An Acad Bras Cienc (2025) 97(2): e20240924 DOI 10.1590/0001-3765202520240924
Anais da Academia Brasileira de Ciências | *Annals of the Brazilian Academy of Sciences*
Printed ISSN 0001-3765 I Online ISSN 1678-2690
www.scielo.br/aabc | www.fb.com/aabcjournal

ENGINEERING SCIENCES

Make or buy strategy for Machine Learning Operations – MLOps

DIEGO NOGARE, ISMAR F. SILVEIRA, RENATO BANZAI & MAÍNA C. ALEXANDRE

Abstract: This research addresses the make or buy strategy for Machine Learning Operations (MLOps), exploring the decision between developing internally or purchasing computational solutions for Machine Learning projects. Considering factors such as cost, quality, technical expertise and strategic alignment, organizations face the challenge of balancing product complexity, core competencies and risk management. This research highlights the importance of understanding the needs of each project when analyzing existing offers to solve problems and maintain competitiveness in the market, offering a guide for drive and support your decision. Additionally, qualitative and quantitative reviews of MLFlow, Airflow, Kubeflow, Databricks, Dataiku, H2O, Amazon AWS, Microsoft Azure, and Google GCP tools are presented, which facilitate the life-cycle management of machine learning models. This research contributes to the understanding of the challenges and strategies involved in the effective implementation of MLOps projects.

Key words: Machine learning, Machine Learning Operations, make or buy, make or buy strategy, MLOps, model life-cycle.

INTRODUCTION

Several machine learning research studies focus on the mathematical theory related to how the machine learning models work, but only a few explore the deployment techniques of these models and tend to overlook aspects of computer Science and software engineering such as security, quality, performance and reliability. Projects involving ML cannot focus only on the specific business problems of the organization and the application domain that they build, they also need to pay attention to the architectural requirements of the proposed system, as well as involve the quality area that is responsible for the explainability of model decisions, data-centric decision-making, continuous monitoring, observability of computational operations, and fault tolerance, in addition to security and privacy issues (Washizaki et al. 2019).

MLOps helps bring machine learning models from research to the real world, creating a system for managing, deploying, and improving these models, which allows companies to utilize machine learning to solve business problems (Testi et al. 2022).

The decision for organizations on whether to build a machine learning (ML) solution in-house or buy an existing one is a complex one with potentially significant implications for factors like agility, software quality, and traceability (Fischer et al. 2020, Serban et al. 2020). While acquiring existing solutions can accelerate the integration of ML into products and services, developing solutions internally allows organizations to tailor systems to their specific needs and potentially reduce long-term maintenance issues (Gharibi et al. 2021, Haakman et al. 2021, Serban et al. 2020).

A variety of factors can influence this decision, including the availability of in-house expertise, data resources, the regulatory environment, and the level of required customization (Haakman et al. 2021) as well as current stage of the company, consideration of competitive advantages and maturity of the commercial tools (Huyen 2022). The increased complexity of ML systems compared to traditional software necessitates a full understanding of the trade-offs involved in building versus buying (Haakman et al. 2021, Njomou & Montreal 2022). A guide based on questions to understand your needs is proposed for support your decision, these answers could drive your moment to knows the best approach is make or buy.

This study provides a qualitative and quantitative analysis of the make-or-buy strategy for MLOps solutions. This is a complex discussion that should consider various factors explored in the specific sections Make or buy strategy and Discussion on this manuscript. The analysis considered technical characteristics of three market-leading tools in each of the following categories: open source, proprietary tools and cloud computing tools, creating a comprehensive analysis of nine tools.

Machine Learning Operations - MLOps

MLOps, or Machine Learning Operations, bridges the gap between machine learning models developed in research environments and their practical application in a corporate setting (Testi et al. 2022). MLOps systems effectively incorporate machine learning models into production, leading to increased adoption by businesses. A well-developed MLOps system with continuous training can result in more efficient and practical machine learning models (Symeonidis et al. 2022).

The goal of MLOps is to bring machine learning products into production, overcoming obstacles between development and operations teams and automating workflows. Achieving this requires modular system and workflow designs tailored to specific applications. This modularity simplifies development, deployment, and monitoring. While there are various ways to implement MLOps, emphasizing DevOps principles and modularity is crucial (Subramanya et al. 2022).

MLOps systems effectively incorporate machine learning models into production, leading to increased adoption by businesses. A well-developed MLOps system with continuous training can result in more efficient and practical machine learning models (Symeonidis et al. 2022).

A typical MLOps workflow includes the following steps (Calefato et al. 2022, Testi et al. 2022):

(a) Business Problem Understanding: This initial step focuses on comprehending the business problem and establishing clear objectives for the ML project (Testi et al. 2022). *(b) Data Acquisition*: This stage involves collecting raw data from various sources, which will be used to train and evaluate the ML model (Testi et al. 2022). *(c) ML Methodology*: This step focuses on selecting suitable ML algorithms and techniques to address the defined business problem based on the nature of the data and the desired outcome (Testi et al. 2022). *(d) ML Training and Testing*: In this stage, the selected ML model is trained on the prepared data, and its performance is rigorously evaluated using various metrics to ensure accuracy and reliability (Testi et al. 2022). *(e) Continuous Integration*: This phase integrates the trained ML model into the existing software development lifecycle, automating the process of merging code changes and ensuring consistency (Testi et al. 2022). *(f) Continuous Delivery*: This step focuses on automating the deployment of the trained ML model into the

production environment, ensuring a smooth and efficient transition (Testi et al. 2022). *(g) Continuous Training*: To maintain optimal performance, this stage involves retraining the ML model periodically or dynamically, adapting to new data and evolving patterns (Testi et al. 2022). *(h) Continuous Monitoring*: This critical phase continuously monitors the performance of the deployed ML model, detecting any deviations, and triggering alerts for potential issues (Testi et al. 2022). *(i) Explainability*: This stage aims to make the ML model's decision-making process understandable to humans, enhancing transparency and trust (Testi et al. 2022). *(j) Sustainability*: This final step involves considering the ethical and environmental implications of the ML model, ensuring responsible and sustainable AI practices (Testi et al. 2022).

Another significance step is the Machine Learning Pipeline, encompassing the entire ML lifecycle (Haakman et al. 2021). It needs to highlight the importance of data preparation, model training and testing, and performance comparison as key aspects of traditional pipelines (Carqueja et al. 2022).

The implementation of MLOps often involves using open-source tools and platforms, choosing the right tools for each task remains a challenge in MLOps (Mei et al. 2022). Flexibility and robustness are key considerations, each with its pros and cons (Carqueja et al. 2022). One approach for MLOps can be done combining opensource tools with enterprise solutions. Combining the best tools from different providers is feasible as most allow connections through APIs. (Kreuzberger et al. 2023).

Main phases of MLOps

The lifecycle of a MLOps project consists of three phases (Matsui & Goya 2022). It starts with the experimentation phase, which contains the steps of data collection and extraction, in addition to the development, which consists of training and testing the model, then it advances to the deployment phase where there are the stages that guarantee the practices of continuous integration and continuous delivery (CI/CD) and, finally, it ends in the operation phase, which involves the steps of publishing the artifacts in production and continuous monitoring of the model (Matsui & Goya 2022).

