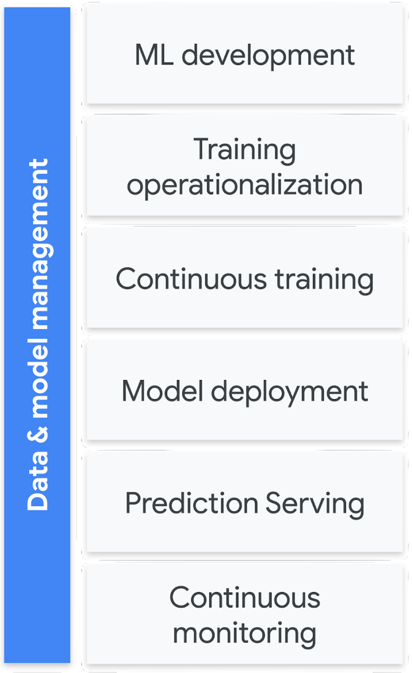
MLOps Overview



While many organizations recognize that Machine Learning (ML) can drive significant value, successful deployments and effective operations are the main bottleneck for gaining value from AI. As of today, more than 87% of data science projects never make it into production.

To support organizations in coming up to speed faster in this important domain, it is important to understand ML Operations (MLOps).

A Brief Introduction to MLOps

Teams across all industries and domains find themselves spending an unnecessary amount of time on the development of ML models, without the necessary people, processes, or technology to efficiently deploy and operationalize them. In the current landscape, we find a high degree of manual, one-off work as many engineers are constantly crafting custom solutions, inflexibility of the resulting deployments as components are not reusable nor reproducible, and a high number of errors as these custom solutions are often poorly documented leading to issues during handoffs between Data Science and IT. As such, ML solutions often fall short of their goals and cost their organizations time and money. In summary, deploying ML models at scale is very challenging, especially due to:

- **Organization Barriers** - Managing different ecosystems like programming languages
- **Compute Constraints** - Continuous and reliable compute resources (dedicated servers or cloud-only options)
- **Portability Issues** - Huge dependencies on legacy systems
- **Seasonality** - ML workloads works in patches; this need auto-scaling capabilities

Businesses are looking to reduce the time from change (via data or code) to deployment, improve deployment frequency, speed up time to value with new ML use cases, lower the failure rate of new releases, shorten the lead time between fixes, and facilitate better collaboration (reusing features and sharing models).

To address this, organizations need to build the necessary ML engineering culture and capability MLOps aims to unify ML system development (ML) with ML system operations (Ops). MLOps strongly advocates automation and monitoring at all steps of ML system construction, from integration, testing, and releasing to deployment and infrastructure management. It takes both its name as well as some of the core principles and tooling from DevOps. This makes sense as the goals of MLOps and DevOps are practically the same: to shorten the systems development life cycle and ensure high-quality software is continuously developed, delivered, and maintained in production. Its Machine Learning's unique challenges and needs – managing the lifecycle of Data, Models, and Code – has led MLOps to quickly evolve as a domain of its own.

To support the engineering community to realize MLOps in practice, Google made available to the world the practitioners guide to MLOps to help developers implement MLOps and smart practices during the ML workflow. We will go through this applied pipeline and we will discuss the unique challenges that emerge as a result of deploying TinyML to many embedded devices.

Amongst many processes and operations, MLOps requires:

- **Continuous Integratio**n: Is no longer only about testing and validating code and components, but also testing and validating data, data schemas, and models.
- **Continuous Delivery**: Is no longer about a single software package or a service, but a system (ML training pipeline) that should automatically deploy another service (model prediction service)
- **Continuous Training**: This is a new property, specific to ML systems, concerning automatically retraining and serving the models.

The maturity of the ML process is defined by the level of integration and automation of these phases shown in the figure below, which reflects the velocity of training new models, quality of the assets generated and reliability of the overall system.