Research Article *Open Access*

Journal of Pioneering Artificial Intelligence Research

ISSN: 3069-0846

DOI: doi.org/10.63721/26JPAIR0123

MLOps Tool Technology Review: Analysis of Support for the Machine Learning Model Lifecycle and Application Recommendations

Helder Rodrigo Pinto

ISTEC Porto, Porto, Portugal

Citation: Helder Rodrigo Pinto (2026) MLOps Tool Technology Review: Analysis of Support for the Machine Learning Model Lifecycle and Application Recommendations. J of Poin Artf Research 2(1), 1-108. WMJ-JPAIR-123

Abstract

This paper critically evaluates MLflow, an open-source MLOps tool for managing the lifecycle of machine learning models. It emphasizes its strengths in experiment tracking, model versioning, and modularity, which support reproducibility and scalability across various frameworks and cloud providers.

However, it points out crucial limitations such as the lack of native monitoring (data drift, fairness), detailed permission control and full integration with CI/CD orchestrators. These shortcomings require complementary tools, increasing complexity and operating costs, especially for regulated environments or those with less technical maturity.

In conclusion, MLflow is recommended for small and medium-sized companies, or teams that need rapid prototyping and flexibility. For large organizations or regulated sectors, although useful in the initial phase, it requires the integration of other tools for complete and robust management of the model lifecycle.

***Corresponding author:** Helder Rodrigo Pinto, ISTEC Porto, Porto, Portugal.

Submitted: 19.01.2026 **Accepted:** 24.01.2026 **Published:** 05.02.2026

Keywords: MLflow, MLOps, Machine Learning, Open Source

Introduction

The increasing use of artificial intelligence (AI) and machine learning (ML) has transformed how companies across different sectors solve problems, optimize processes, and innovate in their markets. In this context, managing the machine learning lifecycle becomes a key factor for the operationalization, sustainability, and compliance of these systems in increasingly complex and dynamic environments. As ML solutions need to be scaled and operated,

the challenges surrounding governance, security, and efficient model lifecycle management intensify, requiring MLOps solutions that enable versioning, scalability, and automation of model development and deployment.

This work's critical analysis focuses on the study of MLflow, a widely used open-source MLOps tool for managing experiments, registration, packaging, and deployment of machine learning models . This choice is motivated by several factors, including broad support for various languages, frameworks , and AA libraries, as well as MLflow's compatibility with cloud environments . MLflow offers a standard workflow for the development and operationalization of machine learning models, helping users manage the experimentation, reproducibility, deployment, and monitoring phases in a standardized and organized way. Furthermore, MLflow's open-source license allows it to be used by companies of different sizes and niches, as well as run on their own infrastructure or in the cloud .

The main objective of this work is to conduct a critical and comprehensive analysis of MLflow, detailing its advantages and disadvantages in each phase of the lifecycle of a machine learning model.

The research question posed is as follows: what are the advantages and disadvantages of MLflow as an MLOps tool for managing the lifecycle of machine learning models in different organizations, in terms of data management, experimentation, CI/CD, monitoring, governance, and security?

Methodology and Current Scenario

To answer the research question, the methodology of this work is based on bibliographic and documentary research, reviewing articles, technical documents, and current studies related to MLOps and MLflow. For the critical analysis of the tool, comparative and evaluative methods are used to examine its compatibility with other MLOps tools and its adherence to best practices in the development, implementation, and operation of machine learning.

The current landscape of MLOps research, and consequently MLflow, shows an advancement in optimization technologies for the large-scale implementation of AI within companies, but also denotes the existence of challenges in implementing this process. Although automation and MLOps tools offer benefits such as reproducibility, auditability, and scalability, some operational challenges persist, such as low data quality, lack of qualified professionals, system integration, and compliance with regulatory requirements.

