/bluetab
an IBM Company

Increasing the usage capabilities of Artificial Intelligence in the organization.

01

Introduction

Across industries, we have seen DevOps and DataOps widely adopted as methodologies to improve quality and reduce time to market for software engineering and data engineering related initiatives, respectively. However, the MLOps discussion often focuses exclusively on tools, overlooking a critical aspect of successful Machine Learning (ML) investment, which is to enable people to achieve their goals. This involves designing the right structure for the organization.

In addition, it is essential to consider the unique aspects of machine learning and how general software development principles may not always be applicable to these projects.

Why does Bluetab believe this is important?

02

Organization Design

The organizational structure influences how effectively people align to achieve business objectives. Depending on the maturity of the organization, Machine Learning (ML) profiles can be integrated into business teams rather than centralized in isolated teams. Defining clear roles is essential for individuals to focus on areas of interest without becoming too dispersed; some companies may also benefit from a dedicated team focused on managing the ML platform.

Below, a series of competencies are detailed, along with reference job titles:

- **Data Scientist**: Explore and find ideas that influence business decisions.
- **Machine Learning Engineer**: Develop and evaluate ML models.
- **Machine Learning Researcher**: Experiment with new ML architectures.
- **Data Platform Engineer**: Manages the data infrastructure.
- **MLOps Engineer**: Manages ML platforms and operations.
- **Machine Learning Auditor**: Controls the risks associated with ML solutions.
- **Product Manager**: Defines which ML solutions are to be built.
- **Software Engineer**: Integrates ML capabilities into existing applications.

A key component in the efficiency of the organization is formalizing how information is exchanged to drive alignment with objectives and minimize duplication of efforts. This not only involves technologies or platforms but also requires defining those processes that are fundamental to the success of ML projects.

Processes

Evaluating which ML projects maximize impact on the business is not typically a primary task integrated into ML projects. Business impact can take various forms: reducing risks, increasing impact on ESG, or, more obviously, improving revenue or reducing operational costs.

Choosing the right projects and assessing business value upfront is crucial for long-term success. As a reference, we have the checklist[1] published by fast.ai where we can find the most important aspects when deciding which ML projects to prioritize. However, most ML projects fail due to deficient organizational structures and processes. Sometimes, the value of a project does not justify its cost, data may not be available, or poor software development practices may hinder experimentation and collaboration. This is to say that there are many reasons why projects should implement an MLOps vision to minimize operational risks.

The governance of MLOps proposes the definition of 7 processes with different degrees of maturity to achieve an integrated Machine Learning platform:

- **Development**: this process aims to identify if there is a sufficient amount of data and define the tasks of preparation and validation to ensure the quality of this data before experimentation.
- **Operativización**: automate the process of training and testing based on repeatable and reliable pipelines.

---

1 https://www.fast.ai/posts/2020-01-07-data-questionnaire.html

- **Continuous training**: Execute the training process in response to new data or code changes, or based on a schedule, potentially with new training configurations.
- **Model deployment**: Define a deployment process to serving infrastructure, either in batch and/or online.
- **Prediction serving**: It is about establishing the infrastructure that implements production for inference.
- **Continuous monitoring**: It consists of a process to monitor the effectiveness and efficiency of a model, as well as data quality, over time.
- **Management**: It is the centralized governance process that supports audit, traceability, compliance, and promotes collaboration.

In the next section, we will share the keys to define each of the processes.

04

MLOps: The Hard Way

Before embarking on an ML project, data often won't exist in an easily retrievable SQL table, or its status may be unknown, just like the project's progress may have unexpected developments. This uncertainty not only affects how much time specific tasks may require but also what tasks to perform. It is crucial to have a clear understanding of how to estimate the business impact or timelines in advance, considering the needs that involve organizational change.

