



# MLflow: Accelerating ML For the Enterprise

Manage the complete machine learning lifecycle with enterprise reliability, security and scale



---

May 4, 2022

# HOUSEKEEPING

- This session will be **recorded** and the **slides and notebooks will be shared**.
- We will use the **Zoom Q&A** for asking any questions during the training – we will do our best to answer them in the Q&A or out loud. Please use the **Zoom Chat** for any general comments. If we are unable to answer your questions during the session your account team will follow-up with you
- Workspaces are provided and available until 5 PM CDT on May 5th.
- Enjoy this training :) and don't forget to **complete our survey!**



# DATA+AI SUMMIT 2022

HYBRID | JUNE 27-30, 2022

## Building the modern data stack on the data lakehouse

The world's largest data and AI conference returns live, to San Francisco and virtually in our new hybrid format. Four days packed with keynotes by industry visionaries, technical sessions, hands-on training and networking opportunities.

- Hear from data and AI thought leaders about latest trends and innovations, including Apache Spark, Delta Lake, MLflow, TensorFlow, dbt and many others.
- See how others are applying the data lakehouse paradigm to unify data, analytics, and AI on one platform
- Take advantage of a full range of hands-on training workshops

Organized by



# The modern data stack takes shape

Training will be held from June 27 through June 30, with an expanded curriculum of half and full day in-person and virtual classes. Most training classes will include both lecture and hands-on exercises.

NEW certification bundles are also available that include courses and exams.

GET CERTIFIED to unlock a VIP experience at during your in-person attendance

Organized by



©2021 Databricks Inc. — All rights reserved

# DATA+AI SUMMIT 2022

HYBRID | JUNE 27-30, 2022

Register Now



# Training by Personas

## Data Analyst

Data Analysis with Databricks  
Introduction to Databricks SQL

**Certification Exams**  
Associate Data Analyst  
Certification

## Data Engineer

Apache Spark Programming  
with Databricks  
Performance Tuning on Apache  
Spark  
Data Engineering with  
Databricks  
Advanced Data Engineering  
with Databricks  
End to End ETL with Databricks

**Certification Exams**  
Associate Data Engineer  
Certification

Professional Data Engineer  
Certification  
Associate Developer for Apache  
Spark 3.0

## Data Scientist

Apache Spark Programming  
with Databricks  
Advanced Machine Learning  
with Databricks  
Managing Machine Learning  
Models  
Deploying Machine Learning  
Models  
Performance Tuning on Apache  
Spark

**Certification Exams**  
Associate Machine Learning  
Certification

Professional Machine Learning  
Certification  
Professional Certified Data  
Science Certification

## Platform Admin

Databricks Platform  
Administration with Unity  
Catalog  
Databricks Lakehouse Overview  
Lakehouse with Delta Lake Deep  
Dive

**Certification Exams**  
Databricks Lakehouse Platform  
Essentials

## All Audience

Databricks Platform  
Administration with Unity  
Catalog  
Databricks Lakehouse Overview  
Lakehouse with Delta Lake Deep  
Dive



# About Me

Div Saini, Solutions Architect, Databricks

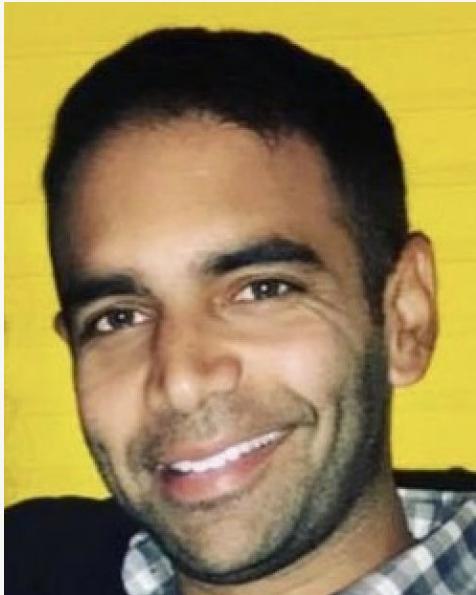


- Foodie (vegetarian)
- Love to explore new recipes (IG @milkandsaffron)
- Enjoy gardening, painting, music, traveling



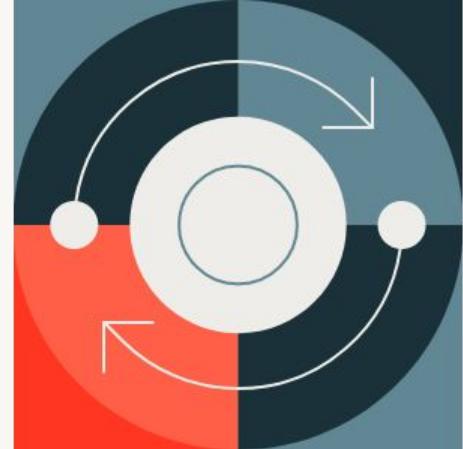
# About Me

Jai Karve, Solutions Architect, Databricks



- Having his first baby any day now
- Shellfish enthusiast
- Likes to experiment with facial hair
- Libra