Deploying a ML model requires a series of steps that consider characteristics of training, performance evaluation, validation, retraining, publishing, monitoring, and unbiased data usage. Initially, a cultural change is required to enable technical change (Matsui & Goya 2022). The preparation of the model considers the characteristics of the data and the algorithms used. Pipeline automation and continuous monitoring are essential to the lifecycle of a ML model, allowing to track the model's performance and correct potential failures. It is important to consider the time it takes to complete each task, which depends on the iteration and the amount of data needed for its execution and conclusion. In addition, technical debt and data degradation can occur, bringing more challenges to this implementation compared to traditional software development (Matsui & Goya 2022).

Despite having their peculiarities such as data centricity, ML projects have great similarities with more traditional fields of computing and therefore share similar challenges. Based on this premise, the development of ML projects can reach more traditional areas such as software engineering, human-computer interaction and systems development, which have already experienced similar problems, and explore the solutions already solved by these areas (Paleyes et al. 2022).

Projects can be seen as pipelines that will take the best performing models to the production environment, but they are not limited to just that, they involve activities that support automation, integration and monitoring at all stages, including: training, integration, testing, publishing and infrastructure management (Testi et al. 2022). In many companies, the model and operation are taken to the production environment in an artisanal way, depending on human intervention, being moved manually without MLOps automation, which generates delays in the industrialization of ML methodologies (Testi et al. 2022). The continued reliance on manual deployment practices in many machine learning projects, rather than adopting MLOps principles, hinders the progression of ML proofs of concept to production and results in operational issues during the management of ML solutions (Kreuzberger et al. 2023).

An IDC report (Wiggers 2019) showed that a large part of Artificial Intelligence (AI) projects fails when going to production due to lack of experience, data bias, and high costs for professionals. Even with the growing number of reports explaining the challenges of deploying ML models, this is not a frequent topic in the academic literature, where the main research focuses are on the phase after the model is already published, seeking to present tasks related to Explainable ML or software engineering for ML (Paleyes et al. 2022).

Make or buy strategy

The decision to make or buy solutions is not confined to physical products or software but extends to strategic functions within organizations as well (Suliantoro et al. 2022).

Market dynamics, technological changes, and industry trends also influence the decision between making (building in-house) or buying externally. Therefore, the challenge lies in comprehensively and strategically evaluating which approach best meets the needs and objectives of the company at a given time, taking into account the various aspects involved in the choice between making or buying. (Petrin 2024) The analysis is complex, as it directly impacts the company's cost structure, flexibility, quality control, and strategic positioning in the market (Channappa et al. 2016). Not to mention the complexity inherent in determining the most cost-effective and efficient approach to acquire systems or components, deciding between building in-house or sourcing software components externally (Zhao et al. 1999).

In the context of AI and new technologies, the traditional decision to make or buy is complicated by factors such as asset specificity, hold-up risks, and the dynamic nature of the technology (Petrin 2024). When there is a significant power asymmetry between companies, the weaker party may face increased vulnerability to hold-up risks, especially if it lacks the financial resources to integrate vertically or switch suppliers easily (Petrin 2024). Hold-up is a situation in which one party to a transaction has significant bargaining power over the other, due to specific circumstances such as the specificity of the assets involved, being a central risk in decisions to make or buy (Petrin 2024).

Early-stage companies may opt for off-the-shelf products to prove business value, but with time and increased knowledge, the investment in third-party tools should be re-evaluated. This strategic assessment can significantly impact the cost, development time, and quality of the final system, so it is critical to conduct this analysis early in the project to guide the next development steps (Zhao et al. 1999).

Take the decision for make or buy

Assumption that MLOps is the subject from the analysis for Make or Buy strategy, we suggested this guide to follow during your decision. Channappa et al. (2016) said by utilizing decision-making models, break-even analysis, and considering exceptions, organizations can navigate these challenges and remain competitive in the market by collaborating with strategic partners and assessing the long-term implications of the make or buy decision.

Figure 1 advises that if MLOps is considered a commodity, that it just means to expedite the movement of models from development to production and customer delivery, then a "buy" strategy is recommended. This involves acquiring a proprietary offering or adopting an existing open-source solution, thereby allowing resources to be concentrated on core business objectives, assuming the availability of mature commercial tools. As Huyen (2022) said, the maturity of the commercial tools also plays a role in this strategy, as the lack of a mature tool may force the company to build it in-house, but this assessment can be revised as the commercial tools mature. Otherwise, if MLOps is core for your business, and it means your business has some strategical and consideration of competitive advantages needs to reach your goals for MLOps, ensure that you have the availability of experts in-house and consider make itself inside the company.

To drive you during this assessment, answer the following questions with 1 for yes or 0 for no, sum the results for each axis and plot the decision on Figure 1.

a) Is MLOps the main objective for your business?
b) Do you have professionals with expertise in MLOps in the company?
c) Do the company have expertise with MLOps process?
d) The tools analyzed don't reach the needs for the company?
e) Your business has specific regulatory or law requirements for the machine learning models?

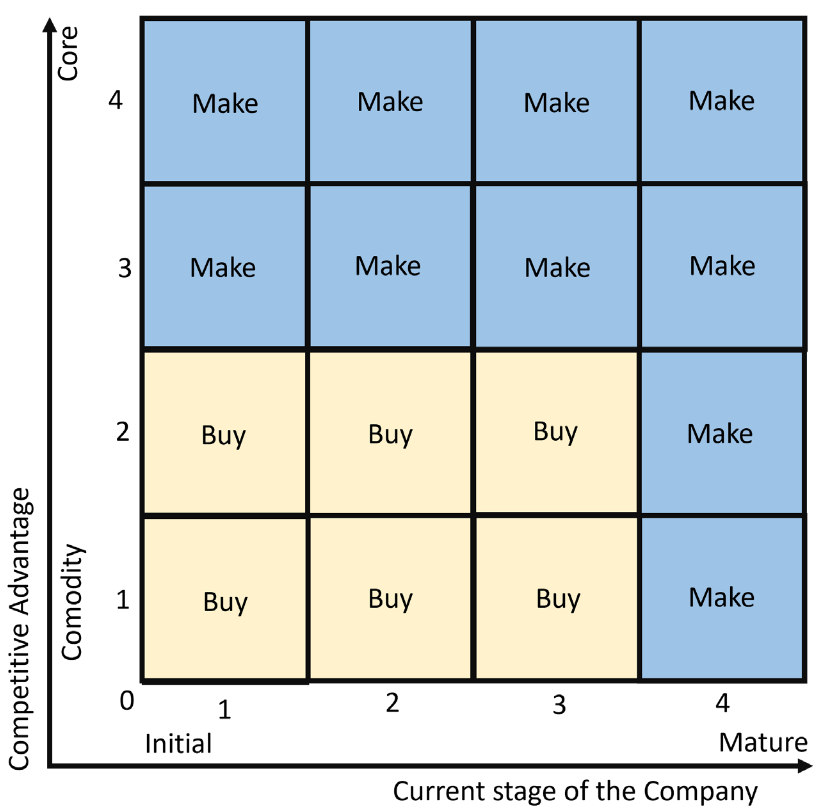

Figure 1. Strategy to decide between Make or Buy solution. Source: The authors, 2024.

f) Your business needs some level for customization?

g) Do you consider MLOps bring competitive advantages for your business?

For the X axis, Current stage of the Company, use this equation:

$$CSC = a + b + c + d$$

For the Y axis, Competitive Advantage, use this equation:

$$CA = a + c + f + g$$

Plot the result for *CSC* in the X axis and *CA* in Y axis. This answer will guide your decision.

