



université  
virtuelle  
Burkina Faso



CITADEL

Centre d'Excellence Interdisciplinaire en  
Intelligence Artificielle pour le Développement

# Déploiement et intégration de modèles de Machine Learning en environnements de production

CITADEL Summer School 2025

By : Abdoul Aziz BONKOUNGOU



# Plan de la présentation

---

1. Cycle de vie d'un projet de Machine Learning
2. Spécificités des projets ML par rapport aux projets logiciels classiques
3. Déploiement des modèles : stratégies et architectures
4. Outils et plateformes de gestion de projets ML
5. Focus sur Mlflow
6. Cas d'utilisation et bonnes pratiques
7. Travaux pratiques (Labs)



# Pourquoi le déploiement est un défi en ML ?

- Un modèle performant n'est pas forcément un modèle exploitable
- Un modèle ML n'a de valeur que s'il est utilisé en production
- Le déploiement pose des défis spécifiques :
  - Données évolutives (data drift)
  - Reproductibilité des résultats
  - Surveillance des performances
  - Intégration avec des systèmes existants
- Beaucoup de projets ML échouent après la phase de modélisation



D'où l'importance du cycle de vie ML et du MLOps



# Cycle de vie d'un projet logiciel classique

Cycle de vie d'un projet logiciel (rappel)

1. Planning
2. Design
3. Développement
4. Tests
5. Déploiement
6. Maintenance



**Hypothèse clé :**

Le comportement du logiciel est déterministe  
Les règles métier sont stables dans le temps



# Cycle de vie d'un projet Machine Learning

1. Préparation des données
2. Analyse exploratoire (EDA)
3. Feature Engineering
4. Entraînement du modèle
5. Validation et évaluation
6. Déploiement
7. Monitoring et amélioration continue



## Définition : Différence majeure :

- Le modèle apprend à partir des données
- Les données et les performances évoluent dans le temps
- Le modèle peut devenir mauvais sans “bug”
  - Un modèle ML vieillit.



# Comparaison : logiciel vs Machine Learning

Logiciel classique VS Projet ML

Logiciel	Machine Learning
Code centré	Données + code
Comportement stable	Comportement évolutif
Tests unitaires	Validation statistique
Déploiement unique	Réentraînements fréquents

→ Le ML nécessite une approche itérative et automatisée



# Qu'est-ce que le MLOps ?

## MLOps (Machine Learning Operations)

- Extension du DevOps au Machine Learning
- Objectif : industrialiser le cycle de vie ML
- Combine : Data Engineering, Machine Learning, DevOps

## Principes clés :

- Automatisation
- Traçabilité
- Reproductibilité
- Surveillance continue



# Stratégies de déploiement des modèles ML

Deux grandes familles de déploiement

## 1. Déploiement embarqué (on-device)

- Mobile, IoT, edge computing
- Avantages : faible latence, confidentialité
- Contraintes : ressources limitées

## 2. Déploiement via API (REST / gRPC)

- Modèle hébergé sur un serveur
- Appels via HTTP
- Solution la plus courante en entreprise



En plus :

- On mesure le temps de réponse
- On garde des traces des requêtes



# Architectures de déploiement courantes

---

## Architecture typique via API

- Client (Web / Mobile / Backend)
- API REST
- Serveur de modèles
- Base de données / Data lake

## Bonnes pratiques :

- Conteneurisation (Docker)
- Scalabilité
- Séparation entraînement / inférence



# Infrastructures Cloud pour le ML

---

## Infrastructures couramment utilisées

- AWS (SageMaker, EC2, EKS)
- Google Cloud Platform (Vertex AI, GKE)
- Microsoft Azure (Azure ML)

## Avantages du cloud

- Scalabilité
- Coût à l'usage
- Services managés pour le ML



# Outils et plateformes ML

## Écosystème d'outils ML

- Développement & visualisation : Jupyter, Streamlit, TensorBoard
- Suivi et gestion : MLflow, Weights & Biases
- Déploiement : Docker, Kubernetes, FastAPI, Flask



**MLflow se positionne comme un outil central du MLOps**



# MLflow : définition

---

## MLflow, c'est quoi ?

- Plateforme open source développée par Databricks
- Objectif : gérer l'ensemble du cycle de vie ML
- Permet de :
  - Suivre les expériences
  - Gérer les modèles
  - Faciliter le déploiement
  - Collaborer en équipe



# Pourquoi utiliser MLflow ?

---

## Problèmes courants sans MLflow

- Résultats non reproductibles
- Perte des paramètres d'entraînement
- Modèles non versionnés
- Difficulté à comparer les expériences

## MLflow apporte

- Organisation
- Transparence
- Standardisation



# Cas d'utilisation de MLflow

---

## MLflow est utilisé pour

- Gestion des expérimentations ML
- Comparaison de modèles
- Reproductibilité des résultats
- Déploiement en production
- Collaboration data scientists / ingénieurs



# Composants principaux de MLflow

---

Les 4 composants majeurs (Partie 1)

## 1. MLflow Tracking

- Paramètres
- Métriques
- Artifacts (modèles, figures)

## 2. MLflow Projects

- Standardisation des projets
- Reproductibilité des environnements



# MLflow Models et Registry

---

Les 4 composants majeurs (Partie 2)

## 3. MLflow Models

- Format standard pour les modèles
- Chargement et déploiement simplifiés

## 4. MLflow Model Registry

- Versioning des modèles
- États : Staging, Production, Archived
- Collaboration et gouvernance



# Exemple de workflow avec MLflow

---

## Workflow typique

1. Entraîner plusieurs modèles
2. Logger paramètres et métriques
3. Comparer les performances
4. Enregistrer le meilleur modèle
5. Déployer en production
6. Surveiller et réentraîner si nécessaire



# Bonnes pratiques en déploiement ML

---

## Recommandations clés

- Versionner données, code et modèles
- Automatiser les pipelines
- Monitorer la performance post-déploiement
- Anticiper le data drift
- Documenter les décisions



# Limites et défis

---

## Attention

- MLflow ne remplace pas Kubernetes ou CI/CD
- Le monitoring avancé nécessite des outils complémentaires
- La gouvernance des données reste un enjeu majeur



# Conclusion

---

## À retenir

- Le déploiement ML est un processus complexe
- Le MLOps est indispensable en environnement réel
- MLflow est un outil clé pour structurer les projets ML
- L'objectif final : fiabilité, performance et valeur métier



# Labs / Travaux pratiques

## Objectifs des labs

- Utiliser MLflow pour suivre une expérience
- Enregistrer et comparer plusieurs modèles
- Déployer un modèle simple via API
- Observer le comportement en production



<https://github.com/azizYaaba/citadel-sc-model-deployment>



# Fin de la présentation



→ Des questions ? Discutons-en !



abbonkoungou@gmail.com