Results of the Comparative Analysis

A comparative analysis of MLflow with other tools, such as ClearML and ZenML, reveals that it delivers good results for managing the development lifecycle phases, but still has market gaps in other areas, such as model monitoring and data management. Analysis of best practices showed that it is possible to obtain excellent results with MLflow, but some points still need improvement. In some cases, other platforms available on the market offer more complete and satisfactory results, especially when seeking automated monitoring and granular data management.

Article Structure

This work is structured as follows: Chapter 2 contextualizes the tool, describing what MLflow is and where this tool comes from within the MLOps ecosystem; Chapter 3 explains the tool in detail, with all available resources, addressing each phase of the machine learning model lifecycle ; Chapter 4 provides a critical evaluation of the tool, highlighting its advantages and disadvantages, and offering recommendations for using MLflow in different companies; and finally, Chapter 5 concludes the work by presenting the findings and possibilities for future research in this field.

MLflow: Contextualization and Purpose

MLflow was created by Databricks to be an open-source platform aimed at managing the lifecycle of a machine learning model , due to the lack of tools that allow traceability, reproducibility, and automation in organizational-level machine learning projects . According to, MLflow, due to its ability to standardize the different technologies used, serves companies that have heterogeneity of frameworks in their computing environment [1].

Open source licensing and the fact that it is not specific to any cloud computing provider are factors that allow it to be used both on-premise servers and in the largest cloud computing providers, offering greater

adaptability to companies that use it, whether small or large, as reported by [2].

Compared to other lifecycle tools, such as Google's TFX and Uber's Michelangelo, MLflow allows for more integrations with different frameworks , such as TensorFlow, PyTorch, and Scikit-learn, which better meets the demand for different workflows. As reported by, this has enabled MLflow to be adopted on a larger scale in projects aiming to port the technology to other environments [2].

Key Capabilities and Scope of Use

Experiment and model versioning stands out as one of the key aspects to be used in projects that need to transfer experiments to production-level environments and that are controlled by change management processes. According to, the modularity of the tool with the Tracking , Projects , Models , and Registry modules is one of the solutions found by machine learning teams to address the challenge of traceability and artifact centralization. This modularity is the main pillar that facilitates the pursuit of reproducibility and allows for governance management in complex projects [3].

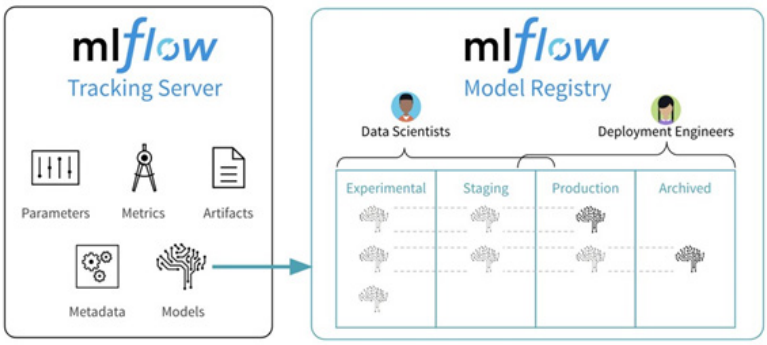

Figure 1: MLFlow Tracking Server, Registration Model [4].

MLflow's ease of integration with different frameworks and its simplicity of use have led startups and smaller teams to adopt it. In MLflow, experimentation, versioning, and deployment processes are centralized, reducing overhead and operational costs for small teams, as reported by [5]. Conversely, for larger companies, MLflow brings automation of operations and standardization of practices, resulting in scalability.

The ability to run in different environments also allows MLflow to adapt to different experimentation flows in distinct pipelines , serving companies that want to more quickly adopt MLOps practices in their machine learning projects. According to, this is the characteristic that allows companies to innovate faster. For this feature to be applied in the best way, the company needs to have a well-established cloud computing infrastructure to take advantage of the platform's elasticity [5].