This same strategy should be evaluated based on the maturity of your development team and their MLOps knowledge. When a team is initially learning about MLOps, purchasing a third-party solution is preferable to internal development. However, if the team is proficient in machine learning operations and has the necessary expertise to comfortably develop the solution within the company, then in-house development is a viable option.

Complementary to the make or buy possibilities, there is a third way that is a mix of make and buy, combining using open source, proprietary, cloud and internal development. It's the most flexible solution and shed light to the high level of required customization and regulatory/law environment, and to do this your company should be mature and consider MLOps core for the current stage of the company. Balancing product complexity, core competencies, market dynamics, and risk management is crucial to making informed decisions (Petrin 2024).

Tools can be categorized as either solution accelerators or competitive differentiators. Secondary tools can be purchased without harming the business, while those that are the main point of what will be delivered require full control of the company.

By evaluating these aspects comprehensively and in line with organizational objectives, it is possible to make informed decisions that optimize system performance and meet the specific needs of the business (Petrin 2024).

Related works

Regardless of the industry, the fundamental challenge lies in striking a balance between the strategic advantages of tailored software solutions and the potential benefits of leveraging existing, commercially available software products, all while addressing the complexities of software integration and potential trade-offs in functionality and control.

This section will discuss existing research on the make or buy decision, examining its applicability in some industrial contexts and highlighting factors influencing this decision, mainly in the software field.

In the construction industry, for instance, companies grapple with this decision when considering digital procurement tools, often weighing factors such as the need for customized solutions versus the cost and complexity of internal development (Shi et al. 2020). The healthcare sector faces similar considerations when evaluating AI software for integration into clinical workflows (Kim et al. 2023). Even within software development itself, the build or buy dilemma emerges when choosing tools and frameworks for tasks like large-scale agile development, with architects needing to balance architectural requirements and development efficiency (Shortridge & Dykstra 2023).

One industry where the "make versus buy" decision frequently arises is construction. Traditionally, the construction industry has relied heavily on personal relationships and direct negotiations when procuring services, particularly for specialized work such as custom-building projects or renovations. However, with the increasing digitalization of business processes, there is growing interest in adopting digital procurement methods in construction. Welsh & Martinez (2023) discuss the potential benefits of digital procurement, including enhanced transparency, efficiency, and sustainability in managing the supply chain. Them also highlight the importance of integrating systems such as Enterprise Resource Planning (ERP), Supplier Relationship Management (SRM), and Building Information Modeling (BIM) to leverage these advantages fully. Specifically, the authors explore how SRM systems can streamline the verification of "green certification" for construction materials, contributing to sustainability efforts and simplifying procurement processes. Despite the potential advantages of digital procurement, its adoption in the construction industry, particularly among small to medium-sized enterprises (SMEs), remains limited. Challenges include skepticism towards new technologies, lack of resources for implementing complex systems, and legal restrictions in some regions, such as Austria, that hinder early contractor involvement and limit the use of framework agreements in public sector projects (Welsh & Martinez 2023).

In another case, Suliantoro et al. (2022) discusses the evolving role of procurement in public hospitals, advocating for its development from a transactional function to a strategic one. Them uses a Supply Management Maturity Model to assess the maturity level of procurement organizations, evaluating them based on characteristics such as procurement planning, organizational structure, human resources, and control mechanisms. The findings suggest that many procurement units, although adopting modern techniques, need to align their procurement strategies with the overall organizational strategy. This highlights the importance of continuous improvement and development of the procurement function to support the strategic objectives of the organization (Suliantoro et al. 2022).

Beyond the construction and health industry, the decision to build a solution or buying a pre-existing one also applies to developing software systems. Uludağ & Matthes (2020) explores the challenges and best practices in large-scale agile development, focusing on the role of enterprise and solution architects in navigating the complexities of these projects. With the increasing complexity and speed of software development, companies need efficient strategies to manage architectural decisions, particularly when scaling agile methodologies to larger projects with multiple teams. The authors emphasize the importance of communication and feedback between enterprise architects and agile teams to ensure alignment between the overall architectural vision and the practical implementation of software systems. It highlights the need for architects to adapt to the agile environment, moving away from traditional "PowerPoint Architect" approaches characterized by excessive documentation and limited technical involvement. Instead, architects should embrace a more hands-on approach, actively engaging with development teams, providing technical support, and adapting to the iterative nature of agile development (Uludağ & Matthes 2020).

Tools evaluation

In order to bring more evidence and support to the proposed discussion, this research presents a comprehensive comparison between the MLOps platforms available in the market, categorized into three main types: open-source tools, proprietary tools, and cloud computing tools. Each of these categories has its own advantages, disadvantages, and ideal use cases. The choice between them can have significant implications for the efficiency, effectiveness, and scalability of ML projects.

Open-source tools provide flexibility to tailor solutions to specific needs, promote transparency for increased understanding and trust, and benefit from active communities that offer support and facilitate continuous improvement (Serban et al. 2020).

Proprietary tools offer robustness, dedicated support from the provider, and seamless system integration, which are crucial aspects for businesses seeking reliable and comprehensive solutions (Njomou & Montreal 2022).

Cloud computing offers flexibility by supporting multiple machine learning frameworks, which prevents developers from being tied to a single framework (Chen et al. 2022). Cloud computing also offers scalability through features like a built-in scaling system that can scale up to process requests and scale down after processing (Jauro et al. 2020).

Scope of the evaluation

The comparisons made in this study establish a baseline for decision-making between making or buying. By understanding the functionalities, strengths, and limitations of each tool, one can move forward in making an individual decision about the chosen platform and developing more efficient and effective MLOps practices for their needs. In addition, this analysis will allow you to make more efficient decisions when selecting the MLOps platform that best suits your specific needs (buy) or building in-house (make).

Heuristic Evaluation

The comparative analysis was made in stages. Initially the names of the three main products or services in each category were defined. Then, a qualitative and quantitative evaluation of its most relevant characteristics was written. Finally, and in our understanding, the most important aspect of this segment of the research, a comparison of functionalities and technical capabilities existing in each tool or service was made. This comparison allowed us to understand the strengths and evolution points of each product analyzed, and to conclude our study. The objective of this comparison is checking all the functionalities covered in the state of art for MLOps process.

DISCUSSION

As challenges presented on Nogare et al. (2022), and forward in enhancement of this platform, this present study used their needs to evaluate features and resources from MLOps that fits the minimum viable product for them.

Three representatives were selected for each category, and the choice was based on criteria for the use of these solutions in the market.

