Introduction au MLOPS

M. Bouneffa
LISIC
EILCO-ULCO
Janvier 2026

MLOPS???

- Terme récent : apparu vers 2015
- S’utilise vraiment à partir de 2017/2018
- Les notions ne sont pas encore bien figées

Structure du cours

- Introduction
- Le cycle de vie du Machine Learning
- Définition du MLOps
- Les principaux termes et outils à connaître

Problématique principale

Comment utiliser le Machine Learning en Production ?

Exemple :

- *Une entreprise de e-commerce qui souhaite utiliser l'IA pour diffuser des publicités personnalisées à ses clients.*
- Elle va faire appel à une équipe spécialisée pour **créer un modèle** et le **mettre en production**.

Les différentes étapes de mise en oeuvre d'une IA

1. Data Collection
2. Preprocessing
3. Modélisation

Machine Learning / Data Scientist

Traditionnellement **Machine Leraning** et **Operations** sont **séparées**

1. Testing
2. Mise en production
3. Monitoring

Operations / Data Engineer

Après 9 mois de développement, les 6 étapes sont terminées

L'entreprise peut-elle utiliser cette application sans difficultés?

Quels problèmes peut-on rencontrer ?

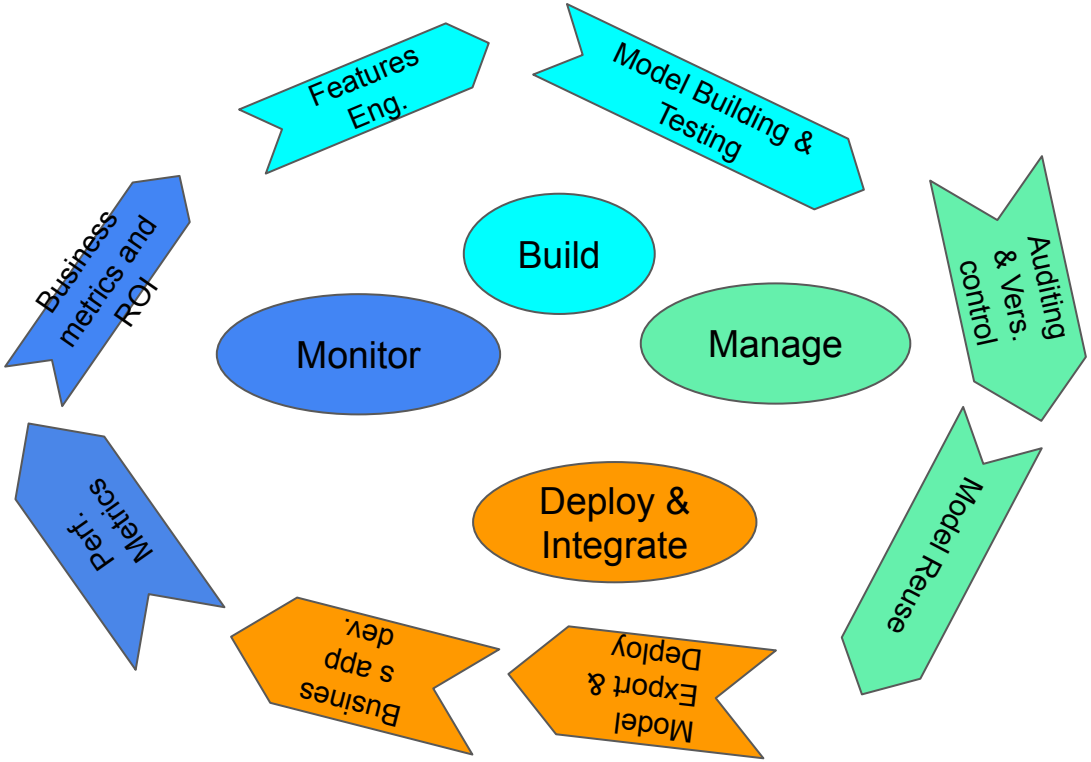
Plusieurs problèmes potentiels :