# Databricks Foundations Training Series



April 20  
9AM – 12PM

Delta Lake

[REGISTER HERE](#)

April 27  
9AM – 12PM

Databricks SQL

[REGISTER HERE](#)

May 4  
9AM – 12PM

MLflow

[REGISTER HERE](#)

Complete a  
**minimum of 2**  
**sessions** &  
provide feedback  
for a certification  
voucher



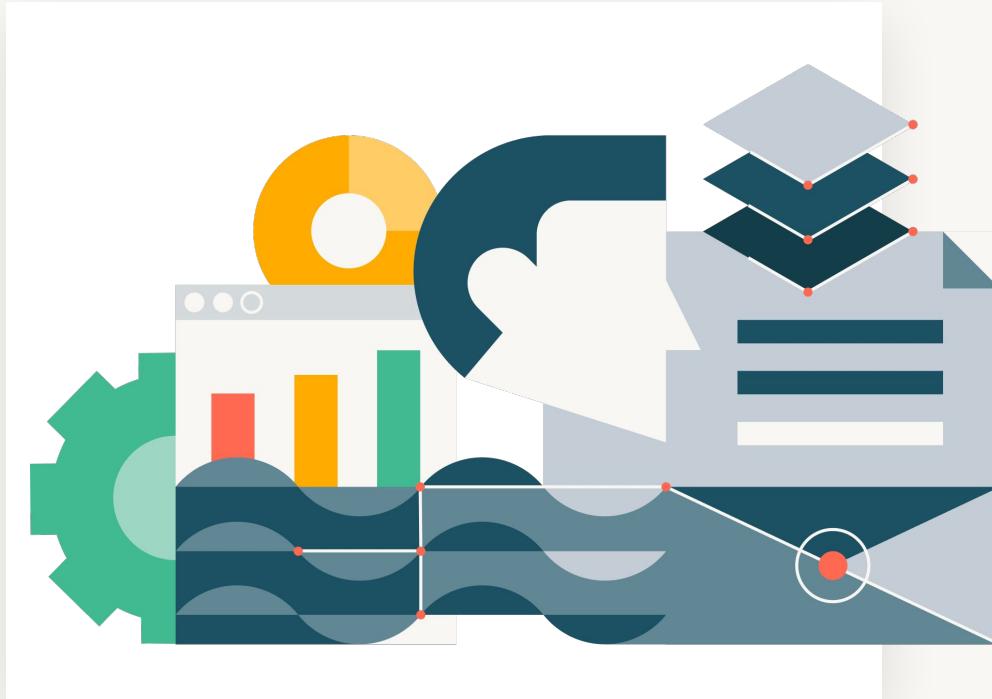
# Continue learning on Databricks Academy

New link: [Databricks Academy](#) and check our [full list of courses](#)

The screenshot shows the Databricks Academy platform. At the top, there is a navigation bar with the Databricks logo, a search bar containing "Search content in the platform", and a magnifying glass icon. Below the navigation bar, there are links for "Back" and "Home". A large banner at the top says "Welcome to Databricks training!". Below the banner, there is a "Home" button with a house icon and the text "User Home Page". The main content area features a large photograph of a diverse group of people, likely Databricks employees or attendees at a conference. Below the photo, there are two sections: "Enrolled Learning" and "My Task List". The "Enrolled Learning" section has a search bar and a grid of course cards. The "My Task List" section shows a summary of tasks: DEADLINES (1), NOT STARTED (8), IN PROGRESS (1), and ILT/WEBINAR.