Qualitative analysis of open-source tools

(a) MLFlow: It is a platform that allows users to manage the entire machine learning model lifecycle, including experimentation, reproducibility, and model deployment. It provides a unified interface for managing parameters, metrics, and artifacts, making it easy to compare and share experiment results. MLflow could benefit from improved integration withother components of the MLOps ecosystem, such as training and deployment pipelines. In addition, improvements in scalability and support for different storage systems would be welcome. *Evaluated on July/2024, version V2.15.1.*

(b) Apache Airflow: It is a tool that allows users to develop, schedule and monitor complex workflows. Although it is not a specific MLOps tool, it is often used together with other tools to orchestrate ML and data pipelines. Its main functionality lies in the ability to define dynamic and complex workflows through code. However, Airflow could improve in terms of scalability and performance, especially for large-scale systems. In addition, the user interface and documentation could be more user-friendly to make it easier to use. *Evaluated on July/2024, version 2.10.0*

(c) Kubeflow: It's a platform that makes ML deployments on Kubernetes simple, portable, and scalable. It provides an end-to-end environment that includes model training, hyperparameter tuning, and model consumption, allowing users to maintain consistency between development and production environments. However, it requires Kubernetes to operate, which may be a barrier for some teams. The evolution of Kubeflow should focus on simplifying setup and improving documentation to facilitate adoption by teams of different sizes and experience levels. *Evaluated on July/2024, version V1.9.*

Qualitative analysis of proprietary tools

(d) Databricks: It is a cloud-based data analytics and AI platform that provides a unified environment for data science and data engineering. Databricks is built around Apache Spark, a large-scale data processing framework, and offers functionality for ML lifecycle management, including experimentation, collaboration between individuals on projects, and model deployment. In other hand, Databricks could be better in terms of cost, especially for small teams or individual projects. In addition, integration with other tools and ecosystems could be simplified to facilitate adoption. *Evaluated on July/2024, version is not relevant for this.*

(e) Dataiku: It is a proprietary platform that enables teams to develop, deploy, and monitor data science and ML solutions. Offers an intuitive and collaborative user interface for data scientists, data analysts, and data engineers to work together on the same project. The platform supports a wide range of third-party tools and frameworks, as well as allows you to deploy models to production with one click. While Dataiku is a powerful tool, it could improve the user experience, especially for beginners. In addition, the documentation and tutorials could be more comprehensive to help users take advantage of the platform's full potential. *Evaluated on July/2024, version is not relevant for this.*

(f) H2O: It is a proprietary ML platform that allows the construction of large-scale and high-performance models, in addition to also offering a free version. One of the main products is H2O Driverless AI, which is a complete Automated Machine Learning (AutoML) solution. It offers features such as feature engineering, model selection, and hyperparameter tuning, all automatic, allowing users to develop high-quality ML models efficiently. While H2O.AI is popular for its efficiency and scalability, it could improve its user interface and documentation. In addition, the community around H2O.AI could be more active in providing support and sharing knowledge. *Evaluated on July/2024, version is not relevant for this.*

Qualitative analysis of cloud computing

(g) Amazon AWS: It is one of the leaders in global cloud computing services and offers a full range of tools for MLOps. AWS SageMaker is its fully managed machine learning platform that enables data scientists to quickly build, train, and deploy ML models. It supports the entire ML pipeline, from data preparation and hyperparameter tuning to model deployment and monitoring. In addition, AWS also offers AWS Lambda for serverless activities and AWS

Glue for Extract, Transform, Load (ETL) allowing the construction of robust and scalable data pipelines. However, the billing process can be complex for beginners and prices can be high and vary depending on usage. In addition, the initial learning curve can be steep for beginners, due to fragmented and difficult to locate documentation.*Evaluated on July/2024, version is not relevant for this.*

(h) Microsoft Azure: It is another great cloud computing platform that offers a variety of services for MLOps. Azure Machine Learning (AzureML) is a service that enables the building, training, and deployment of ML models at scale, using cloud or edge computing. Supports a wide range of traditional and deep learning ML algorithms. In addition, Azure offers Azure Functions for serverless computers and Azure Data Factory for ETL, enabling the creation of complex data pipelines.In turn, the documentation could be more comprehensive and user-friendly to facilitate adoption, in addition to having a difficult integration with non-Microsoft solutions. Beginners to the Azure platform may have difficulty solving trivial challenges. *Evaluated on July/2024, version is not relevant for this.*

(i) Google GCP: It is yet another cloud computing platform that offers a variety of services for MLOps. VertexAI is a unified environment for developing ML projects, from prototyping to deployment. It supports a variety of ML frameworks, including TensorFlow, Keras, and Scikit-learn. In addition, GCP offers Google Cloud Functions for serverless computing and Dataflow for real-time and batch data processing, allowing you to build efficient and scalable data pipelines. In this case, adoption may be lower compared to AWS and Azure because it offers less variety of tool options, in addition to some management and monitoring tools that we consider less mature. *Evaluated on July/2024, version is not relevant for this.*

Quantitative analysis of technical capabilities

To establish a quantitative comparisonbasis of the technical capabilities of each of these analyzed tools and the existing elements that will direct the most appropriate solution for specific MLOps needs, this session compares the technical capabilities required for ML projects were declared, and which vendor meets that requirement.

The analyses were divided into seven logical groups, namely: Experimentation, Implementation, Monitoring, Project Management and friendly user interface, Data Manipulation, Development experience and Security and Data Access.

Each group has a series of functionalities that we checked to see if they existed or not. If the functionality existed, the value 1 was entered in the spreadsheet; if the functionality did not exist, the value 0 was entered. We calculated the average for each tool analyzed in that group and presented the percentage of completeness that the tool achieved. See the functionalities analyzed in each logical segment and chart for the visual comparison.

Figure 2 check these features for the tools analyzed, as follows: Canary Deploy, Blue-green Deploy, A/B test, Champion Challenge for model, fast and simple shift models, automated tests, report for test result models. In this evaluation, the most complete tools are Databricks, Dataiku, AWS, Azure and GCP.

In the Figure 3, is Batch, Streaming, Sync and Async processing, execution and re-execution by point-click, metrics for evaluate model visualization in timeline, report for all deployed models, retraining with new data epoch, elasticity in computational power, GPU TPU and Transformers architecture, support for huge

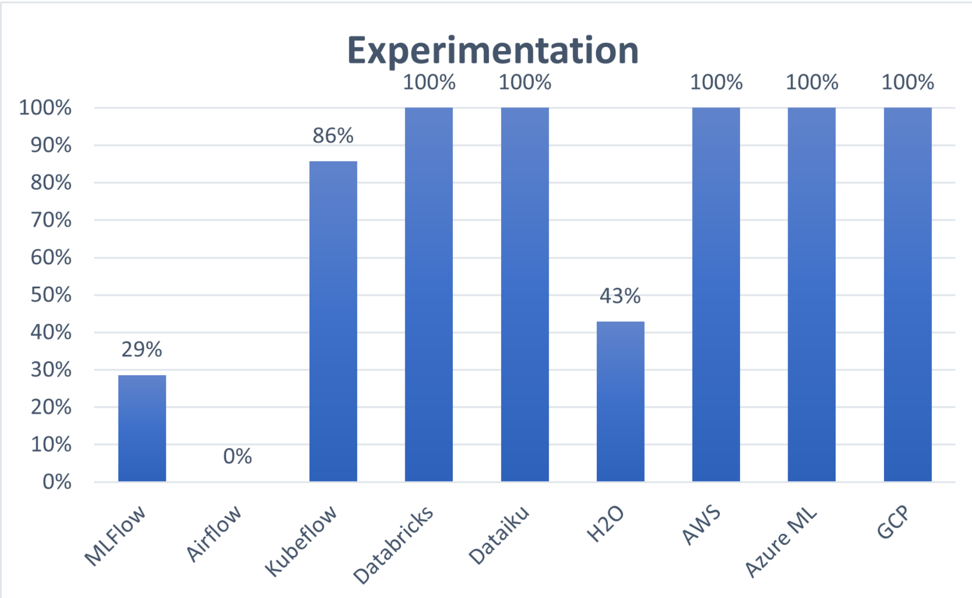

Figure 2. Group of Experimentation. **Source:** The authors, 2024.

