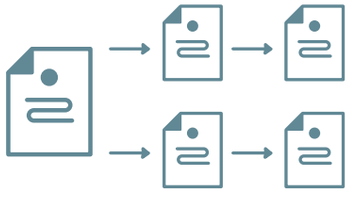
Use case examples

Let's examine three use cases to explore how users can leverage some of the MLflow components.

EXPERIMENT TRACKING A European energy company is using MLflow to track and update hundreds of energy-grid models. This company's goal is to build a time-series model for every major energy producer (e.g., power plant) and consumer (e.g., factory), monitor these models using standard metrics, and combine the predictions to drive business processes, such as pricing. Because a single team is responsible for hundreds of models, possibly using different ML libraries, it's important to have a standard development and tracking process. The team has standardized on Jupyter notebooks for development, MLflow Tracking for metrics, and Databricks Jobs for inference.



REPRODUCIBLE PROJECTS An online marketplace is using MLflow to package deep learning jobs using Keras and run them in the cloud. Each data scientist develops models locally on a laptop using a small data set, checks them into a Git repository with an MLproject file, and submits remote runs of the project to GPU instances in the cloud for large-scale training or hyperparameter search. Using MLflow Projects makes it easy to create the same software environment in the cloud and share project code among data scientists.

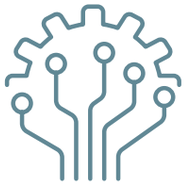
MODEL PACKAGING An e-commerce site's data science team is using MLflow Model Registry to package recommendation models for use by application engineers. This presents a technical challenge because the recommendation application includes both a standard, off-the-shelf recommendation model and custom business logic for pre- and post-processing. For example, the application might include custom code to ensure the recommended items are diverse. This business logic needs to change in sync with the model, and the data science team wants to control both the business logic and the model, without having to submit a patch to the web application each time the logic has to change. Moreover, the team wants to A/B test distinct models with distinct versions of the processing logic. The solution was to package both the recommendation model and the custom logic using the `python_function` flavor in an MLflow Model, which can then be deployed and tested as a single unit.

Open and extensible by design

Since we unveiled and open sourced MLflow in June 2018 at the Spark + AI Summit in San Francisco, community engagement and contributions have led to an impressive array of new features and integrations:

SUPPORT FOR MULTIPLE PROGRAMMING LANGUAGES

To give developers a choice, MLflow supports R, Python, Java and Scala, along with a REST server interface that can be used from any language.



INTEGRATION WITH POPULAR ML LIBRARIES AND FRAMEWORKS

MLflow has built-in integrations with the most popular machine learning libraries — such as scikit-learn, TensorFlow, Keras, PyTorch, H2O, and Apache Spark™ MLlib — to help teams build, test and deploy machine learning applications.

CROSS-CLOUD SUPPORT

Organizations can use MLflow to quickly deploy machine learning models to multiple cloud services, including Databricks, Azure Machine Learning and Amazon SageMaker, depending on their needs. MLflow leverages AWS S3, Google Cloud Storage and Azure Data Lake Storage, allowing teams to easily track and share artifacts from their code.