# Presentation Overview



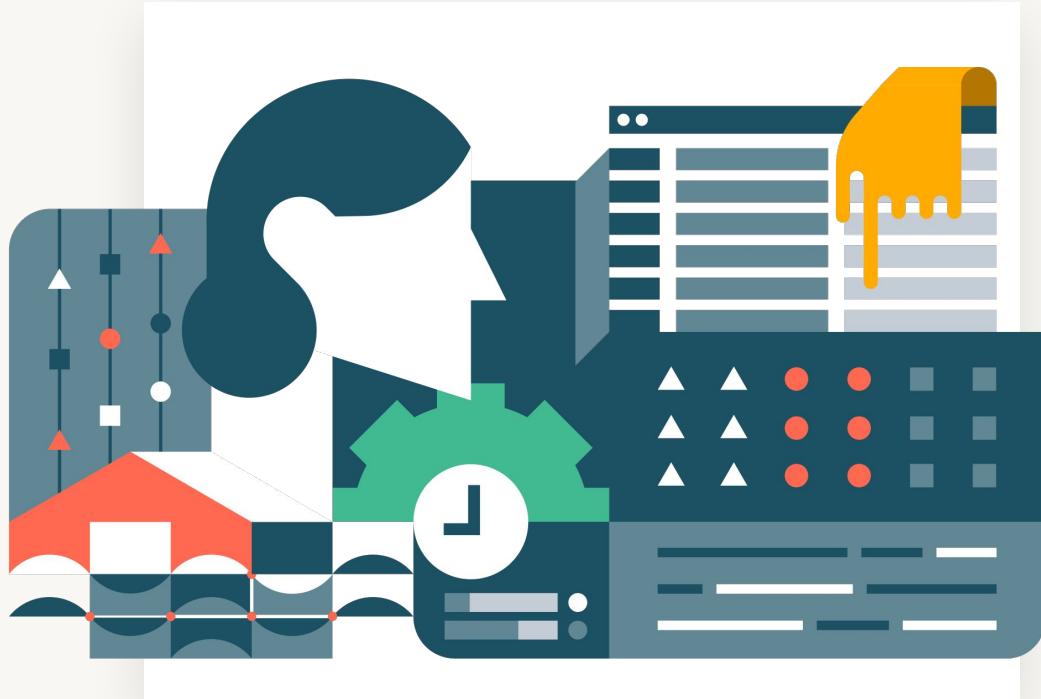
## What we'll cover

- Intro
- Data Science on Databricks
- Why MLflow matters
- How MLflow works

# Hands on Overview

## What we'll do

- Get hands-on with MLflow
- Track, register and serve models
- HyperParameter Tuning with Spark



# Let's get you set up



# Step 1: Workspace Registration

The temporary workspace you'll be using in today's workshop is administered by a third party vendor.

Complete this short registration form for the hands-on portion of the workshop.

The image shows two side-by-side screenshots. On the left is a landing page for 'Databricks Cloud Workshops'. It features a red logo icon, the text 'Databricks Cloud Workshops', 'Databricks on Azure', 'By : Databricks', and a note 'Please sign up to get access to the lab Environment'. It also shows a timer '7 hour(s) and 0 minute(s)' and an email address '\_cloudlabs@databricks.com'. On the right is a registration form titled 'Register Now' with fields for 'First Name\*', 'Last Name\*', 'Email\*', and a checkbox for agreeing to terms and privacy policy. A 'Submit' button is at the bottom.

Databricks Cloud Workshops

Databricks on Azure

By : Databricks

Please sign up to get access to the lab Environment

7 hour(s) and 0 minute(s)

\_cloudlabs@databricks.com

Azure,Databricks

EN

First Name\*

Last Name\*

Email\*

I agree to the Databricks [Terms of Service](#) and acknowledge the Databricks [Privacy Policy](#) (required).

Submit



## Step 2: Launching the workspace

Click “Launch Lab” and allow 5-10 minutes for the environment to load.

The screenshot shows a Databricks workspace interface. At the top, there's a blue header bar with the Databricks logo and a menu icon. Below the header, the text "Databricks on Azure | Jan 13th | United States" is displayed. A central message box contains the text "Please click on 'Launch Lab' button to activate your lab environment." To the right of this message is a blue rectangular button labeled "LAUNCH LAB".





# MLflow: Accelerating ML for the Enterprise

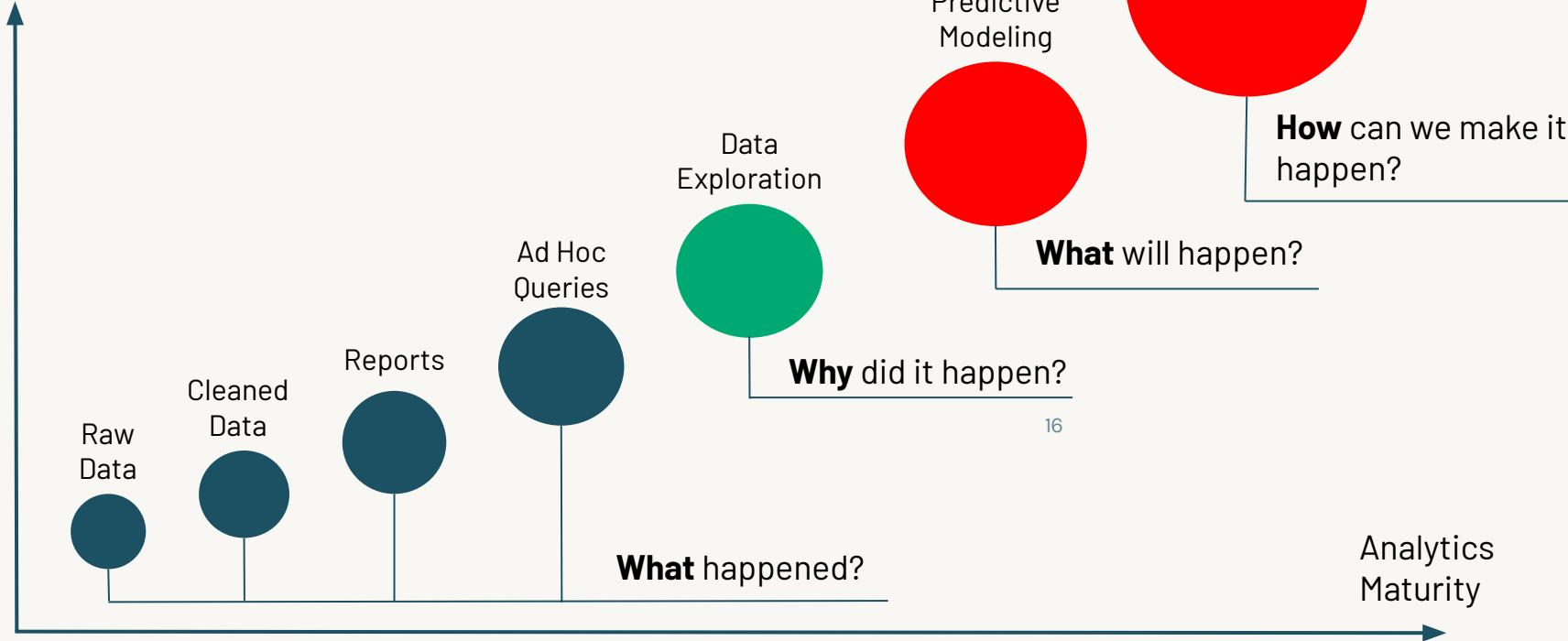
Manage the complete machine learning lifecycle with enterprise reliability, security and scale