- Les données utilisées lors de l'entrainement sont obsolètes
- Le modèle n'est plus adapté aux besoins de l'entreprise
- L'application ne peut pas être mise à l'échelle
- L'application nest pas réutilisable

Il est donc nécessaire de travailler :

- plus rapidement
- en continue
- en parallèle

Cycle de Vie du ML



4 phases :

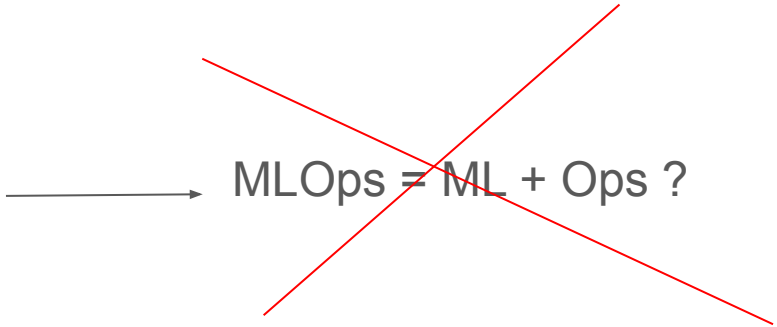
- -Build
- -Manage
- -Deploy & Integrate
- -Monitor

3 objectifs :

- -Automatisder
- -Isoler
- -Surveiller

Définition du MLOps

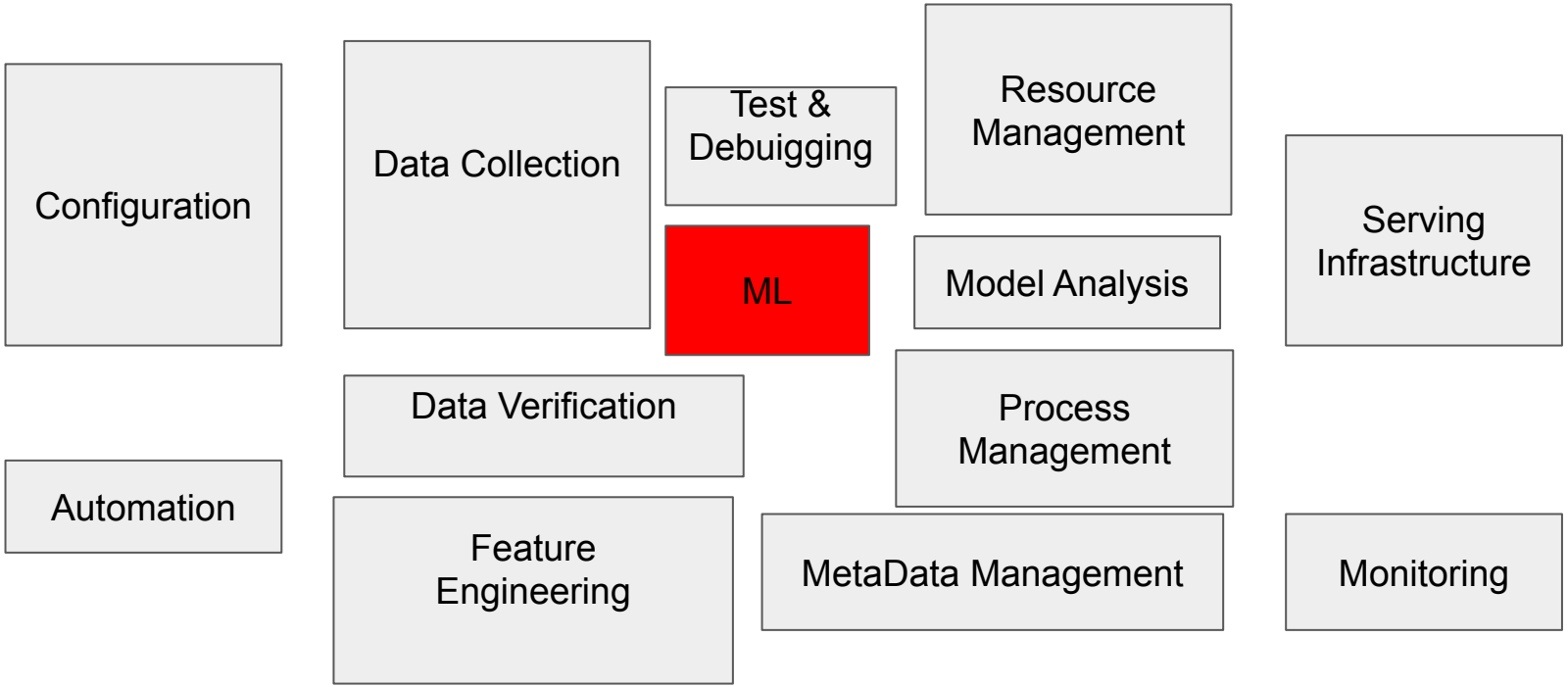
1. Data Collection
2. Preprocessing
3. Modélisation
4. Testing
5. Mise en production
6. Monitoring



