Litterature review on MLOps

Implementing an MLOps approach for production-grade ML

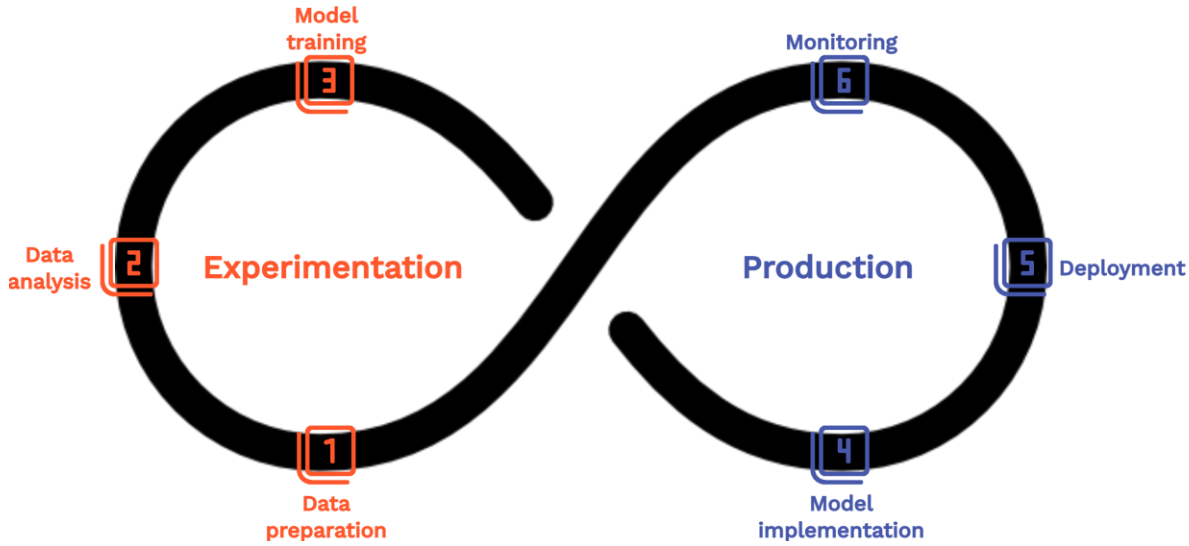
France

Introduction

Machine Learning Operations (MLOps) has emerged to bridge the gap between building machine learning models and reliably deploying, monitoring, and maintaining them in production. Traditional software DevOps principles alone are insufficient for ML systems, which face unique challenges such as data dependency, reproducibility issues, and model degradation over time. In fact, an industry survey in 2020 found that 55% of companies experimenting with ML had not deployed a single model to production, and 18% took over 90 days to deploy one. These delays and complexities highlight how delivering ML in production is not an easy task. The actual model code is only a tiny component of an ML system; configuration, data pipelines, feature engineering, testing, monitoring, and other components make up the bulk of the system's infrastructure. Without a disciplined approach, this surrounding complexity can become "technical debt," undermining system reliability and agility as defined in *Sculley, D et al.* (2015).

What is MLops?

The MLOps approach can be seen as an extension of DevOps, developed to address the specific challenges related to managing the lifecycle of ML models. Fundamentally, both DevOps and MLOps aim at building software in a more automated and robust manner. The main difference is that in MLOps, this software also has a machine learning component. Consequently, the lifecycle of the project gets more complex. The underlying ML model needs to be re-trained regularly, to avoid any loss of performance over time. Data ingestion must also be included in the pipeline, as new data may be used to improve performance. In essence, practicing MLOps means applying continuous integration (CI), continuous deployment (CD), and even continuous training (CT) principles to ML workflows. By automating routine processes and instituting rigorous validation, MLOps enables incremental and reliable deployment of ML improvements in production. Below figure presents the steps of an ML project using the continuous representation traditionally seen in DevOps. This illustrates a fundamental principle of MLOps, which is the need for continuous improvement.



Tools and platforms for MLOps

MLOps encompasses a broad range of tools and platforms that address various stages of the ML lifecycle, from data preparation and experiment tracking to model deployment and monitoring. In recent years, a proliferation of MLOps platforms has emerged. These include open-source toolkits and frameworks as well as proprietary cloud platforms. A comprehensive 2025 survey by *Berberi et al.* examined 16 popular open-source MLOps tools and highlighted core capabilities that a robust MLOps solution should support. Key features identified include: workflow orchestration, automated training pipelines, experiment tracking and metadata management, model versioning and registry, scalable model serving, and performance monitoring. Notably, the study found that **model performance monitoring** was the least supported feature among many tools, indicating a gap that organizations need to address separately.