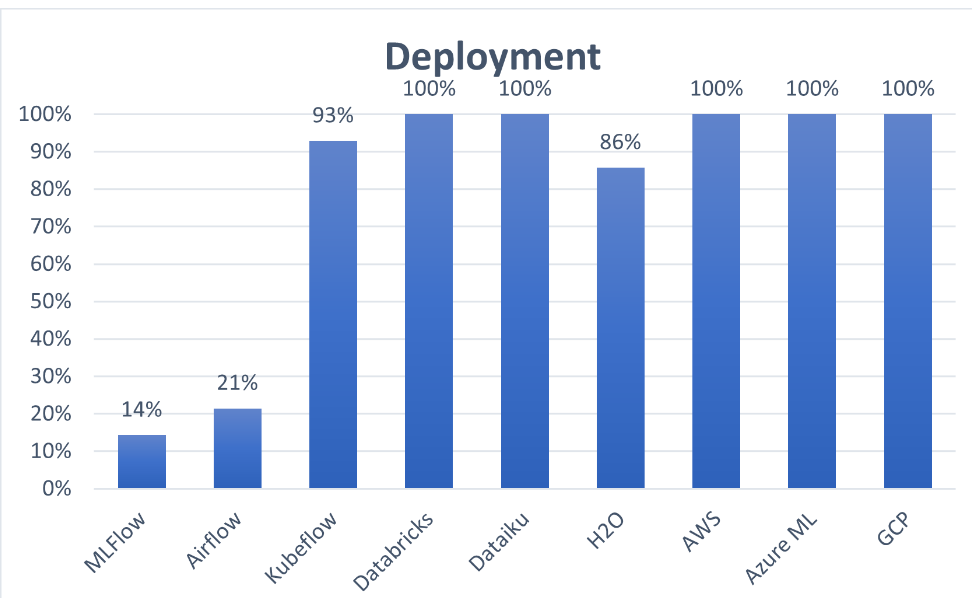

Figure 3. Group of Deployment. **Source:** The authors, 2024.

models, easily error analysis for bug fix, easily model promotion among development, quality and production environment, threshold alert based for model execution. In this segment, the most complete are Databricks, Dataiku, AWS, Azure and GCP.

Figure 4 covers report for explainable model, custom uses library, model, data and concept drift, log for every internal call, easily log extraction, visualization and action, administrative report for stakeholders. In this group, the most complete are Databricks, Dataiku, H2O, AWS, Azure and GCP.

Figure 5 easily environment for Citizen Data Scientist and Stakeholders, unique user interface, automated workflows, unique touch-point for documentation, logic segmentation among project, experiment and model, place for search models, follow-up for model's executions, alerts by email, lock-in for specific

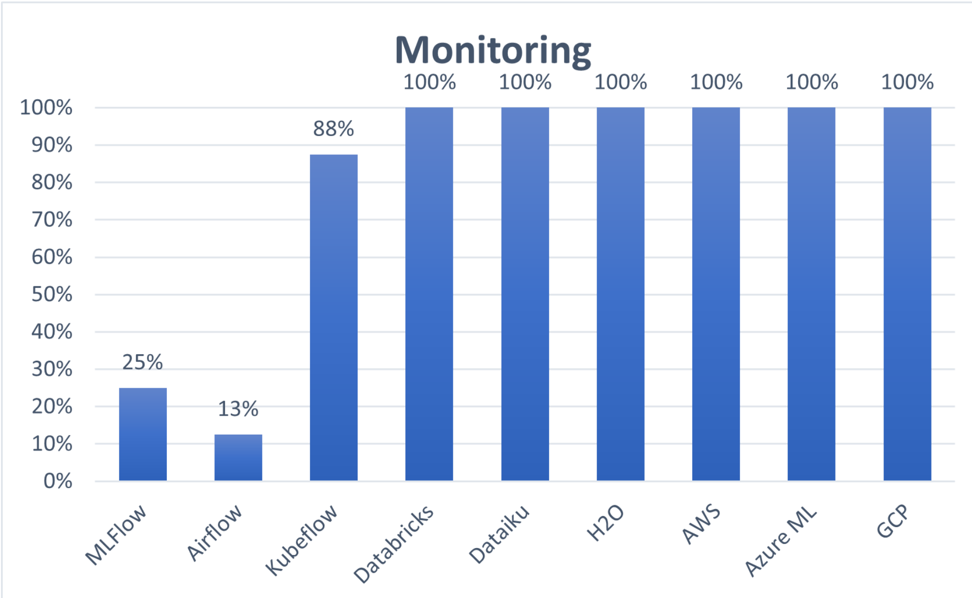

Figure 4. Group of Monitoring. **Source:** The authors, 2024.

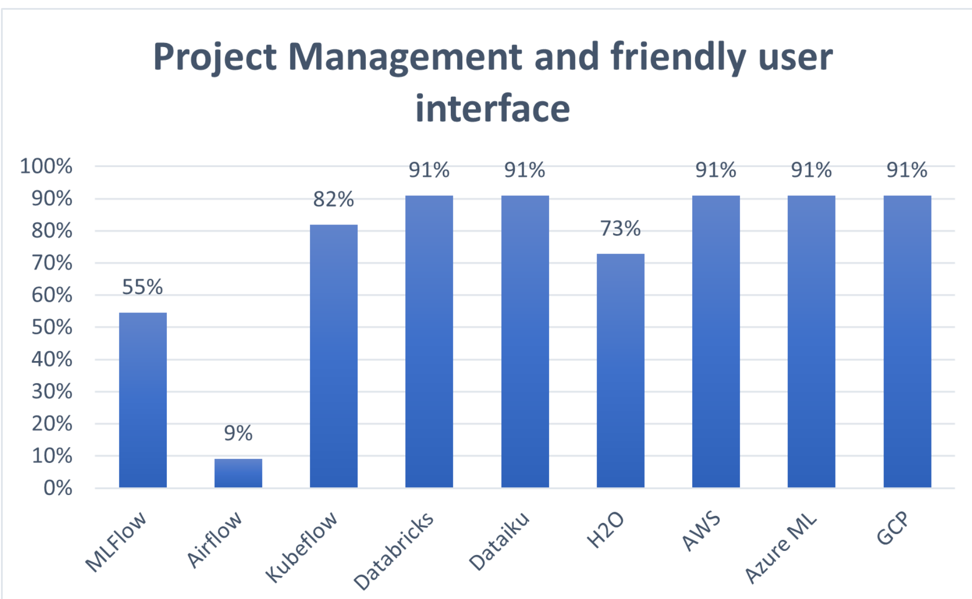

Figure 5. Group of Project management and friendly user interface. **Source:** The authors, 2024.

license or proprietary tool and execution and re-execution by point-click. In this assessment, the most complete are Databricks, Dataiku, AWS, Azure and GCP.

Figure 6 covers normalization, data aggregation, data augmentation, data visualization to graphics and dashboards, write data in feature store, search tables in feature store, search features in feature store, search data in feature store, search metadata in feature store, quality and statistical analysis for data. In this analysis, the most complete are Airflow, Databricks, Dataiku, AWS, Azure and GCP.