Rapid community adoption

2.5M
monthly downloads

200+
code contributors

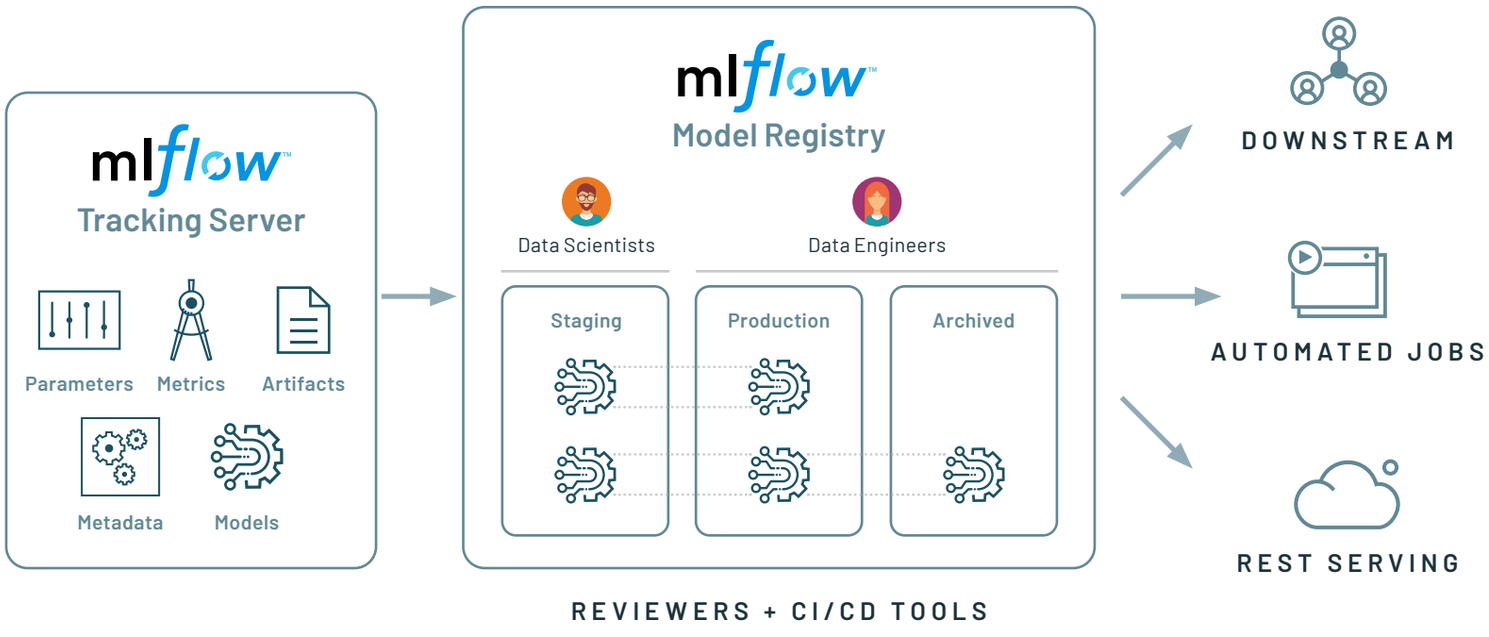
100+
contributing organizations

Organizations using and contributing to MLflow
Source: mlflow.org

CHAPTER 4: A Closer Look at MLflow Model Registry

MLflow originally introduced the ability to track metrics, parameters and artifacts as part of experiments, package models and reproducible ML projects, and deploy models to batch or to real-time serving platforms.

The latest MLflow component — MLflow Model Registry — builds on MLflow's original capabilities to provide organizations with one central place to share ML models, collaborate on moving them from experimentation to testing and production, and implement approval and governance workflows.



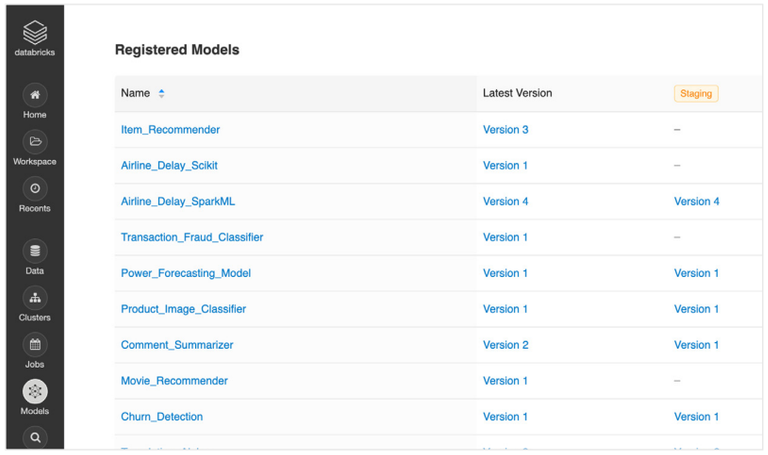
The Model Registry gives MLflow users new tools for sharing, reviewing and managing ML models throughout their lifecycle

The MLflow Model Registry complements the MLflow offering and is designed to help organizations implement good engineering principles with machine learning initiatives, such as collaboration, governance, reproducibility and knowledge management. The next few pages highlight some of the key features of this new component.