Open-source platforms have been instrumental in driving MLOps adoption. For example, Kubeflow is a Kubernetes-based toolkit aimed at making ML workflow deployments portable and scalable. It provides pipelines for end-to-end training and serving, and supports components for distributed training, model development, deployment, and experiment tracking. Its modular design lets teams use the entire suite or pick and choose components, integrating with containerized infrastructure. Another widely used tool is MLflow and has been detailed in Romain Avouac et al. MLflow focuses on experiment tracking, model packaging, and model registry; it allows data scientists to record training runs, metrics, and artifacts for reproducibility and later deployment. Many other tools[1] exist that might be relevant depending on your IT skills, requirements, or preferences. If you don't have specific constraints, a good indicator to consider is the number of GitHub stars for the open-source project.

[1] We can cite TensorFlow Extended (TFX) Weights & Biases (W&B), ClearML, Airflow or Pachyderm and Feast for more specific needs.

Alongside open-source solutions, all major cloud providers offer managed MLOps platforms. Services such as AWS SageMaker, Microsoft Azure ML, and Google Cloud Vertex AI bundle data processing, training, deployment, and monitoring into integrated workflows. These cloud platforms can accelerate implementation by handling infrastructure concerns, though at the expense of higher ongoing costs and potential vendor lock-in. Organizations often weigh these trade-offs: while cloud MLOps services provide convenience and scalability out-of-the-box, many teams opt for open-source stacks to maintain flexibility and control. In practice, a hybrid approach is also common – for instance using open-source MLflow for experiment tracking and a cloud service for model deployment – to leverage strengths of each. Additionally, platforms like SSPCloud offer an open architecture that can be internally redeployed using Onyxia software, representing yet another trade-off between developing in-house expertise or outsourcing these skills.

Organizational challenges

Implementing MLOps in practice often encounters organizational challenges beyond the technical hurdles. A recurring theme in recent practitioner reports is that *"MLOps is an organizational problem, not a technology problem" (see Mailach et al (2023)*). While tools and automation are necessary, the hardest parts can be aligning teams, processes, and incentives to support the machine learning lifecycle. Research at the intersection of software engineering and data science identifies several common socio-technical challenges and provides insights into how organizations are overcoming them.

One major challenge is the siloing of roles and responsibilities. In many enterprises, data scientists (who build models) and IT or software engineers (who deploy and operate systems) belong to different teams with different priorities. This separation can lead to miscommunication, "throwing models over the wall," and lack of ownership of the end-to-end solution. *Mailach et al. (2023)* conducted an extensive qualitative study of MLOps practitioners and identified *17 anti-patterns*, many of which stem from poor coordination between teams or unclear organizational processes. For example, one anti-pattern is *"Requesting data is hard"* – where an ML team struggles to obtain needed data from another department due to bureaucracy or misaligned incentives. Another issue observed is *"Tight coupling between producer and consumer teams"*, meaning roles are not clearly divided and everything becomes entangled, making it unclear who is accountable for data quality or model failures. Such findings echo earlier studies that noted *"lack of documentation, non-valued engineering, and unclear processes"* impede collaboration between ML and software teams. In essence, many organizations have had to confront cultural shifts – valuing reproducibility and engineering rigor in data science work, and fostering collaboration between data experts and infrastructure experts.

To address these issues, companies are instituting cross-functional teams and clearer MLOps roles. Successful MLOps implementations often involve hybrid roles like *ML Engineers* or *MLOps Engineers* who possess both data science understanding and software engineering skills. They act as bridges between pure data science and DevOps, ensuring that models are not only accurate but also deployable and maintainable. Recent literature stresses the importance of *"clearly defining and allocating the resources needed to implement MLOps"*, including human roles. *Stone et al. (2025)*

propose an MLOps maturity framework that explicitly maps roles (business analysts, data scientists, ML engineers, DevOps, etc.) to each stage of the lifecycle, helping organizations identify skill gaps and invest in the right talent. The need for such role clarity is also seen in national statistical offices implementing MLOps, where teams include domain experts, data scientists, IT specialists, and even privacy officers to ensure all perspectives are covered.

Implementing MLOps is as much about people and processes as it is about pipelines and code. Organizations that invest in breaking down silos, clarifying roles, and cultivating a culture of transparency and collaboration tend to navigate the MLOps transition more smoothly as shown in *Mailach et al. (2023).*

MLOps in NSI

National Statistical Institutes (NSIs) and official data agencies present a unique context for MLOps. We are increasingly exploring machine learning to enhance official statistics – for example, using ML to classify survey text, detect anomalies in data editing, or integrate alternative data sources (like satellite imagery or scanner data) into economic indicators. However, NSIs operate under strict requirements for accuracy, privacy, transparency, and reproducibility. The adoption of MLOps in such settings is driven by the need to institutionalize ML processes so that they meet the same standards as traditional official statistics pipelines. Recent resources specifically addressing MLOps for official statistics highlight both the opportunities and challenges (see Ritter *et al.* (2024) and *Avouac et al*. (2024)).