Impact on Governance, Collaboration, and Inherent Limitations

Another interesting point is the versioning and centralization of machine learning artifacts , such as models, parameters, and metrics [6]. Point out that this makes it possible to automate tasks, which helps prevent human errors and facilitates work in regulatory environments, where it is crucial to have total control over assets, as in the cases of the health and finance sectors.

Companies in the regulatory sector need to have all data collection steps mapped out, up to the available model, to serve external audit policies. With MLflow, it is possible to control the traceability of the complete execution of algorithms to ensure that the model's data provenance is not compromised and that data governance laws are followed. Highlight that the benefit of this control in regulatory projects is that it is possible to replicate and trace the delivered models, increasing control, reliability and, consequently, facilitating audits and approving processes in compliance requirements [6].

Projects with multiple stakeholders can also benefit from the workflow organization provided by MLflow, as it enables alignment of vision between compliance , research, and development teams. This allows for the identification of potential workflow bottlenecks and ensures that quality requirements are met, thus delivering models to production on time. According to, it is a method for managing different processes with the integration of all stakeholders [1].

Natively performing model fairness testing and lacking data drift monitoring in its dashboard . For its use in high-level compliance and security environments, numerous integrations with third-party tools are necessary, as reported by, increasing the operational effort of teams. This limitation is even more critical in companies that operate with sensitive and confidential data [7].

This overhead in tool integration does not help the progress of MLOps practices, given that it does not offer mechanisms for automating model testing. Point out that in complex projects, controlled by compliance , it is essential to automate continuous evaluation and the application of machine learning in production , and the limitations of this tool do not allow such practices to be established more quickly [7].

In short, MLflow allows for rapid experimentation and, because of its modularity and integration capabilities, allows for easy scalability. In some regulatory projects, however, it requires additional integrations to perform monitoring and security at scale.

MLflow Features and Resources

MLflow Tracking provides detailed experiment tracking, automatically logging parameters, metrics, source code, and artifacts from each run, ensuring reproducibility of results and comparison between different modeling approaches [1]. However, despite automation, there is inconsistent data and/or metric logging, which can compromise the tool's success.

Support for recording experiments in multiple languages, such as Python, R, and Java, facilitates collaboration between teams with different technology stacks [3]. However, since MLflow integrates different languages and frameworks , its efficiency can be questioned, as maintaining these different stacks on a single platform can increase the complexity of maintenance or auditing processes.

The tracking provided by MLflow facilitates monitoring the development of a machine learning model , as well as tracking audits of experiments [1]. However, this tracking depends on metadata, which cannot always be adequately maintained by all teams. Automated tracking of parameters and artifacts allows for the identification of regressions and internal best practices [8]. Automation, to avoid human error, can, in turn, generate a potential negative impact, as there is a possibility that tracking will not be effective when automation pipelines are not well configured.

MLflow Tracking lacks native integration with data versioning tools, which prevents the automated detection of data changes [9]. This problem is significant for organizations that work with large amounts of dynamic data, which need to be monitored for changes.

The MLflow Registry centralizes all versions of machine learning models, storing information on training data, validation metrics, and model code [3]. This centralization streamlines collaboration among team members and facilitates auditing, but it does not work in sectors with high levels of bureaucracy, such as finance and the military, because defining permissions is extremely important.

The shifting of machine learning models across various stages, such as preparation, production, and archiving, is supported through versioning and access controls for model governance [8]. However, due to limited access control, MLflow cannot be used in cases requiring segregation of duties, hindering its use in regulated industries [10].

Registering machine learning models allows for collaboration among machine learning team members, with the possibility of retrieving specific model versions for auditing or rollback if necessary [1]. However, there is a need to integrate MLflow with other tools to ensure the tracking of all machine learning information, such as library versions and project artifacts, which complicates the operational management of a machine learning model's lifecycle.

Modular Architecture and Project Management

MLflow's modular architecture, comprised of the

Tracking, Projects, Models, and Registry components, allows for customization of the tool according to each organization's needs [3]. However, modularity requires technical knowledge of the tool to correctly configure all components and ensure proper application throughout the machine learning lifecycle.