# Data+AI Maturity Curve

From descriptive to prescriptive

Competitive  
Advantage



# The future is here

...it's just not evenly distributed

**83%**

of CEOs say AI is a strategic priority

**MIT Sloan**  
Management Review

**85%**

of big data projects fail

**Gartner**

**\$3.9T**

in business value created by AI in 2022

**Gartner**

**87%**<sup>17</sup>

of data science projects never make it into production

**VB**



# Hardest Part of ML isn't ML, it's Data

*"Hidden Technical Debt in Machine Learning Systems," Google NIPS 2015*

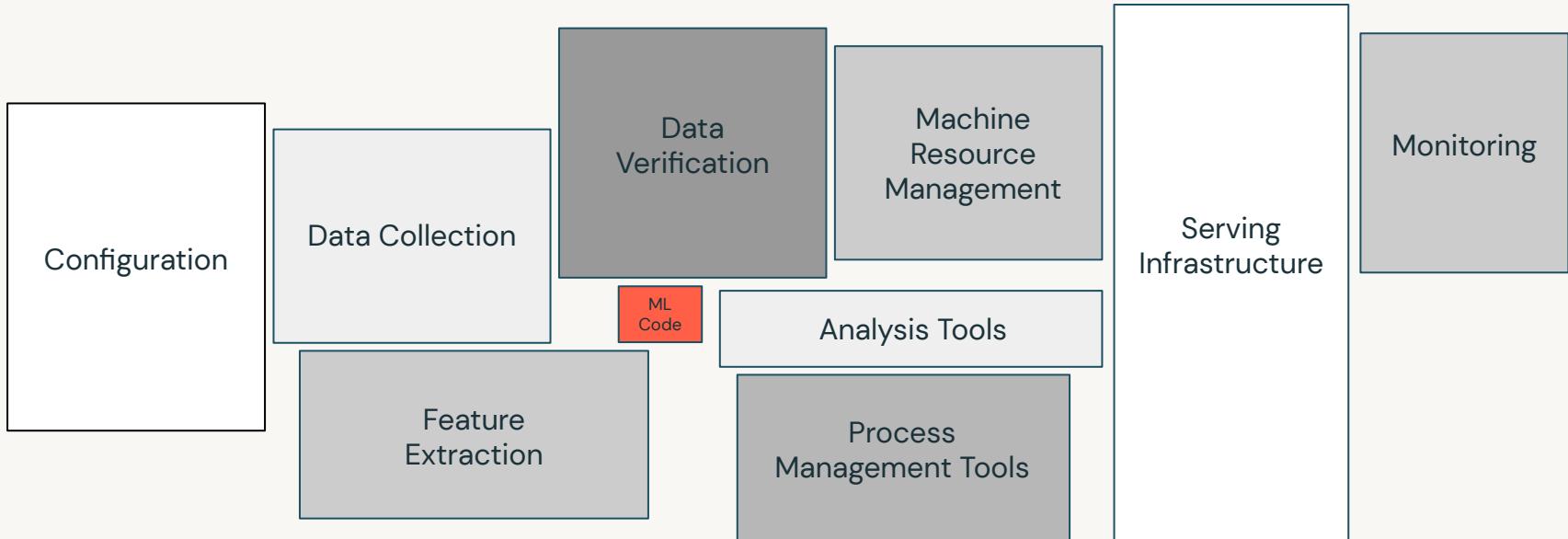


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small red box in the middle. The required surrounding infrastructure is vast and complex.