One hub for managing ML models collaboratively

Building and deploying ML models is a team sport. Not only are the responsibilities along the machine learning model lifecycle often split across multiple people (e.g., data scientists train models whereas production engineers deploy them), but also at each lifecycle stage, teams can benefit from collaboration and sharing (e.g., a fraud model built in one part of the organization could be reused in others).

MLflow facilitates sharing of expertise and knowledge across teams by making ML models more discoverable and providing collaborative features to jointly improve on common ML tasks. Simply register an MLflow model from your experiments to get started. The MLflow Model Registry will then let you track multiple versions of the model and mark each one with a lifecycle stage: development, staging, production or archived.

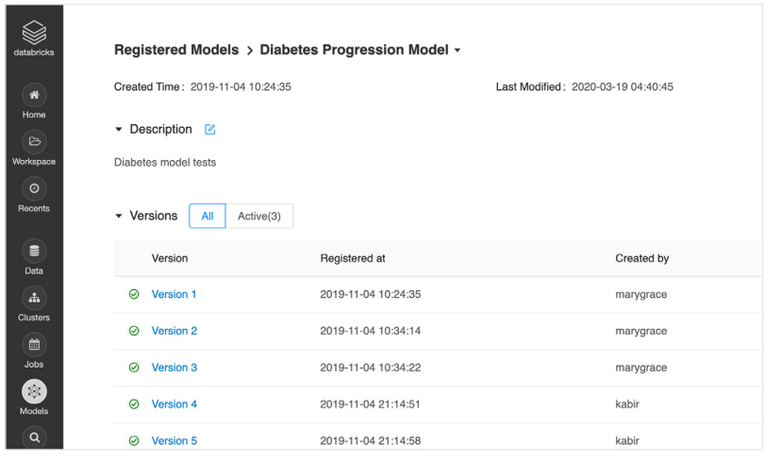


Sample machine learning models displayed via the MLflow Model Registry dashboard

Flexible CI/CD pipelines to manage stage transitions

MLflow lets you manage your models' lifecycles either manually or through automated tools. Analogous to the approval process in software engineering, users can manually request to move a model to a new lifecycle stage (e.g., from staging to production), and review or comment on other users' transition requests.

Alternatively, you can use the Model Registry's API to plug in continuous integration and deployment (CI/CD) tools, such as Jenkins, to automatically test and transition your models. Each model also links to the experiment run that built it — in MLflow Tracking — to let you easily review models.

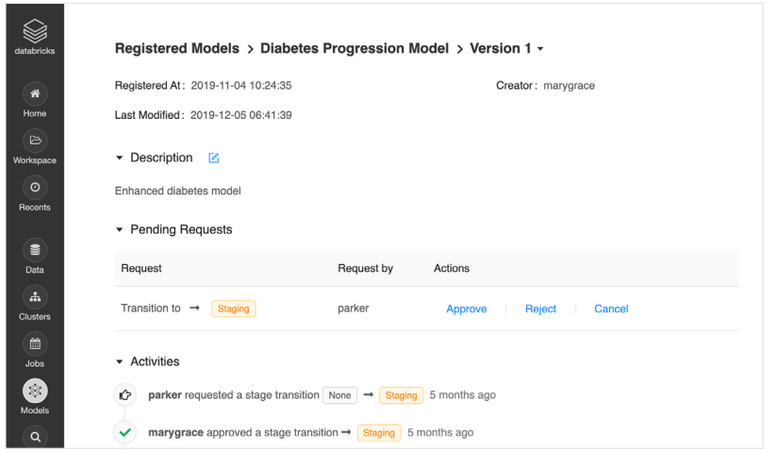


The machine learning model page view in MLflow, showing how users can request and review changes to a model's stage

Visibility and governance for the full ML lifecycle

In large enterprises, the number of ML models that are in development, staging and production at any given point in time may be in the hundreds or thousands. Having full visibility into which models exist, what stages they are in and who has collaborated on and changed the deployment stages of a model allows organizations to better manage their ML efforts.

MLflow provides full visibility and enables governance by keeping track of each model's history and managing who can approve changes to the model's stages.



Identify versions, stages and authors of each model