Définition du MLOps

- MLOps n’est pas un métier, c’est **un ensemble de pratiques.**
- Dans un environnement MLOps les data scientists, ML engineers, et Data Engineers collaborent pour automatiser, montorer, gouverner et segmenter le cycle de vie du Machine Learning.

Définition du MLOps



Mots clés

- DevOps / Agilité
- Mise en production / Déploiement
- Monitoring
- CI/CD (Intégration continue / Développement et déploiement continus)
- API (Application Programming Interface)
- Data Drift / Model Shift
- Testing

Les outils

Versioning

- Github
- Gitlab
- Data Version Control (DVC)

Continuous Integration / Continuous Deployment (CI/CD)

- Pytest
- Jenkins
- Github/Gitlab

Les outils

Versioning

- Github
- Gitlab
- Data Version Control (DVC)

Continuous Integration / Continuous Deployment (CI/CD)

- Pytest
- Jenkins
- Github/Gitlab

Les outils

Virtulaisation

- Docker

API

- Fast API
- Flask
- Django

Les outils

Monitoring

- mflow
- Prometheus
- Grafana

Orchestration

- Airflow
- Kubernetes
- Kubeflow

Les outils

Les services cloud

- Amazon Sagemaker
- Azure ML
- Vertex AI

Les cerficiations cloud

- ML/Speciality (Aws Certified)
- Azure AI Engineer (Microsoft Certified)
- ML Engineer (Goocle Cloud Certified)

Conclusion

Le MLOps est un sujet à la mode, récent et encore mal compris par certains acteurs de la Data Science.

Les outils et technologies évoluent rapidement, l est nécessaire de se tenir informé et de se remettre en question.

Sujets MLOps – Attendus & Références

Sujet 1 — Pourquoi le MLOps ?

Attendus (ce que tu veux voir)

- Expliquer **pourquoi un modèle ML échoue en production**
- Introduire la notion de **dette technique en ML**
- Faire la différence entre :
    - *entraîner un modèle*
    - *exploiter un système ML*
    - **Illustration pratique attendue**
- **Exemple Python simple** :
    - modèle entraîné
    - modification des données d'entrée
    - dégradation visible des performances

      (ou étude de cas technique si le code est minimal)

Références de départ

- Google — *MLOps: Continuous Delivery and Automation Pipelines* (whitepaper)
- Sculley et al. — *Hidden Technical Debt in ML Systems* (article + résumés en ligne)
- Blog : *Why ML models fail in production*

Sujet 2 — Cycle de vie du Machine Learning

Attendus

- Décrire les étapes :
  - collecte des données
  - entraînement
  - déploiement
  - surveillance
- Comparer **projet ML** vs **projet logiciel classique**
- Identifier **les points de rupture possibles**

Illustration pratique attendue

- **Schéma clair du cycle de vie**
- Script ou notebook Python illustrant un pipeline ML simplifié