# Why are companies struggling with Data Science and ML?

1

## DATA IS NOT ML-READY

Data is siloed, fragmented, and difficult to put to work

2

## LOW DATA TEAM PRODUCTIVITY

ML projects require cross-team effort, but no good collaboration mediums exist

3

## ML IS HARD TO PRODUCTIONIZE

The machine learning lifecycle is patch-worked and doesn't scale



# Our approach:

## An open and unified platform for the full data and ML lifecycle

1

High quality data,  
readily accessible



High quality data sets discoverable  
in one place and from any source

2

Increase data  
science teams  
productivity



Databricks  
Data Science Workspace  
with ML Runtime

Use the tools, languages, and  
frameworks of choice on one  
platform

3

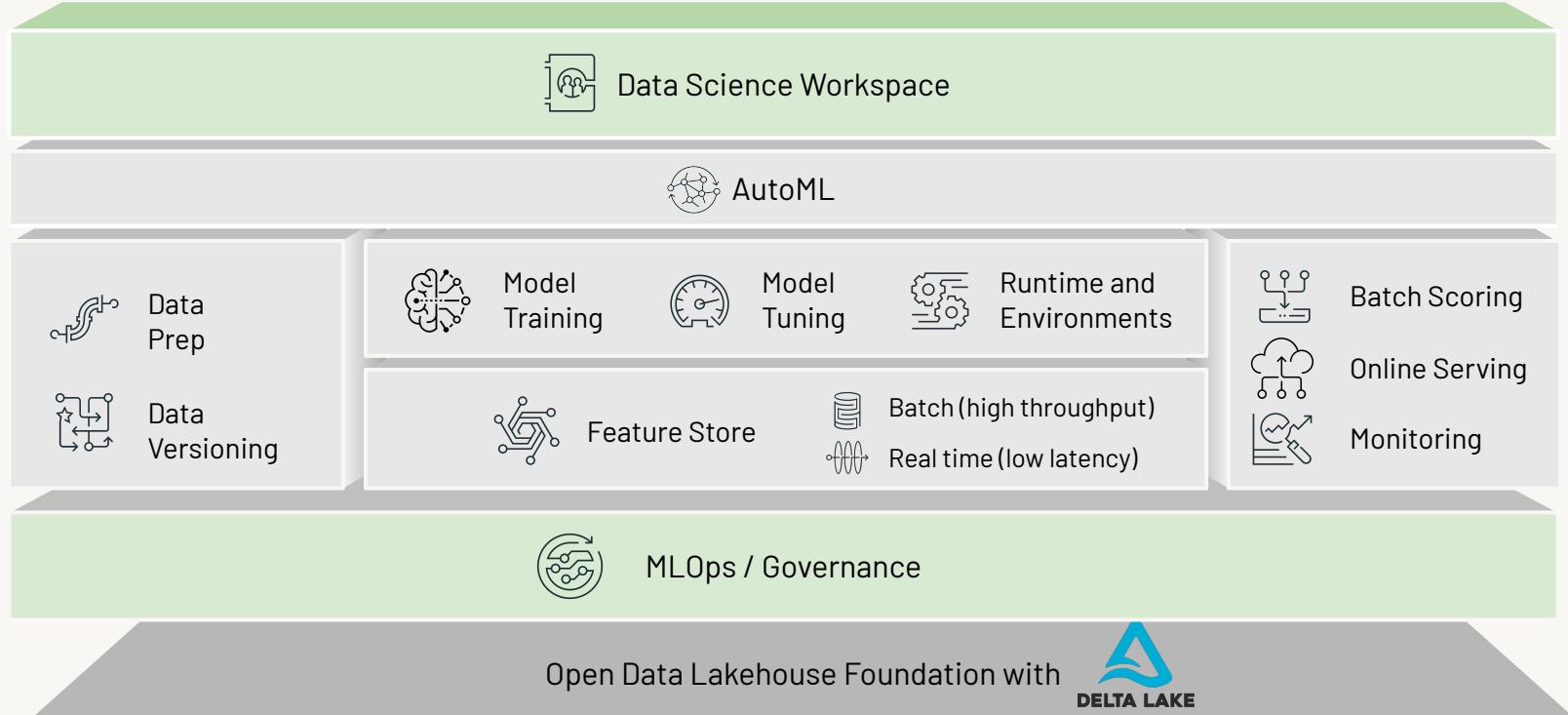
Standardize the  
full ML lifecycle



Seamlessly and securely move  
models from experimentation to  
production

# Databricks Machine Learning

A data-native and collaborative solution for the full ML lifecycle