Machine Learning Development Process

In an early phase of the process, the goal of data science is to quickly test various ideas through data exploration, as well as manage expectations by applying artificial intelligence-based algorithms. It is also important that the experimentation phase should only begin when the ML use case is well-defined, meaning it is well-documented, and the following questions have been answered:

- **What is the final task?**
- **What is the evaluation metric?**
- **Which data is relevant?**
- **How can business impact be measured?**
- **What are the consumption needs?**

Data scientists should use in the experimentation phase to obtain a prototype model effective enough for the defined use case. As introduced, data scientists tend to implement an end-to-end system, but the ML platform needs to manage the execution of the components proposed by the product teams. These components are software artifacts organized in a pipeline that simulates the work of a data scientist's programming. Additionally, through the experimentation phase, the following tasks are defined:

(1) Discovery of available data, analyzing its nature, and selecting variables that identify the expectations.
(2) Data preparation by making modifications deemed appropriate for each of the selected variables using a rapid prototype tool or code.
(3) Selection of a model that fits the properties of our variables.

Finally, data scientists will begin to refine the solution to the proposed use case through an iterative process until achieving business impact.

Process for Industrializing Training

Before deploying any ML model, we must conduct experiments to validate its feasibility and estimate its impact on the business. Operationalization is related to preparing the assets that define shared workflows in the organization, enabling data scientists to iterate on their ideas efficiently and seamlessly. This task is one of the main responsibilities of ML engineers who support the ML platform.

A robust process of model development and evaluation will enable data scientists to iterate quickly, communicate effectively with stakeholders, and properly assess models. ML engineers should be able to integrate a system that facilitates these tasks, meaning that at least the following aspects are defined:

(1) Define what the target environment is going to be.
(2) Runtime access to data sources.
(3) Appropriate permissions to execute the code.
(4) Evaluate the quality metrics of the models.

Often, it's necessary to look at metrics not only as a whole but also in specific data segments, which requires a preliminary step of setting up a development environment. This involves configuring a remote infrastructure with significant computing capacity rather than relying on personal devices. Jupyter Notebooks are crucial for quickly prototyping and experimenting with ideas, so any ML platform should be adapted for its use.

Continuous Training Process

Unlike the first two processes, where tasks performed by individuals are described, the continuous training process involves the orchestration and automation of training pipelines at the infrastructure level. It defines the frequency with which model training is repeated, depending on the rules and cost impact of the use case. While this marks the end of the development process, it is also the beginning of the model maintenance cycle.

Models must be monitored and periodically retrained to address efficiency issues, leverage new input variables, or simply adapt to changes in the code. This process is truly about a systematic iteration to continuously refine and ensure model accuracy. However, identifying the right time to re-run training for a model is not a trivial task: retraining too often can lead to service interruptions without significant new improvements, while not retraining frequently enough can result in the degradation of model performance.

Each execution of the continuous training pipeline can be triggered in various ways, including the simplest ones such as the following:

> A. Scheduled jobs based on the desired configuration (e.g., time).
> B. Executions based on events, such as when new data becomes available above a certain threshold that modifies the data distribution shape.
> C. When a substantial deterioration is detected in the monitoring process, manual invocations can be performed, managed by the ML platform team.

Typical tasks for these types of processes include the following workflows:

1. **Data Ingestion.** Training data is extracted from the source dataset and the feature repository using extraction criteria defined by data scientists (also known as queries) and the period of the most recent update.
2. **Data Validation**. The training data that has been extracted is validated to ensure that the model is not trained using biased or incomplete data.
3. **Data Transformation:** The data is typically split into training, evaluation, and validation sets, and then transformed to obtain the features as expected by the model.
4. **Model Training.** The algorithm is trained, and the hyperparameters are adjusted during each iteration of the training to produce the best possible model.
5. **Model Evaluation**. The model is evaluated against test data to obtain theoretical performance, using various metrics and employing different data partitioning strategies.
6. **Model Validation**. The results of the model's previous evaluations are scrutinized to ensure that the model meets the business criteria.
7. **Model Registry**. The validated model is stored in the model registry with the necessary metadata to make it operational.

Finally, as we will see in the continuous monitoring process, promoting a model to production requires following a defined workflow, and we also need to be able to track the lineage of the model to associate data with metrics, as we will see later on.

