

1) Gouvernance & périmètre

- **Portée** : tous les modèles en production (batch, streaming, online inference, LLMs/RAG, AutoML).
- **Documents obligatoires** : *Model Card*, *Data Card*, *Risk Assessment* (dont biais, sécurité, conformité), *Runbook d'incident*, *Plan de retraining*, *Plan de tests*.
- **RACI type** :
  - *Product Owner* (A) : objectifs métier, seuils d'acceptation, priorisation.
  - *ML Engineer* (R) : déploiement, observabilité, CI/CD, infra, retraining.
  - *Data Scientist* (R) : performances, features, dérive, explicabilité.
  - *Security/Privacy* (C) : menaces, secrets, PII, DPIA.
  - *Compliance/Legal* (C) : régulations (GDPR/IA Act), conservation.
  - *Ops/SRE* (R) : alerting, SLO/SLA, escalade.

---

2) KPI de performance (définition & mise en œuvre)

But : Mesurer la qualité technique et métier du modèle, ainsi que l'opérabilité.

2.1 KPI techniques (exemples et formules)

- **Précision (Accuracy)**
- **AUC/ROC**, **PR-AUC** : utiles pour classes déséquilibrées.
- **Temps de réponse (p50/p95/p99)** : latence en ms par endpoint.
- **Taux d'erreur système** : 5xx, timeouts, OOM.
- **Disponibilité** : % de réussite des inférences sur la période.
- **Coût par requête** : $/1K inférences ou tokens (LLM).

2.2 KPI métier (à co-définir avec PO)

- **Taux d'erreur métier** : % de prédictions invalidées par les règles métier (ex : faux positifs critiques).
- **Uplift / ROI** : valeur générée vs baseline (AB test).
- **Taux d'escalade humaine** (HITL) : part nécessitant revue.
- **Qualité perçue** : score NPS utilisateur, rating annotateurs.

2.3 KPI confiance & conformité

- **Biais** : écart de métriques entre sous-groupes (ex : ı

  |Precision_A - Precision_B|

|PrecisionA –PrecisionB |).

- **Robustesse** : dégradation sous perturbations contrôlées (tests d'adversarial/ ruido).
- **Explicabilité** : couverture d'explications générées (ex : % requêtes avec SHAP/ LIME), temps d'explication.

2.4 Seuils & SLO de référence (adapter au contexte)

- **SLO latence** : p95 < 300 ms (temps réel), p99 < 800 ms.
- **SLO disponibilité** : ≥ 99.5% mensuel.
- **SLO performance** : F1 ≥ 0.80, ou dégradation ≤ 2 pts vs baseline.
- **Budget coût** : $ X / 1K requêtes (alarme si +20%).

Livrables envisagé : un tableau de KPI par modèle avec *owner*, *formule*, *source de vérité*, *seuils*, *alertes*.

---

3) Monitoring continu (dérive, biais, sécurité)

But : Détecter tôt dégradations de données, de performance et menaces.

3.1 Données & prédictions

- **Dérive de données** :
  - *Feature drift* (KS test, PSI) & *label drift*.
  - *Data quality* : complétude, plage de valeurs, anomalies.
- **Drift de distribution de sorties** : shift de scores, calibration.
- **Performance live** : si labels retardés, utiliser *proxy labels* / *delayed labels* + backfill.

3.2 Biais & équité

- Suivi de métriques par sous-groupes protégés.
- Test d'invariance (ré-échantillonnage) et *fairness constraints*.

3.3 Sécurité & abus

- **LLM** : détection prompt injection, fuite de secrets, jailbreak, PII.
- **Général** : rate limiting, authN/Z, WAF, vérification d'input (schema, taille), sandbox.
- **Supply chain** : hashing modèles, signature, contrôle des dépendances.

3.4 Observabilité (outil-agnostique)

- **Logs** : inputs (hashés/masqués), features, outputs, latence, erreurs.
- **Métriques** : export Prometheus/OpenMetrics.
- **Traces** : OpenTelemetry (span par prédiction, charge, I/O).
- **Dashboards** : p50/p95/p99, QPS, erreurs, coût, drift.

Livrables :

- Table "santé" quotidienne (OK/Warning/Critique) par modèle.

- Alertes codées (PromQL/JSON) avec seuils et anti-flapping.

---

4) Plan de retraining automatique

But : Maintenir la performance en présence de dérive ou nouvelles données.

4.1 Déclencheurs

- **Fréquence fixe** : ex. hebdo/mensuelle.
- **Basé sur métriques** : PSI > 0.2, F1 < seuil, coût +20%.
- **Événements** : nouveaux produits, saisonnalité, campagne.

4.2 Pipeline (CI/CD MLOps)

i. **Ingestion** → validation (Great Expectations) → *feature store*.
ii. **Training** reproductible (seed, env, deps lockfile).
iii. **Évaluation** : offline + tests de robustesse & biais.
iv. **Validation** par gate (automatique + approbation humaine si risque élevé).
v. **Packaging** : versionnage (SemVer + hash), *model registry*.
vi. **Déploiement** : canary/blue-green, AB test, rollback automatique.
vii. **Post-déploiement** : shadow mode, perf watch 24–72h.

4.3 Gouvernance des versions

- **Model Registry** : stages *Staging* → *Production* → *Archived*.
- **Traçabilité** : code commit, données, hyperparams, artifacts, métriques, approbations.

Livrables : YAML de pipeline, règles de promotion, calendrier de retraining.

---

5) Mécanismes de fallback

But : Assurer la continuité de service et limiter l'impact client.