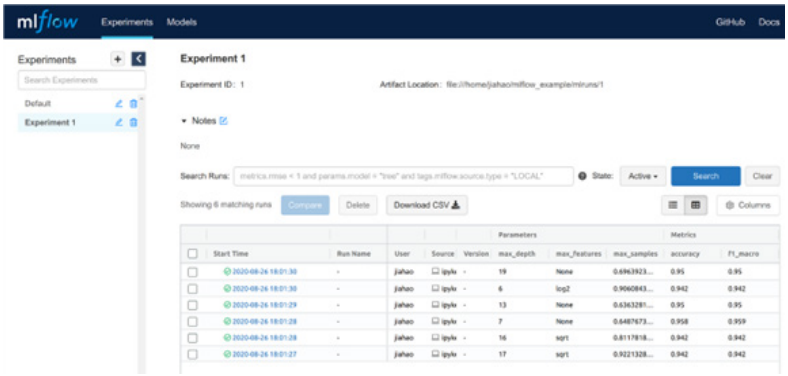

Figure 2: MLFlow Dashboard [11].

MLflow Projects uses a declarative approach to execute workflows , where project dependencies and environment variables can be specified in YAML files for portability across various environments, both locally and remotely [1]. Conversely, the need for knowledge of YAML files can hinder its use for people with little experience and increase human error due to the manual writing process.

Challenges and Functional Gaps

fairness monitoring , data drift , and integration with orchestrators such as Kubeflow and Airflow [12-13]. These advanced capabilities are important for cases where the machine learning pipeline is extremely robust, containing many steps.

MLflow can be customized to the needs of various organizations, but it can increase the technical complexity of the project, making it difficult for inexperienced teams to use [14]. This impact becomes negative, especially for small startups and public bodies.

Although YAML files make configuration sharing easier, they can cause problems when dealing with different configurations, which can lead to many dependency conflicts and, consequently, problems that require significant technical expertise to resolve [3]. For this reason, it is not appropriate for organizations with less technically skilled teams.

The absence of data management functionalities, such as automated data versioning and data drift detection , necessitates the use of other tools and increases operational risk [9]. The need to use other tools to ensure data versioning raises the operational cost of the tool and reduces the scalability of the machine learning model .

Although MLflow performs versioning of the data used in the model, it does not have an effective mechanism for tracking and managing data versions, since it does not allow for the automatic detection of changes and/or alterations made to the data used [8]. For this reason, it is necessary to use other tools to ensure this versioning and, in this way, reduce human errors and lack of information, which can increase operational risk and misuse of data.

Machine learning model lifecycle makes MLflow central in many organizations, but this alone is not enough to achieve operationally mature environments [3, 10].

Evaluation and Recommendations

The modularity and interoperability of MLflow are crucial elements for its widespread use. The modular architecture, composed of the Tracking , Projects , Models , and Registry components , is indeed adaptable to multiple frameworks and platforms. However, this flexibility still causes some complications in the automation of the pipelines themselves, that is, in the integration with orchestrators such as Kubeflow and Airflow. The fact that they are not natively supported results in increased operational and technical costs. For companies that want a fully automated CI/CD cycle, this feature is a disadvantage in choosing MLflow as the sole tool. On the other hand, its agnostic nature regarding cloud providers allows organizations to build their flow in any way they want [3, 10].

Experiment monitoring and model versioning are the two main features that make MLflow a powerful governance tool in the machine learning lifecycle [8, 15]. The ability to monitor parameters, metrics, code, and artifacts is crucial for ensuring model transparency and passing audits [3]. However, the lack of mechanisms to control the datasets themselves, such as data drift , is a disadvantage in the overall workflow . Organizations need to resort to other tools to ensure the scalability of projects using MLflow [10, 16]. The lack of fairness and explainability monitoring tools also impacts algorithmic accountability, which is crucial for sensitive areas such as finance and the public sector [3].