In Figure 7, the tools check Jupyter Notebook IDE, Python and R support, install custom packages for dependencies and requirements, elasticity in computational power, custom metrics for evaluate model, fine tuning hyper-parameter, feature importance report, automated machine learning (AutoML), bring your own serialized

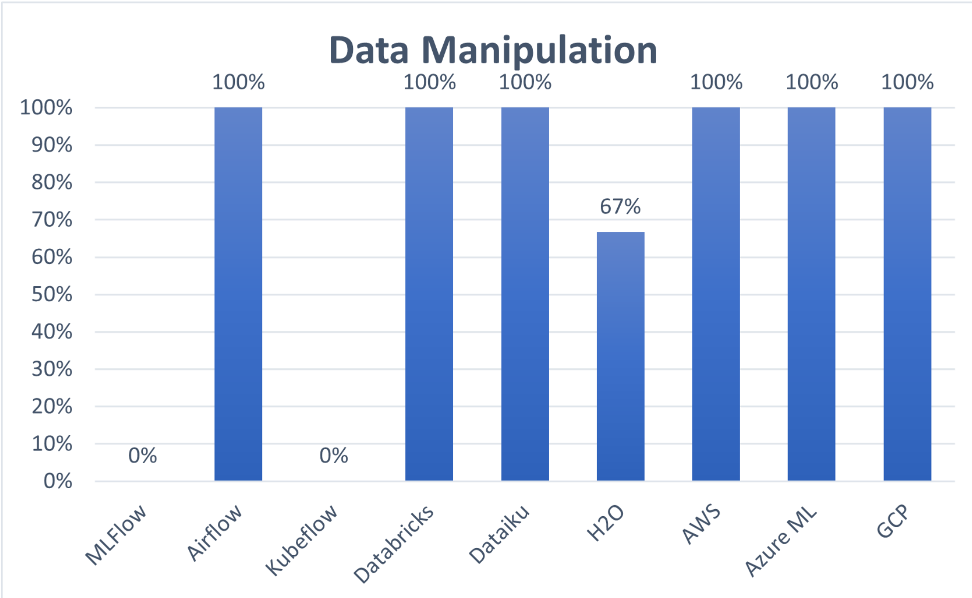

Figure 6. Group of Data manipulation. **Source:** The authors, 2024.

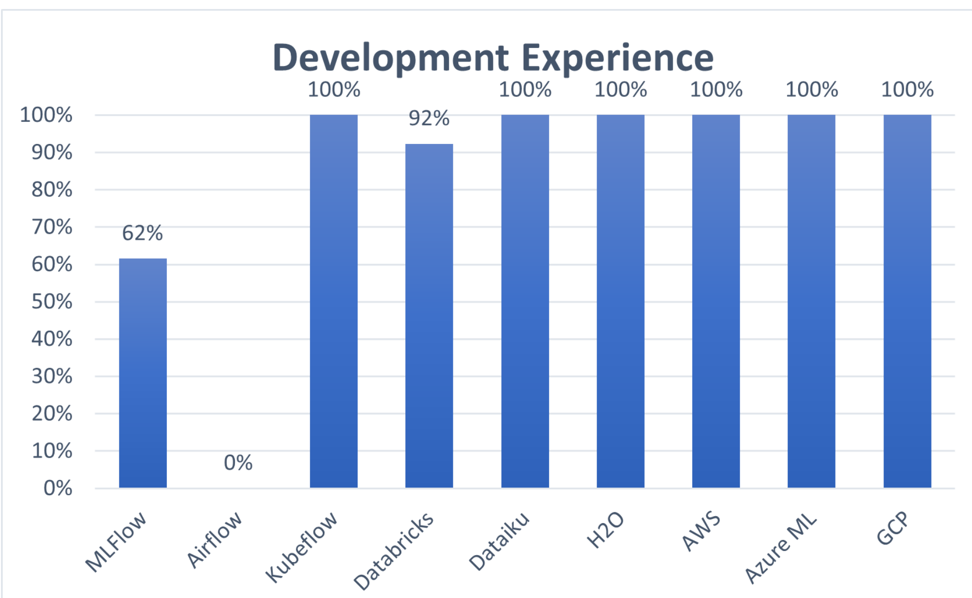

Figure 7. Group of Development experience. **Source:** The authors, 2024.

model, cross-validation with point-click and git/github integration. In this evaluation, the most complete tools are Kubeflow, Dataiku, H2O, AWS, Azure and GCP.

In Figure 8, is possible to see the Integration with Azure AD, Integration through Single Sign-On SSO, Access data in CDP, Compatibility with Data Mesh architecture, Access local file, Integration with cloud providers as AWS (Amazon), GCP (Google), OCI (Oracle) and Azure (Microsoft). In this final group, the most complete tools are Airflow, Dataiku, AWS, Azure and GCP.

Finally, the complete qualitative analysis can be seen in the Figure 9. It represents the comparison overall features and tools. It's based on the Table I, Table II and Table III, that represents the percentage of covering tools in the group. The percentage means the tools and features covered by the provider in that group.

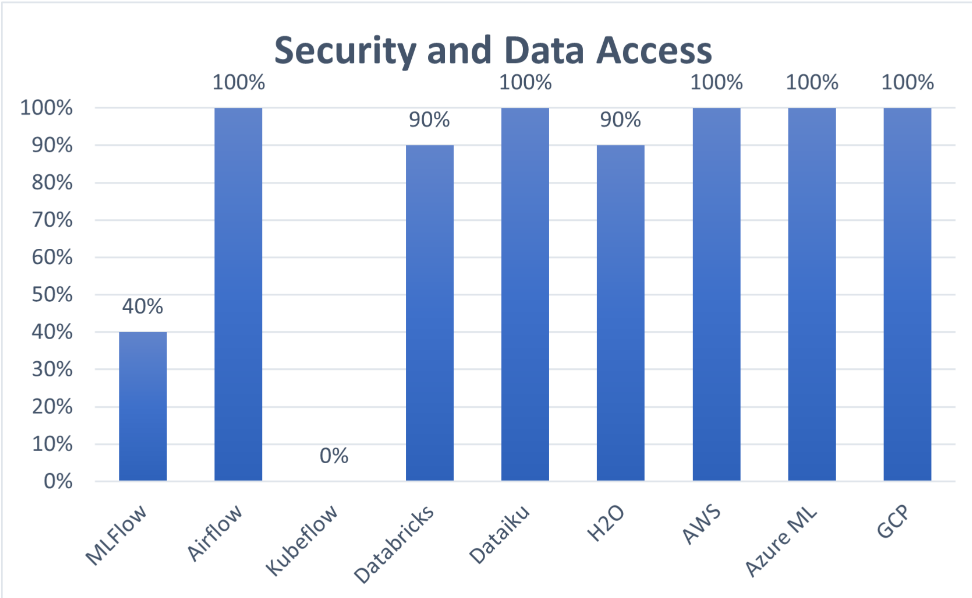

Figure 8. Group of Security and Data access. **Source:** The authors, 2024.

