indium
Make Technology Work

MLOps Powered by the Databricks Lakehouse Platform

A Whitepaper

MLOps refers to a set of processes and automation that enable organizations to manage data, code, and models. It enables the productionization, maintenance, and monitoring of machine learning models in a streamlined manner. It is a combination of DataOps, DevOps, and ModelOps that helps data scientists accelerate an innovative approach to model architecture, feature engineering, and hyperparameters.

Implementing MLOps best practices helps data scientists and engineers automatically build, test, and deploy new pipelines to the production environment. With a robust and automated MLOps system, organizations can improve the return on investment in data and ML from their AI initiatives.

MLOps Implementation Challenges

Despite its advantages, MLOps implementation presents several challenges. As data, models, and code evolve with time in machine learning (ML) systems, it can cause friction when refreshing schedules and ML pipelines. This data drifting can also lead to degradation in performance over time, as the model needs to be retrained.

One of the most common and painful challenges businesses face with MLOps implementation is the widening gap between data and ML, split across tools and teams that are not integrated well.

Data scientists may also discover improved features or model families for enhanced performance as they explore the datasets and experiment with various approaches to improve their models. This will require the code to be updated and redeployed from time to time.

The challenges stem from the fact that the machine learning life cycle is essentially triggered as a response to a business problem and involves the following process, which can be done manually or using an automatic pipeline.

Stage 1: Data Preparation

The building of ML models requires clean and large volumes of data for training. Data scientists extract data from multiple sources and perform EDA (exploratory data analysis) to understand the data characteristics and schema for building models. The data needs to be cleaned, and categorized for training/testing/validation, after performing tasks such as encoding, imputing missing values, transforming, and feature engineering, among others.

Stage 2: Model Training and Deployment

After data preparation, the next step is the application of various algorithms for ML model training, its performance evaluated, and the model validated to ensure it can be deployed. The performance of the model is compared with a specific baseline model and if it performs as expected, then it is deployed and productionized.

Stage 3: Monitoring the Model

The performance of the predictive model must be monitored continuously for iterations as the need grows.

Since ML models are created as a response to a business problem, each time a new batch of data is ingested, the ML model needs to undergo an iteration without the response times being affected or the app performance being affected. As the apps run into several thousand lines of code and the data set size expands, this can be challenging. This iterative process has created the need for data science models to be brought to production. Scaling becomes difficult and rebuilding to fit models in distributed platforms becomes critical.

Also, the earlier ML methodology becomes ineffective as data changes, needing new models to be built and deployed. It also becomes expensive to change the model, datasets, or derived data due to the need to repeat the experiments to make them accurate.

Best Practices for MLOps, Crystallized in Databricks

To overcome these challenges, automation of data pipelines will help to reduce the need to rewrite code by delivering value continuously, consistently, and efficiently.

Centralized tracking and visualization of the progress of all models across the organization and automatic data validation policies will also help simplify the maintenance of ML-models.

A robust and automated Continuous Integration/Continuous Deployment (CI/CD) system for a mature MLOps that enables testing and deployment of ML pipelines and Continuous Training (CT) and Continuous Monitoring (CM) is required to overcome the MLOps challenges.

With Continuous Monitoring, model performance degradation of the pipeline triggers automated re-training. Building MLOps on top of the Databricks Lakehouse platform improves the joint management of data, code, and models to enable the automation of code, data, and model management.

This ensures stable performance and greater efficiency of the ML systems.

The assets can be managed on a single Databricks Lakehouse platform with unified access control. It facilitates the development of data applications and ML applications on a single platform, thereby speeding up the process of moving data around with reduced risks and delays.

Databricks extends DevOps tooling and CI/CD processes to ML by clearly defining the processes for moving code, data, and models to production.

It further simplifies operations using feature computation, inference, and other data pipelines which follow the same deployment process as the model training code.

The MLflow Model Registry, a designated service, allows the independent updating of code and models, thereby enabling the adoption of DevOps methods for ML.

Collaboration and Management

The Lakehouse provides a unified platform to support data engineering, data science, production ML, and business analytics built on a shared lakehouse data layer.