# Full ML Lifecycle: MLOps for Data Teams

MLOps = DataOps + DevOps + ModelOps



Data  
Versioning with  
Time Travel



Code Versioning  
with Git Integration

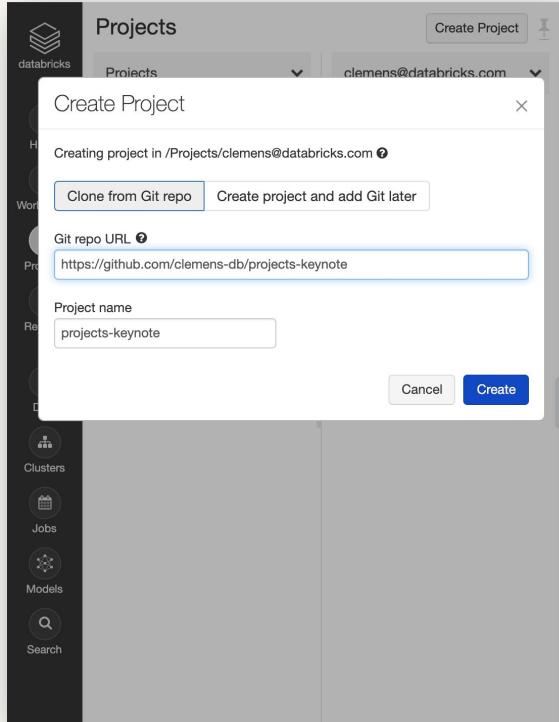


Model Lifecycle  
Management with Model  
Registry

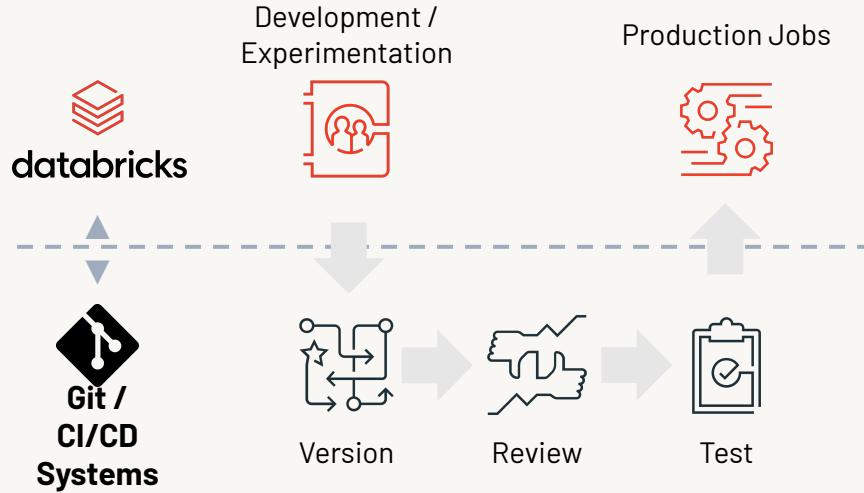


MLOps / Governance

# (Git-based) Projects



## CI/CD Integration



## Supported Git Providers



Azure DevOps



GitHub



GitLab



Bitbucket



# File support within Repos

Seamlessly bring any workload to Databricks

The screenshot shows the Databricks interface with the 'Repos' tab selected in the sidebar. The main area displays a list of repositories under the heading 'Repos'. A dropdown menu shows 'clemens.mewald@databricks.c...'. Below it, two repositories are listed: 'fb-prophet-example22' and 'samplemod'. The 'fb-prophet-example22' repository is expanded, showing its contents: 'data', 'forecasting.md', 'pdfs', 'prophet-forecasting.ipynb', 'README.md', 'requirements.txt', and 'test'. An 'Add Repo' button is located at the top right of the repository list.



