Microsoft

Introduction to MLOps and the ML lifecycle

In this document, we discuss the typical machine learning (ML) lifecycle and its common challenges.

Companies don't have to build an ML model from scratch. They can purchase products that have AI baked in or use out-of-the-box models like Azure Cognitive Services.

But this provides an overview of the process if you're building a "custom" model, either from the ground up or using a pre-trained model as a starting point.

ML challenges

As more organizations experiment with AI, they find that creating a machine learning (ML) model is just the first of many steps in the ML lifecycle.

Managing the entire lifecycle at scale is complicated. Organizations have to be able to document and manage data, code, model environments, and the machine learning models themselves. They need to establish processes for developing, packaging, and deploying models, as well as monitoring their performance and occasionally retraining them. And most organizations are managing multiple models in production at the same time, adding to the complexity. All of this is challenging due to lack of:

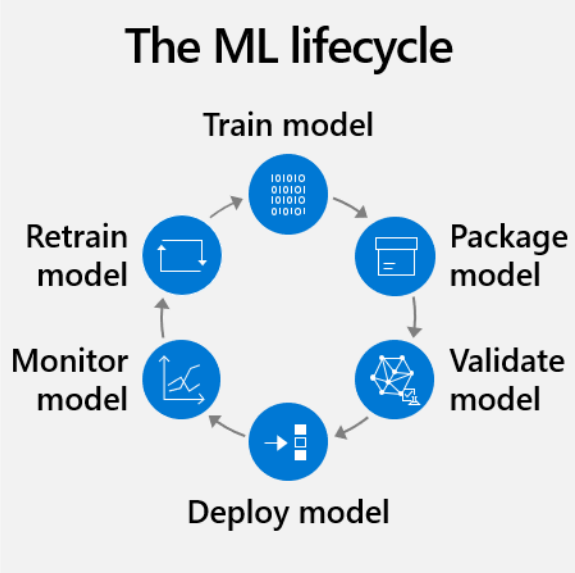
- **Cross-team alignment:** Siloed teams impede workflow alignment and collaboration.
- **Standard, repeatable processes:** Without automated and repeatable processes, employees have to reinvent the wheel each time they create and deploy a new model.
- **Resources:** Large amounts of time and personnel are required to manage the lifecycle.
- **Auditability:** It can be difficult to ensure that models meet regulatory standards and performance thresholds over time.
- **Explainability:** Black box models make it difficult to understand how the model works.

These challenges are similar to what application development teams face when creating and managing apps. To help, they use DevOps, the industry standard for managing operations for an application development cycle. To address these challenges with machine learning, organizations need an approach that brings the agility of DevOps to the ML lifecycle. We call this approach MLOps.

Typical ML lifecycle

Train and test: First, data scientists need to prepare training data. This is often the biggest time commitment in the lifecycle. Preparation includes standardizing the data so it's in a usable format and identifying discrete "features" or variables. For example, to predict credit risk, features might include customer age, account size, and account age. Next, they apply algorithms to the data to "train" a machine learning model. Then they test it with new data to see how accurate its predictions are.

Package: ML engineers containerize the model with its environment, which means creating a docker container for the model to run in with all its dependencies. The model



environment includes metadata like code libraries that the model needs to execute seamlessly.

Validate: At this point, the team evaluates how model performance compares to their business goals. For example, a company might want to optimize for accuracy over speed in some cases.

> **Repeat steps 1-3:** It can take hundreds of training hours to find a satisfactory model. The development team may train many versions of the model by adjusting training data, tuning algorithm hyperparameters, or trying totally different algorithms. Ideally the model improves with each round of adjustment. Ultimately, it's the development team's role to determine which version of the model best fits the business use case.

Deploy: Finally, they deploy the model in the cloud (often through an API), on an on-prem server, or at the edge on devices like cameras, IoT gateways, or machinery.

Monitor and retrain: Even if a model works well at first,
it needs to be continually monitored and retrained to stay relevant and accurate.

MLOps processes and tools are valuable throughout all of these stages of the lifecycle.

- They help teams collaborate and provide visibility through shared, auditable documentation.
- They provide the ability to save and track changes to data sources, code, libraries, SDKs, and models.
- And they create efficiencies and accelerate the lifecycle with automation, repeatable workflows, and reusable assets.