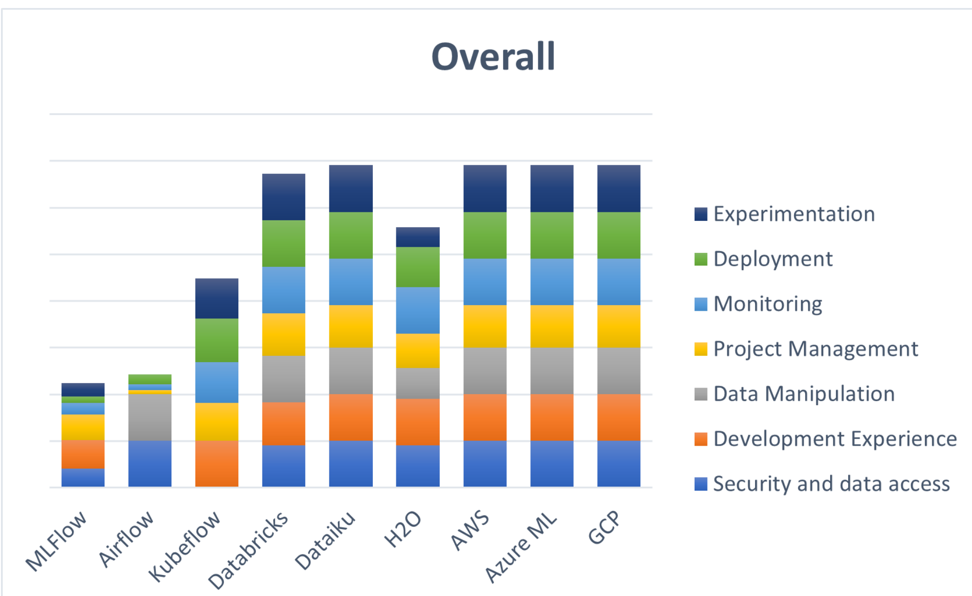

Figure 9. Global features shared across the group. **Source:** The authors, 2024.

Based on the results of the assessment, we understand that integrating MLflow with Airflow or Kubeflow leverages the strengths of each tool in managing and orchestrating machine learning workflows. MLflow, with its centralized registry, facilitates comprehensive experiment tracking and model versioning, addressing a critical need in the lifecycle of machine learning models. Airflow excels in workflow orchestration, enabling the scheduling, monitoring, and execution of tasks within complex pipelines, although it lacks a native model registry like MLflow. By integrating MLflow with Airflow, users can harness Airflow's orchestration capabilities to manage the entire workflow, while MLflow tracks experiments and records parameters, metrics, and artifacts associated with trained models. Conversely, Kubeflow, which is inherently designed for deployment and orchestration within Kubernetes environments, offers

Table I. **Technical capabilities provided for each product or service in Open-Source tools.**

	MLFlow	Airflow	Kubeflow
Security and data access	40%	100%	0%
Development Experience	62%	0%	100%
Data Manipulation	0%	100%	0%
Project Management	55%	9%	82%
Monitoring	25%	13%	88%
Deployment	14%	21%	93%
Experimentation	29%	0%	86%

Source: The authors, 2024.

Table II. **Technical capabilities provided for each product or service in proprietary tools.**

	Databricks	Dataiku	H2O
Security and data access	90%	100%	90%
Development Experience	92%	100%	100%
Data Manipulation	100%	100%	67%
Project Management	91%	91%	73%
Monitoring	100%	100%	100%
Deployment	100%	100%	86%
Experimentation	100%	100%	43%

Source: The authors, 2024.

advanced services such as interactive notebooks, TensorFlow model training, and ML pipelines within Docker containers. Integrating MLflow

Table III. **Technical capabilities provided for each product or service in cloud computing.**

	AWS	Azure ML	GCP
Security and data access	100%	100%	100%
Development Experience	100%	100%	100%
Data Manipulation	100%	100%	100%
Project Management	91%	91%	91%
Monitoring	100%	100%	100%
Deployment	100%	100%	100%
Experimentation	100%	100%	100%

Source: The authors, 2024.

with Kubeflow combines the robust experiment tracking and model management of MLflow with the scalability, flexibility, and deployment advantages provided by Kubernetes, thereby creating a comprehensive and scalable solution for managing machine learning workflows in diverse computational environments.

In the second group of tools analyzed, Databricks, built on Apache Spark, demonstrates significant efficacy in big data environments by offering substantial scalability and flexibility, particularly in cloud infrastructures such as Azure and AWS, while simultaneously facilitating collaboration among data science teams through collaborative notebooks and automated pipelines; however, its steep learning curve necessitates advanced technical knowledge, and the associated costs can be prohibitive, especially for small and medium-sized enterprises. Conversely, Dataiku provides a more accessible option for a broader range of users, including those with limited technical expertise, by offering an intuitive drag-and-drop interface that enables the visual creation of machine learning pipelines and seamless integration with various data sources and open-source tools, though this ease of use may constrain scalability and customization, making it less suitable for large-scale operations or those with specific requirements. Meanwhile, H2O.AI excels in advanced automation and deep learning support, with tools like H2O Driverless AI (tool for AutoML) that streamline model building and optimization, yet its complexity may pose challenges for teams lacking high technical proficiency, and its “black box” nature can limit model control; while potentially more cost-effective than Databricks, H2O.AI still demands a significant investment.

The third group, Cloud Computing, as follows the result presented in Table III and as well in the Figure 9, we understand it´s not make sense compare tools, functionality and resources in this group because all of three provide has equals resources with just a little bit customization and kind of implementation.

CONCLUSIONS

The approach to decide make or buy consist in many factors, and shed light to this challenge, we proposed a simplify chart with questions about your business to drive you during your decision to make or buy. This process helps you to decide best approach for your needs. Complementary, for buy strategy, it is possible to observe that each tool, whether open source, proprietary, or cloud computing, has a unique set of features and functionalities. However, no single tool is complete and comprehensive or solves all the existing steps in a unified MLOps platform. Open-source tools focus on specific niches, which makes it infeasible to use them in isolation without other tools in a complete scenario. Proprietary solutions have greater flexibility and a wide range of capabilities, which can be a viable option when considering the possibility of balanced deliveries with the associated costs, however the price and contract factor must be considered to assess the feasibility of the project. Finally, the cloud computing environment allows greater flexibility in the combination of parts, services, and tools, which enables a wide range of possibilities for integration and project delivery. This is significant for exploring combined tools and services to meet the needs of a project. By understanding the nuances of these analyzed tools, it was possible to conclude that, even using the best tools and services available today individually, there is no unison completeness that can be used separately. However, as each ML project needs are unique, it is appropriate to understand the positive and improvement points of each solution analyzed to identify the one that best meets your needs. Today, based on the result obtained in this research, we can argue that each of these tools analyzed has a set of resources and capabilities, which makes them suitable for different needs and scenarios.

As seen in related works, in the example that illustrates the construction industry, the software is important but not fundamental to the core business of that example, making its use a commodity and not the deliverable product that makes this industry profitable. In the case of software brought as an example, the solution was to develop it internally and the authors of the work highlight mainly the challenge inherent to the software architecture and interaction between people so that the project works and solves the proposed problem. Finally, in the example of the hospital industry, the approach presented recommends a hybrid approach between starting with ready-made software and extending the solution by expanding the development to adapt to the needs of the business, ensuring the regulation and legal demands that are inherent to the industry.

Acknowledgments

We would like to express our deep gratitude to the Instituto Presbiteriano Mackenzie and Universidade de São Paulo for the support provided during the development of this research. The financial support, infrastructure, and resources provided were fundamental to the success of this work. We also want to extend our thanks to the Instituto de Ciência e Tecnologia Itaú (ICTi) for their continued encouragement and investment in Brazilian science. We firmly believe in the importance of their contribution to the advancement of knowledge and research in our country. Any opinions, findings, and conclusions expressed in this manuscript are those of the authors and do not necessarily reflect the views, official policies or position from Universidade Presbiteriana Mackenzie, Universidade de São Paulo or Itaú Unibanco.

REFERENCES

CALEFATO F, LANUBILE F & QUARANTA L. 2022. A Preliminary Investigation of MLOps Practices in GitHub. International Symposium on Empirical Software Engineering and Measurement, 283-288 p.