Implications of Openness and Regulatory Compliance

The fact that it is open source and agnostic to cloud providers allows MLflow to be used by startups to large organizations [3, 17]. Compatibility with frameworks such as TensorFlow and PyTorch is also an advantage [3, 17]). However, the openness of the platforms raises concerns about access control and data consent control for compliance with GDPR and HIPAA [10]. Massive customization of the environment increases complexity in the organization and in governance itself [17]. In organizations that do not need to comply with previous regulations, complexity still exists due to the need to install each version of each dependency. It is important to analyze the technical level of the companies and the scalability of each project before choosing MLflow as the software to be used [10].

MLflow contributes to improving the maturity of machine learning in the MLOps area, standardizing processes, automatically logging experiments, and controlling model versions [17, 8]. However, the lack of continuous monitoring and data drift control hinders the implementation of pipelines in areas where machine learning engineering practices are not well established. Compliance with strict regulations, such as GDPR and HIPAA, makes the control platform less attractive, especially for organizations with less technical capacity. Thus, it becomes much more difficult to obtain a return on investment, as the MLflow learning curve is steeper [10, 16].

Recommendations and Key Factors for Choice

Given the advantages and limitations of MLflow, this tool is best suited for data science teams in small and medium-sized enterprises that need speed in experimentation and prototyping. Its interface and open architecture make this software very appealing in less regulated companies where code scalability is fundamental [3]. However, in the case of large organizations and vertically integrated areas, MLflow is limited to prototyping and requires the complement of other more specialized tools. In this case, ClearML and ZenML are excellent alternatives [10].

In conclusion, the maturity level of companies is very important in choosing MLflow, as is the project typology. A cost-benefit analysis of environment customization versus its disadvantages and dependencies is necessary.

Conclusion

This study critically evaluated MLflow as an MLOps tool for managing the lifecycle of machine learning models . From defining the problem and addressing the challenges of traceability, reproducibility, automation, governance, and compliance , the study assessed how MLflow addresses these requirements, highlighting its strengths and weaknesses. The research answered the research question by indicating that MLflow is an excellent tool for tracking and versioning experiments and models, but still has limitations in terms of more advanced automation, native monitoring, and more detailed data and permission management.

Throughout the study, it was identified that the Tracking , Logging , Projects , and Models features allow MLflow to be used for experimenting and tracking runs, comparing runs and selecting models, logging models, encapsulating training artifacts in reproducible execution environments, building machine learning pipelines , and managing the transition between different production environments. Being open-source and available for various systems, it delivers broad technological independence to companies. Being modular and flexible in configuration, MLflow can meet the needs of different teams with varying levels of maturity, from beginners to the most experienced, especially focusing on less regulated processes.

A critical assessment of the limitations indicated that MLflow does not natively offer monitoring, fairness , data drift , and more detailed permissions. These functionalities depend on external tools and configurations, making operation more complex and hindering compliance for more sensitive domains such as healthcare and finance.
Therefore, it is clear that the ecosystem needs additions to meet governance, security, and scalability regulations, especially in highly regulated environments.

The experience gained from developing this work allowed me to learn more about the world of MLOps and to conclude the importance of constantly evaluating market technologies. Therefore, this work contributes to the development and adoption of MLflow, identifying the main challenges and market trends, and assisting in choosing the best MLOps tool. Finally, it was possible to understand the importance and relevance of governance, compliance , and innovation in the lifecycle of models and knowledge management in the context of digital transformation.

Contributions and Positioning of MLFlow
Regarding research in MLOps, this work contributes by broadening the understanding of the use of MLflow and its positioning compared to other options available on the market, highlighting the maturity that the tool provides. At the same time, it highlights market trends seeking more governance, automation, and security in the model lifecycle and raises current debates on the subject. Through the research carried out, MLflow is recommended for less regulated MLOps processes and for more dynamic companies, but which require complements according to their use cases.