5.1 Stratégies

- **Baseline déterministe** : règles métier simples.
- **Modèle précédent** : rollback automatique si SLO violés.
- **Mode dégradé** : seuils plus stricts + escalade humaine.
- **Cache / réponses canoniques** : pour requêtes fréquentes.
- **LLM** :
  - *Fallback provider/model tiering* (ex : petit modèle local → modèle cloud plus capable en cas d'échec).
  - *Guardrails* avant et après génération (filtrage, citations obligatoires, limite de longueur).

5.2 Critères de bascule

- Taux d'erreur > seuil X sur 5 min.
- Latence p99 > Y ms pendant Z minutes.
- Score qualité < seuil (annotations online / votes).

5.3 Runbook d'incident

- Détection → triage (modèle vs infra vs données) → mitigations → RCA → action corrective → post-mortem (≤ 5 jours ouvrés).

---

6) Outils & bonnes pratiques

6.1 Monitoring ML

- **Prometheus/Grafana** : métriques systèmes/app, alerting.
- **Evidently AI** : drift, qualité, monitoring de performances.
- **WhyLabs/WhyLogs** : profils de données, dérive, anomalies.
- **OpenTelemetry** : traces, corrélation requêtes.

6.2 Tests de robustesse & stress

- **Tests unitaires/contrats** : validation d'API, schémas.
- **Perturbations données** : bruit, valeurs extrêmes, OOD.
- **Adversarial** : attaques (texte/image), jailbreak LLM.
- **Charge** : tests p95/p99, *soak tests*, *fault injection* (chaos).

6.3 Auditabilité & explicabilité

- **LIME/SHAP** : explications locales/globales, stabilité.
- **Explainable Boosting/GLM** : modèles intrinsèquement explicables.
- **Model/Prediction Logs** : pour audit ex-post, ré-exécution.

6.4 Sécurité & conformité

- **PII** : masquage, tokenisation, minimisation, chiffrage au repos & en transit.
- **Secrets** : KMS/Vault, rotation, *no secrets in code*.
- **Accès** : RBAC/ABAC, *least privilege*, journaux d'accès.
- **Conformité** : DPIA, consentement, conservation & purge, IA Act (évaluation de risque, transparence), droit d'accès & d'effacement.

---

7) Indicateurs de succès & reporting

7.1 Indicateurs proposés

- **% de modèles suivis en temps réel** : modèles avec dashboards + alertes actives / total.
- **Taux d'incidents liés à des erreurs IA** : incidents classés "ML/LLM" / incidents totaux.

- **MTTD/M** : temps moyen de détection et de mitigation.
- **MTTR-drift** : temps moyen correction/retraining après dérive.
- **Δ perf vs baseline** : variation mensuelle F1/AUC.
- **Couverture explicabilité** : % inférences avec explication disponible.
- **Coût par 1K inférences** & **dérive de coût**.

7.2 Cadence de revue

- **Hebdo** : santé opérationnelle, incidents, coût.
- **Mensuel** : dérive, biais, retrainings effectués.
- **Trimestriel** : audit sécurité & conformité, post-mortems.

---

8) Modèles de règles & exemples (à adapter à votre contexte)

8.1 Règles d'alerte :

- alert: HighErrorRate
expr: sum(rate(inference_errors_total[5m])) /
sum(rate(inference_requests_total[5m])) > 0.02
for: 10m
labels: {severity: critical}
annotations:
summary: ">2% d'erreurs d'inférence sur 10m"

- alert: HighP99Latency
expr: histogram_quantile(0.99, sum(rate(inference_latency_ms_bucket[5m]))
by (le)) > 800
for: 5m
labels: {severity: warning}

- alert: DataDriftPSI
expr: avg(psi_feature_*) > 0.2
for: 30m
labels: {severity: critical}

8.2 Seuils de fairness (exemple)

- |Précision_groupeA – Précision_groupeB| ≤ 5 pts.
- *Demographic parity* : ratio de taux positifs entre 0.8 et 1.25.

8.3 Politique de retraining (exemple)

schedule: monthly # + retraining immédiat si PSI>0.2 ou F1<-2pts
validation:

min_f1: 0.80
fairness:
demographic_parity_ratio: [0.8, 1.25]
robustness:
max_perf_drop_under_noise: 3pts
rollout:
strategy: canary
canary_traffic: 10%
rollback_on:
error_rate: ">1%"
p99_latency_ms: ">800"

---

9) Spécificités LLM (compléments)

- **Qualité** : *instruction-following*, factualité (pass@k), taux de refus justifiés, hallucinations.
- **Sécurité** : tests red-team (prompt injections, data exfiltration), filtres de sécurité en entrée/sortie.
- **RAG** :
  - Monitoring du *retrieval* : *recall@k*, *MMR diversity*, latence vectordb.
  - Vérification de citations (source grounding) & *answer consistency*.
- **Coût & tokens** : budgets/tokens, *prompt caching*, *output compression*.
- **Fallback LLM** : routing de requêtes (petit modèle local → grand modèle), timeouts, ré-essais idempotents.

---

10) Plan d'implémentation en 6 semaines (exemple)

- **S1** : Inventaire des modèles, KPI cibles, design observabilité, sécurité.
- **S2** : Instrumentation métriques (Prom/OTel), dashboards initiaux.
- **S3** : Intégration règles de dérive & biais.
- **S4** : Pipeline retraining (CI/CD, registry, tests, gates).
- **S5** : Fallbacks, canary/rollback, runbooks & exercices d'incident.
- **S6** : Revue conformité, audit sécurité, mise en production élargie.