Model Deployment Process

After training, validating, and adding a candidate model to the model registry, we are finally ready for its final deployment. During the deployment process, the defined software is packaged, tested for proper operation, and deployed in an environment to provide service.

The continuous deployment part is similar to the progressive delivery found in the DevOps methodology. Such deployments are carried out through the execution of strategies like canary or blue-green deployments, which focus on the efficiency of the serving process, avoiding any errors that might result in a service outage. Obtaining online predictions is a particularly important milestone in the context of ML. Deciding whether a new candidate model should replace the production model is more complex compared to a typical task in software engineering.

In the progressive delivery approach, the new candidate does not immediately replace the previous version but does so after a certain period during which both models coexist in parallel in production. A subset of consumers is redirected to the new candidate, increasing traffic in several stages until the final outcome determines that the model is fully released and replaces the old one. For this task, A/B testing is useful and can be employed to quantify the impact of the new model on the use case objectives and the applications that consume them.

Serving Process

The serving process for inferences begins right after the model has been deployed in the production environment. The model starts accepting requests for information about the data and generates appropriate responses.

The serving system can be provided with the following forms of consumption:

- **Online real-time inference for tasks with high frequency using endpoints.**
- **Streaming for cases where events arrive at queues or buses.**
- **Batch offline, typically integrated into bulk ETL processes.**
- **Integrated inference as part of IoT systems or edge devices.**

All records generated in the predictions and other consumption metrics during inferences should be stored for subsequent analysis and monitoring, as we will see in the next continuous monitoring process.

Continuous Monitoring Process

Continuous monitoring is vital in any platform, and model monitoring is a crucial area of the MLOps methodology for controlling efficiency and degradation in production. This process regularly and proactively verifies the model's performance to prevent any incidents in performance. As requests have new data that varies their distribution over time, making its properties start to deviate from the references used to train and evaluate the model.

This process leads to a more effective model over time by preventing degradation. Additionally, modifications in complementary systems that perform the transformation or retrieval of information from requests can produce changes in variables that, consequently, may result in poor predictions for the inference system. Therefore, monitoring is used to search through the records for any indicative signs that can identify anomalies, biases, or outliers. A process must include at least the following steps to be effective:

1. A sample of requests and responses should be extracted from the central log storage service.
2. The system periodically loads the most recent inferences to calculate statistics about the predictions the model is producing.
3. The generated data is compared with the reference to highlight possible biases, in addition to comparing the calculated statistics with those extracted from the evaluation data.
4. If there are manually intervened supervised predictions for the published data, the system can assess the effectiveness of the predictions.
5. If any degradation in efficiency is ultimately identified, alerts are generated and can be sent through various notification means to trigger a new retraining cycle.

In particular, when we talk about the model suffering from a degradation in the expected effectiveness, it actually has to be defined in terms of data and concept drift. The first describes the growing divergence in the dataset used to train, evaluate, and validate the model and the actual variables the model is using to make predictions, while the second involves the change in relationships between input and output data in the use case.
When we talk about data drift, we are involving two key aspects:

- Modifications in the schema that occur when the training and serving data do not share the same structure.

- The distributions of the variables forming the features for training are significantly different from the distribution obtained in the production of predictions.

In addition to the aforementioned techniques, other techniques can also detect deviations in data, including outlier detection in the data, changes in the function of variables, or ethical or gender connotations in productive models. Similarly, in some scenarios, it is viable to add information captured from the organization's back-office, using the feature repository to store it and be used in the future during a new execution of the continuous training process.

In addition to measuring effectiveness, the infrastructure service itself has to record resource consumption activity through the following metrics:

- Usage of computing resources, including CPUs, GPUs, and memory.
- Latency of prediction generation to indicate the health of the service.
- Throughput, which is a general metric for any deployment.
- Error rates produced in serving.

Quantifying these metrics is not only useful for maintaining and improving the overall performance of the system but also in cost management.

05

The platform

Often, it is difficult to decouple processes from technology. In an ideal world, we would start with the process and choose the most suitable technology to implement it. However, in reality, technology influences our processes and constrains them. It is worth building a machine learning platform that is flexible enough to support workflows, which in turn integrate with various different technological solutions.