A 2024 review by *Nunes and Ashofteh (2024)* surveyed applications of data science in official statistics and found that while many NSOs have piloted ML methods, there is a *"research gap in the post-training steps of the ML process,"* including how to reliably deploy and maintain models in production. In other words, plenty of prototypes exist, but operationalizing these into repeatable production processes (the essence of MLOps) remains an open challenge. Recognizing this, international groups like the UNECE HLG-MOS (High-Level Group for Modernisation of Official Statistics) have begun developing guidance. The UNECE's recent report (2024) outlines objectives for NSOs to *"ensure seamless integration, deployment, monitoring, and maintenance of ML models"* while upholding official statistics principles

Statistics Canada provides a concrete case study of MLOps adoption in the national statistics context. Ritter *et al.* (2024) describe how StatCan's Data Science Division implemented MLOps for a machine learning system supporting the Canadian Consumer Price Index (CPI). The CPI use case involves using ML to automatically classify products from alternative data sources (like retail scanner data) into CPI categories. Because the product universe changes over time (new products appear, old ones disappear), the ML model faces continuous concept drift. StatCan addressed this by investing in an MLOps pipeline that monitors model predictions each month, validates samples of the automated classifications, and triggers frequent retraining with up-to-date data. They integrated processes for data and model monitoring, human review of flagged cases, and scheduled model refreshes, ensuring the model's degradations are caught and corrected before they can affect the published price index.

Other NSOs are also moving in this direction. In France, Insee has leveraged cloud-native technologies to significantly enhance its ability to operationalize machine learning models in accordance with MLOps principles. In *Avouac et al*. (2024), they highlight a concrete example: the improved process for classifying French companies' activities according to the NACE classification. By integrating tools such as MLflow and employing cloud-native technologies, Insee successfully implemented training, deployment, and monitoring cycles for its machine learning models, significantly increasing the efficiency and accuracy of official statistical production.

Still, challenges remain, especially given resource constraints and the need for upskilling staff in new technologies. Organizationally, NSIs often need to blend the culture of official statisticians (who are accustomed to rigorous but slower-moving processes) with the more experimental culture of data scientists. MLOps provides a framework for this blend: it introduces discipline and automation that can satisfy rigor, while preserving the agility to update models frequently.

References

1 Sculley, D & Holt, Gary & Golovin, Daniel & Davydov, Eugene & Phillips, Todd & Ebner, Dietmar & Chaudhary, Vinay & Young, Michael & Dennison, Dan. (2015). Hidden Technical Debt in Machine Learning Systems. NIPS. 2494-2502. URL

2 Berberi, L., Kozlov, V., Nguyen, G. et al. Machine learning operations landscape: platforms and tools. Artif Intell Rev 58, 167 (2025). URL

3 Romain Avouac, Thomas Faria, Frédéric Comte. A cloud-native data science platform for official statistics (2024). URL

4 Fredéric Comte, Arnaud Degorre, and Romain Lesur. SSPCloud: a creative factory to support experimentations in the field of official statistics. Courrier des Statistiques, INSEE, 7:68–85, (2022). URL

5 . Mailach and N. Siegmund, "Socio-Technical Anti-Patterns in Building ML-Enabled Software: Insights from Leaders on the Forefront," 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, (2023). URL

6 J. Stone, R. Patel, F. Ghiasi, S. Mittal, and S. Rahimi, "Navigating MLOps: Insights into Maturity, Lifecycle, Tools, and Careers", (2025). URL

7 C. Ritter, W. Spackman, T. Best, S. Goussev, G. DeVilliers, and S. Theobald, "Adopting a High Level MLOps Practice for the Production Applications of Machine Learning in the Canadian Consumer Prices Index," Statistics Canada – Data Science Network, (2024). URL

8 C. E. Ramos Nunes and A. Ashofteh, "A Review of Big Data and Machine Learning Operations in Official Statistics: MLOps and Feature Store Adoption," 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), Osaka, Japan, (2024). URL

9 UNECE High-Level Group for Modernisation of Official Statistics, “Organisational aspects of implementing ML based data editing in statistical production” Workshop on ML for Data Editing, (2024). URL

10 Google Cloud Architecture Center, “MLOps: Continuous delivery and automation pipelines in machine learning,” (2024). URL