CARQUEJA A, CABRAL B, FERNANDES JP & LOURENÇO N. 2022. On the Democratization of Machine Learning Pipelines. Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence. SSCI 2022, 455-462 p.

CHANNAPPA SD, PICHUKUPPAN A & VISWANATHAN S. 2016. Make or Buy Decision for Systems–An ESO perspective. In INCOSE International Symposium, Vol. 26. Wiley Online Library, p. 25-39.

CHEN KH AT AL. 2022. Apache submarine: A unified machine learning platform made simple. EuroMLSys 2022 - Proceedings of the 2nd European Workshop on Machine Learning and Systems, 101-108 p.

FISCHER L, EHRLINGER L, GEIST V, RAMLER R, SOBIEZKY F, ZELLINGER W, BRUNNER D, KUMAR M & MOSER B. 2020. AI system engineering - key challenges and lessons learned. Mach Learn Knowl 3: 56-83.

GHARIBI G, WALUNJ V, NEKADI R, MARRI R & LEE Y. 2021. Automated end-to-end management of the modeling lifecycle in deep learning. Empir Softw Eng 26 1-33.

HAAKMAN M, CRUZ L, HUIJGENS H & DEURSEN AV. 2021. AI lifecycle models need to be revised: An exploratory study in Fintech. Empir Softw Eng 26: 95.

HUYEN C. 2022. Designing machine learning systems. " O'Reilly Media, Inc.".

JAURO F, CHIROMA H, GITAL AY, ALMUTAIRI M, ABDULHAMID SM & ABAWAJY JH. 2020. Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend. Appl Soft Comput 96: 106582.

KIM JE, BOAG W, GULAMALI F, HASAN A, HOGG HDF, LIFSON M, MILLIGAN D, PATEL M, RAJI ID & SEHGAL A. 2023. Organizational governance of emerging technologies: AI adoption in healthcare. In proceedings of the 2023 ACM conference on fairness, accountability, and transparency. p. 1396-1417.

KREUZBERGER D, KÜHL N & HIRSCHLS. 2023. Machine learning operations (MLOps): Overview, definition, and architecture. IEEE Access 11: 31866-31879.

MATSUI BM & GOYA DH. 2022. MLOps: five steps to guide its effective implementation. In: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, p. 33-34.

MEI S, LIU C, WANG Q & SU H. 2022. Model Provenance Management in MLOps Pipeline. ACM International Conference Proceeding Series, 45-50 p.

NJOMOU AT & MONTREAL P. 2022. On the Challenges of Migrating to Machine Learning Life Cycle Management Platforms Marios Fokaefs.

NOGARE D, MELLO RF & LOPES MA. 2022. Automação no processo de publicação de modelos de Ciência de Dados. In Anais Estendidos do XIII Congresso Brasileiro de Software: Teoria e Prática (Uberlândia/MG). SBC, Porto Alegre, RS, Brasil, 40-43 p.

PALEYES A, URMA RG& LAWRENCE ND. 2022. Challenges in deploying machine learning: a survey of case studies. ACM computing surveys 55: 1-29.

PETRIN M. 2024. AI, New Technologies, and Corporate Governance: Three Phenomena. Seattle University Law Review, Forthcoming.

SERBAN A, DER BLOM KV, HOOS H & VISSER J. 2020. Adoption and effects of software engineering best practices in machine learning. In: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), p. 1-12.

SHI VG, HUGLES R, MCKAY A, CHITTENDEN R & WALLER A. 2020. Visualising the impact of early design decisions on a modular housing supply network. In: 2020 Winter Simulation Conference (WSC). p. 1432-1443

SHORTRIDGE K & DYKSTRA J. 2023. Opportunity Cost and Missed Chances in Optimizing Cybersecurity. Commun. ACM 66: 96-104

SUBRAMANYA R, SIERLA S & VYATKIN V. 2022. From DevOps to MLOps: Overview and Application to Electricity Market Forecasting. Appl Sci (Switzerland) 12: 9851.

SULIANTORO H, CAESARALI A & HANDAYANI NU. 2022. The Development of Procurement Role as a Strategic Function in Public Hospital. In: Proceedings of the 2022 International Conference on Engineering and Information Technology for Sustainable Industry, 1-8 p.

SYMEONIDIS G, NERANTZIS E, KAZAKIS A & PAPAKOSTAS GA. 2022. MLOps - Definitions, Tools and Challenges. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference. CCWC 2022, 453-460 p.

TESTI M, BALLABIO M, FRONTONI E, IANNELLO G, MOCCIA S, SODA P & VESSIO G. 2022. MLOps: a taxonomy and a methodology. IEEE Access 10: 63606-63618.

ULUDAĞ Ö & MATTHES F. 2020. Large-scale agile development patterns for enterprise and solution architects. In: Proceedings of the European Conference on Pattern Languages of Programs 2020. p. 1–22.

WASHIZAKI H, UCHIDA H, KHOMH F & GUÉHÉNEUC YG. 2019. Studying software engineering patterns for designing machine learning systems. In: 2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP). IEEE, p. 49-495.

WELSH D & MARTINEZ SB. 2023. Framework for usage potential for Digital Procurement in the Construction Industry: Digital Procurement in Construction Projects Potential for more transparent and effective procurement of maintenance and installation services in the construction industry. In: Proceedings of the 2023 9th International Conference on Computer Technology Applications. p. 128-136. Kyle Wiggers. 2019. IDC: For 1 in 4 companies, half of all AI projects fail, 2019. Retrieved Oct 14 (2019), 2022.

ZHAO Y, HELANDER M, SANDHAL K & KARSSON J. 1999. Decision making for make vs. buy of software components. In Proceedings of 2nd Conference on Computer Science and Systems Engineering, 207-215 p.

How to cite

NOGARE D, SILVEIRA IF, BANZAI R & ALEXANDRE MC. 2025. Make or buy strategy for Machine Learning Operations – MLOps. An Acad Bras Cienc 97: e20240924. DOI 10.1590/0001-3765202520240924.

Manuscript received on August 30, 2024;
accepted for publication on February 9, 2025

DIEGO NOGARE[1,2]
https://orcid.org/0000-0003-0796-9431

ISMAR F. SILVEIRA[1]
https://orcid.org/0000-0001-8029-072X

RENATO BANZAI[2]
https://orcid.org/0009-0008-6056-7750

MAÍNA C. ALEXANDRE[2]
https://orcid.org/0009-0000-3102-7730

[1]Universidade Presbiteriana Mackenzie, Programa de Pós-Graduação em Engenharia Elétrica e Computação - PPGEEC, Rua da Consolação, 930, 01302-907 São Paulo, SP, Brazil

[1]Instituto de Ciência e Tecnologia Itaú- ICTi, Rua Volkswagen, 1, 04344-902 São Paulo, SP, Brazil

Correspondence to: **Diego Nogare**
diego.nogare@gmail.com

Author contributions

DIEGO NOGARE was responsible for the main idea, conducting experiments, planning, developing the methodology and drafting the manuscript. ISMAR FRANGO SILVEIRA worked on overall supervision, directing the development and experiments. RENATO BANZAI participated in experimental execution, project planning, and drafting the manuscript. MAÍNA CAÇADOR ALEXANDRE participated in planning and execution of experiments and supporting translation for English. All authors actively participated in reviewing and editing the manuscript.