

Cours : CI/CD pour Machine Learning et MLOps

1 CI/CD en ML : définition simple

- **CI (Continuous Integration)** : intégration continue
→ automatisation des tests, validation du code et des données, entraînement du modèle.
- **CD (Continuous Deployment / Delivery)** : déploiement continu
→ automatisation du déploiement du modèle en production, avec rollback et canary.

Exemple réel

Code push → Git → tests + entraînement → modèle validé → déploiement canary sur FastAPI → monitoring → retraining automatique si drift.

2 Pourquoi CI/CD est essentiel en ML

Reproductibilité (pilier)

- Même code + mêmes données + même environnement → même modèle.
- Exemple : relancer le pipeline 1 an après et obtenir le même modèle.
- Métaphore : refaire exactement la même recette de gâteau.

Évolution

- Données changent → nouveau modèle.
- Exemple : Data v1 → modèle v1, Data v2 → modèle v2.
- Métaphore : nouvelle photo avec le même appareil.

Déploiement fiable

- Versioning des modèles
- Rollback immédiat
- Canary release (déploiement progressif)

Monitoring et retraining automatique

- Détection du drift (data drift, concept drift)
- Relance automatique du pipeline
- Le modèle reste “vivant” et performant

—

3 Reproductibilité en ML (détail)

Conditions

- Données versionnées (ex : DVC)
- Code versionné (Git)
- Seed fixe (numpy, sklearn, torch...)
- Environnement stable (Docker)
- Pipeline automatisé (CI/CD)

Objectif

Pouvoir refaire exactement le même modèle n’importe quand.

Différence avec évolution

- Reproductibilité → mêmes données → même modèle
- Évolution → nouvelles données → nouveau modèle

—

4 Déploiement fiable : concepts clés

Versioning du modèle

Chaque modèle a un nom, version, métriques, commit Git, dataset.

Rollback automatique

Détecte anomalies ou drift → revient à une version stable.

Canary release

Nouveau modèle servi à une partie du trafic pour tests réels avant prod complète.
Métaphore : avion / autopilote qui reprend le contrôle si problème.

—

5 Monitoring & retraining automatique

Surveillance en production

- Données : distribution, valeurs manquantes, nouvelles catégories
- Modèle : prédictions, confiance
- Business : KPI (taux de conversion, fraude...)

Détection du drift

Règles automatisées (KS test, variation prédiction, métriques métier)

Retraining automatique

Nouvelles données → nouveau training → tests → modèle validé → déploiement

Métaphore : ton modèle = sportif, CI/CD = coach automatique qui le remet en forme quand nécessaire.

6 Architecture d'un pipeline CI/CD ML

Vue globale

```
Git (code)
↓
CI (tests, training, logging)
↓
Model Registry (versionning, staging/production)
↓
CD (déploiement, canary, rollback)
↓
Production (API ML)
↓
Monitoring → déclenche retraining
```

Briques importantes

- Git : versionne le code
- CI : tests + entraînement + validation
- Data validation : vérifie les données avant training
- Model Registry : stocke et versionne les modèles (MLflow)
- CD : déploie, canary, rollback
- Production : service API

- Monitoring : détection du drift
- Retraining automatique : pipeline relancé si besoin

—

7 MLflow dans le pipeline CI/CD

Rôle

- Tracking : log des paramètres, métriques, artefacts
- Model Registry : version, stage, production, rollback
- Projects : reproductibilité
- Models : déploiement simple (API REST)

Limite

MLflow ne gère pas tout le CI/CD → il doit être intégré à un pipeline d’orchestration (GitHub Actions, Jenkins, Airflow, Kubeflow...).

—

8 Métaphores à retenir

Concept	Métaphore
Reproductibilité	refaire exactement la même recette
Évolution	nouvelle photo avec le même appareil
Registry	bibliothèque officielle pour modèles
Rollback	autopilote d’un avion
Monitoring & Retraining	coach automatique

—

9 Phrase clé finale

“Un pipeline CI/CD pour ML orchestre l’entraînement, la validation, le versioning, le déploiement, le monitoring et le retraining automatique des modèles, garantissant reproductibilité, fiabilité et adaptation continue.”