- Portability of code
- Library files



- Environment specification

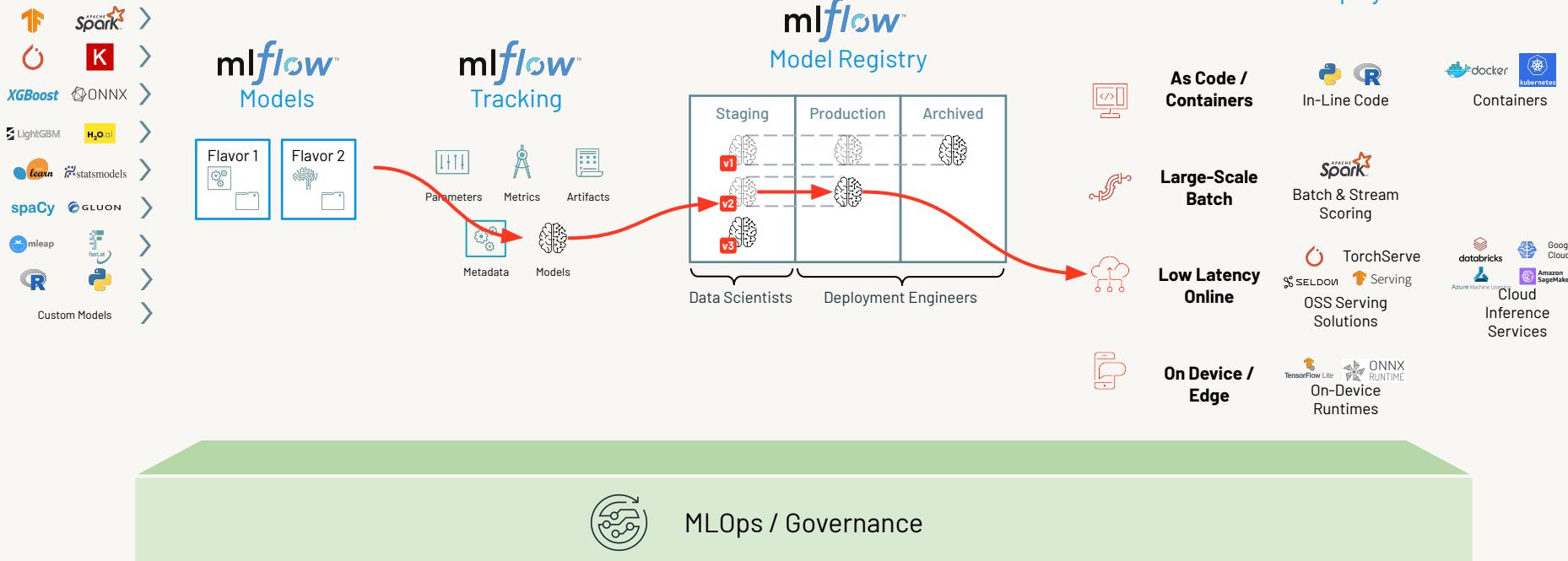


- Small data ease of use
- Relative imports



- Automate workflows with Repos API

# Full ML Lifecycle: MLOps for Data Teams



# Exercise #1: Workspace Access



# Step 3: Accessing the Workspace

Copy-paste the provided Databricks Workspace URL into an incognito window.

Tip: Using an incognito window prevents you from logging into an existing workspace. If issues persists, try disconnecting from VPN and trying again.

Environment Details		
Fields	Credentials	Action
Username	odl_user_512944@databrickslabs.com	
Password	wpsj11LAU*kn	
Resource Group : 20438		
Key	Value	Action
Databricks Workspace URL	https://adb-766847864694359.19.azure.databricks.net	



# Step 4-5: Signing in on Azure

Sign in with the provided username and password from the credentials page

Select **Sign in with Azure AD**



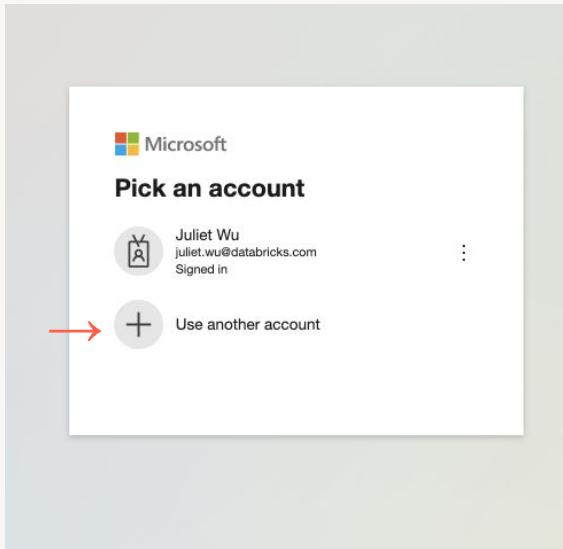
Sign In to Databricks

Sign in using Azure Active Directory Single Sign On. [Learn more](#)

 **Sign in with Azure AD**

Contact your site administrator to request access.

Select **Use another account**



Copy-paste **provided credentials**



**Sign in**

Email, phone, or Skype

[Can't access your account?](#)

Back

Next

 [Sign-in options](#)

# Arrive in your Databricks Workspace

Find the notebook under the Workspace tab and within your User folder.

1. From the navigation panel on the left side, select **Workspace**
2. Select **Users** to find your folder of course materials

The screenshot shows the Databricks homepage. On the left, a dark sidebar navigation bar includes icons for Home, Workspace (circled in red with an arrow pointing to it), Recents, Data, and Clusters. The main area features the "Welcome to databricks" header, a "Explore the Quickstart Tutorial" button, and a "Import & Export" section with a "Drop files or click" button. A dashed box highlights the "Import & Export" section.

The screenshot shows the Databricks workspace interface. The left sidebar has the same navigation as the homepage. The main workspace area has two dropdown menus: "Workspace" (set to "Shared") and "Users" (set to "odl\_user\_354301@databrickslab..."). Both dropdowns have a "Users" option highlighted with a red box. A dashed box highlights the "Users" dropdown. To the right, there is a preview of a notebook titled "KS" with a "Drop files or click" button and an "Import" section.