Références

- Google — *ML lifecycle overview*
- Article : *ML systems vs traditional software*
- Blog pédagogique : *End-to-end ML lifecycle*

Sujet 3 — Gestion des versions en ML : code, données, modèles

Attendus

- Expliquer pourquoi **Git ne suffit pas**
- Distinguer :
  - version du code
  - version des données
  - version des modèles

Illustration pratique attendue

- Exemple concret de versioning de données (CSV)
- Utilisation de **DVC** ou équivalent (local)

Références

- DVC — *Get Started*
- Blog : *Data versioning for machine learning*
- Documentation Git (rappels simples)

Sujet 4 — Suivi des expériences et reproductibilité

Attendus

- Montrer pourquoi on ne sait plus quel modèle est “le bon”
- Expliquer le rôle du **suivi des expériences**
- Introduire les notions de paramètres, métriques, artefacts

Illustration pratique attendue

- Notebook Python :
  - plusieurs entraînements
  - suivi avec **MLflow**
  - comparaison des résultats

Références

- MLflow — *Quickstart*
- Blog : *Why experiment tracking matters*
- Documentation scikit-learn (pour les modèles simples)

Sujet 5 — Déploiement de modèles par API

Attendus

- Expliquer la différence entre :
  - traitement par lots
  - prédiction à la demande
- Montrer pourquoi une API est centrale

Illustration pratique attendue

- API simple en Python (FastAPI ou équivalent)
- Endpoint `/predict` recevant des données JSON

Références

- FastAPI — *Official Tutorial*
- Blog : *Serving ML models with FastAPI*
- Article : *From model to service*

Sujet 6 — Tests et automatisation en Machine Learning

Attendus

- Expliquer ce que signifie **tester un système ML**
- Distinguer :
  - tests de données
  - tests de modèles
  - tests de performance

Illustration pratique attendue

- Scripts Python avec **tests simples** (pytest)
- Exemples de tests sur données ou sorties de modèles

Références

- Pytest — *Getting Started*
- Blog : *Testing machine learning systems*
- Article : *CI/CD challenges in ML*

Sujet 7 — Dérive des données et des modèles

Attendus

- Définir clairement :
  - dérive des données
  - dérive du modèle
- Expliquer pourquoi un modèle se dégrade avec le temps

Illustration pratique attendue

- Comparaison de distributions (avant / après)
- Visualisation ou statistique simple en Python

Références

- Evidently AI — *Data Drift Tutorials*
- Blog : *What is data drift?*
- Article pédagogique : *Model decay explained*

Sujet 8 — Surveillance des modèles en production

Attendus

- Expliquer ce qu’il faut surveiller après déploiement
- Distinguer performance, qualité des données, usage

Illustration pratique attendue

- Notebook ou script illustrant le suivi de métriques
- Exemple de tableau de suivi commenté

Références

- Evidently AI — *Monitoring ML models*
- Blog : *ML model monitoring basics*
- Article : *Post-deployment ML monitoring*

Sujet 9 — Pipelines ML et automatisation

Attendus

- Expliquer pourquoi automatiser les étapes ML
- Introduire la notion de **chaîne de traitement**

Illustration pratique attendue

- Pipeline ML scripté en Python
- Ou démonstration conceptuelle avec étapes clairement séparées

Références

- Article : *What is an ML pipeline?*
- Documentation Airflow (lecture introductive)
- Blog : *ML pipeline best practices*

Sujet 10 — Plateformes MLOps (analyse critique)

Attendus

- Comprendre ce que proposent réellement les plateformes cloud
- Identifier avantages et limites
- Porter un **regard critique**, pas marketing

Illustration pratique attendue

- Étude de cas technique commentée
- Analyse d'une architecture type (schéma + explication)

Références

- AWS SageMaker — *Getting Started*
- Azure ML — *Overview*
- Google Vertex AI — *Introduction*
- Blog comparatif : *MLOps platforms compared*