From a technological standpoint, we have three indicators that make it easier to make the most relevant decisions when building your machine learning platform.

1) **Frameworks**

   The low-code or no-code tools are increasingly gaining ground in ML platforms, but professionals continue to prefer coding (PyTorch, Tensorflow, Keras, Scikit-learn, etc.).

2) **Ops tools**

   Performing each task manually is not efficient; there is a tendency to adopt development tools that facilitate best practices and yield optimal results. Tools such as Kedro, MLflow, or Weights & Biases aid in model tracking, logging and labeling, versioning, monitoring, and final orchestration.

3) **Infrastructure**

   In order to execute workflows, a significant amount of processing and storage resources is required, with elastic demand over time. These resources are often obtained through cloud computing (AWS, GCP, Azure), although some organizations still choose to manage them on their own.

It is important to strike a balance between the tasks of different roles within the organization. The more data scientists focus on a particular process, the more freedom they will desire. In general, creating an ML platform that governs this flexibility is a good idea. Providing high-level solutions that reduce the time spent configuring infrastructure resources may limit their freedom but will focus on specific tasks where they can add the most value. Finally, it is crucial to understand developers, offering them a familiar environment with APIs, Python, or SQL, and supporting them in development environments where they feel comfortable, such as Jupyter Notebooks.

06

Our Point of View

AI is experiencing an unprecedented wave of innovation.

We are witnessing an unprecedented wave of innovation in the field of AI. In a survey conducted by McKinsey last August titled "The state of AI in 2023: Generative AI's breakout year," one-third of respondents indicated that Generative AI is being regularly used for some tasks in their organizations. Generative AI has exploded in just the past 12 months!

How is that possible?

AI is a special technology. One way to conceptualize AI is through a large toolbox (the models). These tools come in various types, such as supervised learning, Generative AI, or reinforcement learning, and they are capable of performing a wide range of tasks, such as converting an image into text, identifying email spam, or translating. This diversity of tasks makes the transformative potential of AI very high.

One characteristic of this wave of innovation is the emergence of new types of highly powerful tools. Generative AI has added new capabilities to the toolbox, such as summarizing conversations, writing emails, or managing knowledge, and it has much faster development cycles—going from months to weeks!

---

https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-br eakout-year

The most common AI tools are those of supervised learning. These tools are excellent at recognizing patterns, but creating such a tool requires a significant amount of time. One needs to prepare a very large training dataset and undergo the training process, which typically takes more than 6 months. Generative AI allows for the creation of tools much more quickly. We start with a pre-trained version of the tool, and we only need to adapt it using a small dataset. The process takes a few weeks.

In this scenario of having a toolbox with increasingly more capabilities and faster development cycles, it is normal for CEOs and CIOs of companies to have AI on their strategic agendas.

Installing elements and governance practices for AI is one of the key elements for success in this endeavor.

Adopting AI in a company, expanding its toolbox, has accelerated significantly. Ultimately, the goal is to transform the company by making its activities more efficient and productive or creating new activities.

Using AI introduces risks such as biases, lack of precision, or inequity. AI is continually improving, and these risks are diminishing, but they should always be considered. Just as importantly, ethical considerations must always be taken into account when using this technology.

A good idea for success in AI adoption is to establish AI governance elements and practices from the outset and to improve these governance elements and practices as we learn to use AI.

AI governance encompasses everything from managing the toolbox, training, risk analysis, measuring impacts, ways of working on the data and AI platform, monitoring, to decisions about prioritizing use cases, investing in capabilities, or establishing ethical principles for AI use in the company.

We perceive AI governance as a journey with a maturity curve. That's why we see it as crucial to install its elements and practices from the beginning.