As a result, the same lakehouse architecture used to manage the ML data is accessed by other data pipelines, making hand-offs simple.

With appropriate access controls, execution environments, code, data, and models can be accessed by the right teams with appropriate authorization, thus simplifying security and governance.

It provides flexibility and visibility to data scientists to develop and maintain models while allowing control over production systems to ML engineers.

Integration and Customization

As the Lakehouse uses open formats and APIs such as Git, relevant CI/CD tools, Delta Lake and Lakehouse architecture, and MLflow, the ML assets are stored in open formats and supported by services with open APIs.

This makes integrating modules with the existing infrastructure and customization possible.

The Databricks MLOps Architecture

Databricks helps move away from a model-centric approach to solving business problems to creating a data-centric approach. It also helps with model governance to improve compliance with external systems.

Databricks Machine Learning uses the lakehouse architecture and supports critical MLOps and governance needs such as model management, secure collaboration, testing and documentation.

The Databricks MLOps architecture allows the loose coupling of models and code that allows flexible updating of production models without the need to change the code, and vice versa.

Its execution environments allow codes to consume or create models and data. The ML pipelines are defined by code and stored in Git for version control and facilitating model training and tuning, featurization, monitoring, and inference.

Data scientists may begin with notebooks when starting ML pipelines and later transition to modularized code, if required, using Databricks or IDEs.

Data is stored in Delta Lake, a lakehouse architecture on the cloud that allows storing of structured and unstructured data in an open and efficient storage layer. Storing raw data and feature tables as Delta tables guarantees consistency and performance and is supported by an optimized Delta Engine.

MLflow is used to manage models from any ML library and for any deployment mode with access control and scalability. It makes models reproducible MLflow by allowing the logging of codes, data sources, library dependencies, infrastructure, the model, and arbitrary artifacts such as SHAP explainers and pandas profiling at the time of training.

They can be activated during training. When the models are moved to the centralized Model Registry, data is preserved and provides an audit trail for design, authorship, and data lineage.

As a result, model versions can be maintained in the registry and rolled back for debugging and investigation by tracing a model artifact back to its source.

Indium to Implement MLOps on Databricks Lakehouse Platform

Indium Software, recognized by ISG as a strong contender for data engineering projects, has a proven capability in Databricks solution development, machine learning, and DevOps. Our team of experts with cross-domain experience partners with customers to build the bespoke MLOps architecture using Databricks Lakehouse and accelerate innovation.

Indium's Unified Data Analytics platform on Databricks, Ibrix, showcases Indium's expertise in Databricks technology and improves business agility and performance by providing deep insights for different use cases.

Some of our strengths include:

- 120+ person-years of expertise in Spark
- A dedicated Databricks lab and COE
- ibriX – A homegrown Databricks accelerator for faster time-to-market
- Cost optimization framework – Greenfield and Brownfield engagements
- E2E Data Expertise – Lakehouse, ML Ops, advanced analytics, and data products
- Wide industry experience – healthcare, financial services, manlog, retail, and realty

To know more, visit:
https://www.indiumsoftware.com/databricks-consulting-services/

FAQs

What are the best practices of MLOps?

Some of the best practices for MLOps include:

- Iterative exploration, sharing, and preparation of data for the machine learning lifecycle by creating datasets, tables, and visualizations that are reproducible, editable, and shareable
- Iterative transformation, aggregation, and de-duplication of data for creating refined features that are visible and shareable across data teams
- Training and improvement of model performance using popular open-source libraries such as scikit-learn and hyperopt
- Tracking and management of the model lineage, versions, artifacts, and transitions through their lifecycle.
- Management and automation of the pre-production pipeline
- Deployment and monitoring of the models through automated permissions and cluster creation to enable the production of registered models.
- Automation of model retraining

USA
Cupertino | Princeton
Toll-free: +1-888-207-5969

INDIA
Chennai | Bengaluru | Mumbai | Hyderabad
Toll-free: 1800-123-1191

UK
London
Ph: +44 1420 300014

SINGAPORE
Singapore
Ph: +65 6812 7888

www.indiumsoftware.com

For Sales Inquiries
sales@indiumsoftware.com

For General Inquiries
info@indiumsoftware.com