



MLflow is an open source platform for managing the machine learning (ML) model lifecycle, addressing challenges in evaluation, reproducibility, observability, collaboration, and deployment. It eases and standardizes model experimentation, sharing, deployment, and monitoring, while supporting diverse ML tools.

Usage Sample

Tracking model metadata involves interweaving MLflow API calls into model training code:

```
with mlflow.start_run():  
    mlflow.log_param("lr", 0.001)  
    # Model training code goes here  
    mlflow.log_metric("loss", loss)
```

To access the dashboard in a browser:

```
mlflow ui --port 5000
```

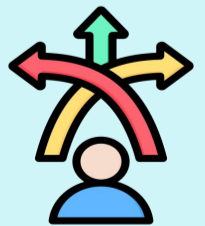
Companies using MLFlow



Strengths



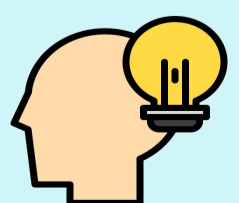
Open source and active
well maintained, well documented, well utilized, and with strong industry support.



Flexible
Compatible with diverse ML libraries and workflows.

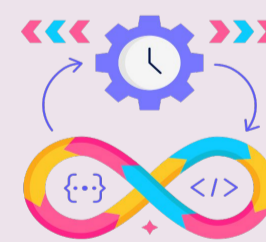


Minimalistic
Convenience methods like `.autolog()` reduce code changes for standard routines.



Intuitive
Built-in UI simplifies visualization and plotting

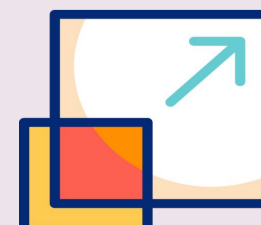
Weaknesses



DevOps
Self-hosted setup requires more effort than managed tools like Weights & Biases.

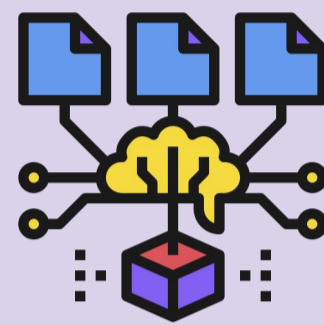


Lock-in
Advanced scalability features may tie users to Databricks.



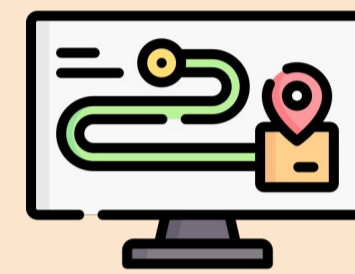
Scaling
Alternatives often preferred for pipeline management and distributed computing.

MLflow Tracking



API and UI for logging, querying, and visualizing ML model metadata (e.g., parameters, code versions, metrics, outputs) and artifacts (any output files produced).

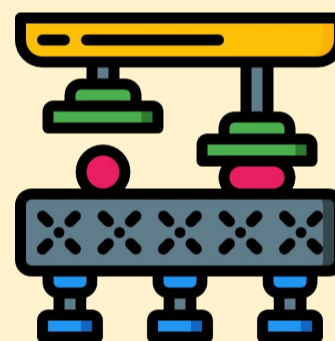
MLflow Models



Standard format for deploying ML models across platforms like Docker, Apache Spark, AWS SageMaker, and Azure ML.

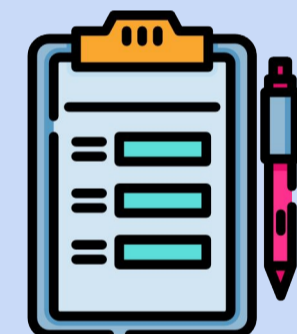
Core Components

MLflow Projects



Standardized packaging for reproducible code, with configurable environments and execution entry points.

MLflow Registry



Centralized store for managing models with versioning, stage transitions, and Markdown annotations.