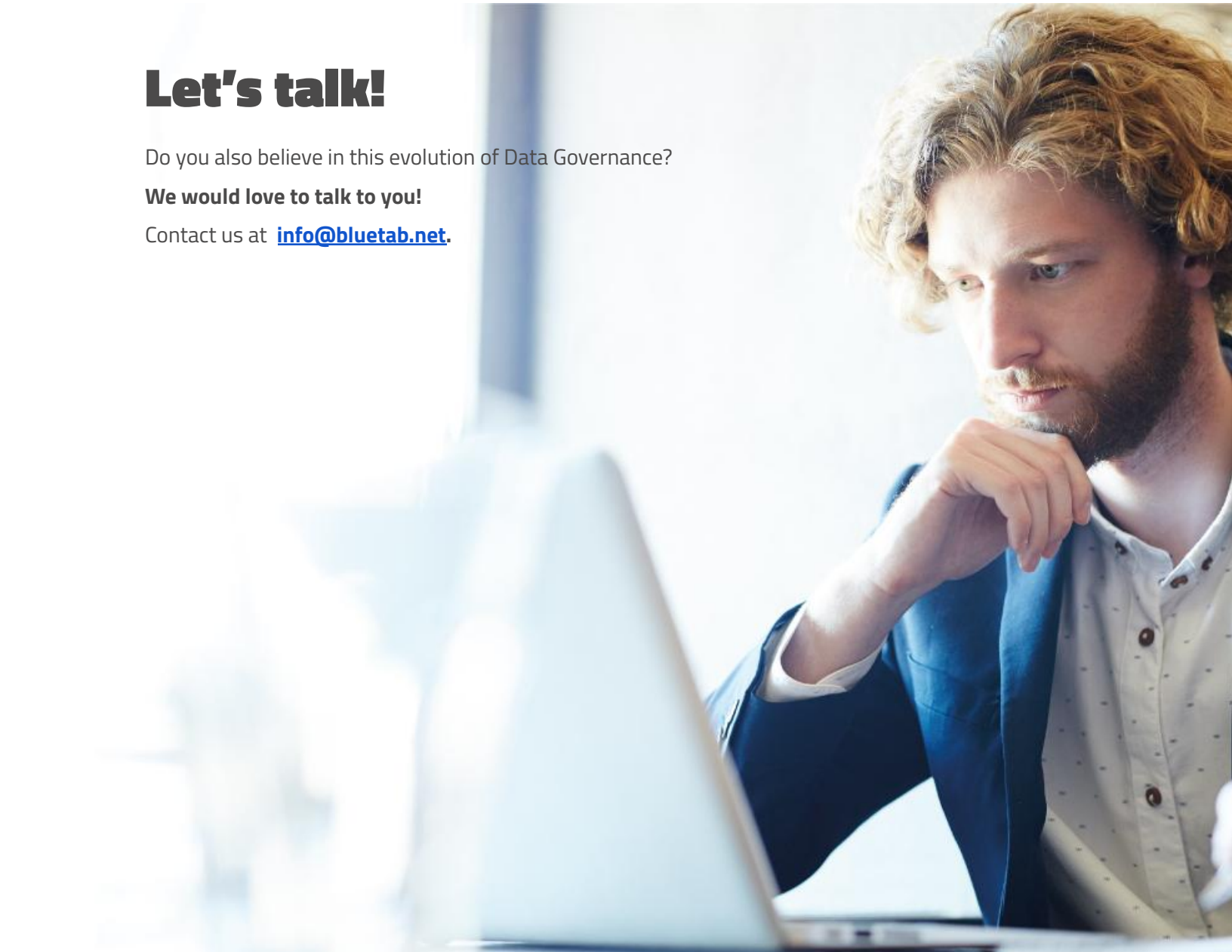
Another interesting question from the McKinsey survey "The state of AI in 2023: Generative AI's breakout year" is about the elements posing the greatest challenge in reaping the benefits of AI. Respondents classified as "high performers" point to elements related to their maturity in AI usage and governance, such as tool performance monitoring and retraining. Meanwhile, other respondents highlight more foundational elements like strategy and talent.

Let's talk!

Do you also believe in this evolution of Data Governance?

We would love to talk to you!

Contact us at **info@bluetab.net.**




bluetab
an IBM Company