Limitations of the Research and Future Research

Regarding the limitations of the investigation, it was demonstrated that there is a need to complement the study through validation and integration of MLflow with other tools, experimentation at scale and in a highly regulated domain, and validation in relation to other MLOps tools that address some requirements not yet covered. It is also important to mention the dependence on the breadth of the study sources and the lack of empirical testing to generate broader and more generalizable results. Therefore, future steps could include case studies in organizations of different sizes to validate the conclusions and suggestions of this work.

Figure 2 A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

References

1. Inuwa M (2022) Explaining MLOps using MLflow Tool. Data Science Blogathon. Analytics Vidhya https://www.analyticsvidhya.com/blog/2022/12/explaining-mlops-using-mlflow-tool/.
2. Symeonidis G, Kazakis A, Nerantzis E, Papakostas GA (2022) MLOps - Definitions, tools and challenges. arXiv 1-6.
3. Berberi L, Kozlov V, Alibabaei K, Esteban B (2023) MLflow and its usage. AI4EOSC Platform User Workshop, AI4EOSC https://indico.kit.edu/event/3845/contributions/14701/attachments/6987/11043/1.6_MLflow_and_its_usage.pdf.
4. Trencseni M (2021) Gettring Started with MLFlow https://bytepawn.com/getting-startedwith-mlflow.html.
5. Klein B (2021) AWS Orientação prescritiva. Amazon Web Services, Inc https://docs.aws.amazon.com/pt_br/prescriptive-guidance/latest/ml-operations-planning/ml-operations-planning.pdf.
6. Wazir S, Kashyap GS, Saxena P (2024) MLOps: A review. Department of Computer Science and Engineering, SEST, Jamia Hamdard, School of Data Science, University of North Carolina Charlotte

https://arxiv.org/pdf/2308.10908.

7. Fang Z, Yuan Y, Zhang J, Liu Y, Mu Y, et al. (2023) MLOps spanning whole machine learning life cycle: A survey https://arxiv.org/pdf/2304.07296.
8. Frenzel C, Nagaraja S, Romo S (2023) Evaluating your ML project with the MLOps checklist. Amazon Web Services, Inc https://docs.aws.amazon.com/pt_br/pre scriptive - guidance/latest/mlops - checklist/mlops - checklist.pdf.
9. Amorim BF, Souza ACC, Costa LM, Souza FCM (2023) An investigation of challenges in the machine learning lifecycle and the importance of MLOps: A survey. Computer on the Beach, XIV 379-385.
10. Safwan, Roadrup E, Synthesis SS (2024) MLOps Lifecycle Toolkit. Apress https://link.springer.com/content/pdf/10.1007/978-1-4842-9642-4.pdf.
11. Weng J (2020) Managing Your Machine Learning Experiments with MLFlow https://medium.com/data-science/managing-your-machine-learning-experiments-with-mlflow1cd6ee21996e.
12. Ruf P, Madan M, Reich C, Ould-Abdeslam D (2021) Demystifying MLOps and presenting a recipe for the selection of open-source tools. Applied Sciences 11: 8861.
13. Berberi L, Kozlov V, Nguyen G, Sainz-Brown Diaz J, Calatrava A, et al. (2025) Machine learning operations landscape: platforms and tools. Artificial Intelligence Review 58:1-37.
14. Dawson R, Sato D (2021) Evaluating MLOps Platforms. Thoughtworks https://www.thoughtworks.com/content/dam/thoughtworks/documents/ per/tw_whitepaper_guide_to_evaluating_mlops_platforms_2021.pdf.
15. Hanchuk D O, Semerikov SO (2025) Implementing MLOps practices for effective machine learning model deployment: A meta synthesis. CEUR Workshop Proceedings 3918: 329-337.
16. Hewage N, Meedeniya D (2022) Machine Learning Operations: A Survey on MLOps Tool Support. Arxiv 12.
17. Das SD, Bala PK (2024) What drives MLOps adoption? An analysis using the TOE framework. Journal of Decision Systems 33: 376-412.