# Break – return in 5 mins



# MLFlow Part 1

## Feature Store

## Tracking

## Hyperparameter Tuning

## AutoML



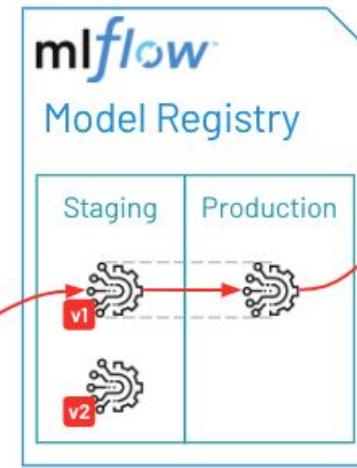
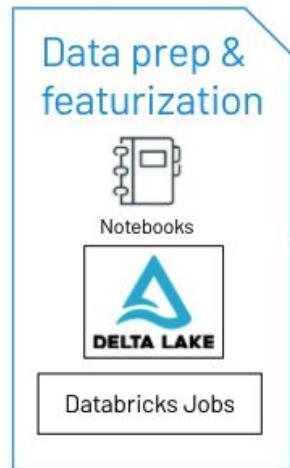
# The Full ML Lifecycle

**Data Scientists** build features.  
**Data Engineers** provide infra for automating featurization.

**Data Scientists** build models and log them to MLflow, which records environment info.

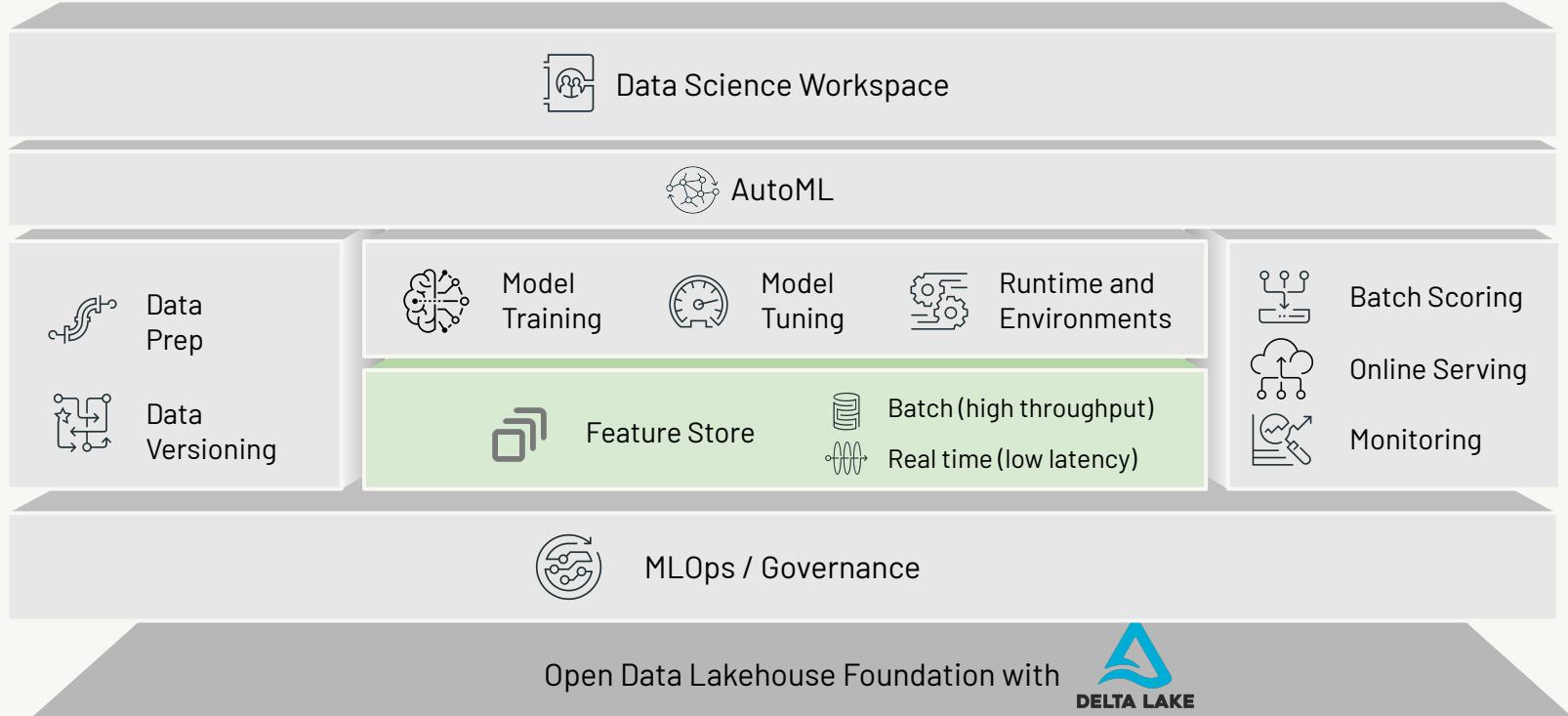
**Data Scientists** move models to Staging.

**Deployment Engineers** manage CI/CD tools which promote models to Production.

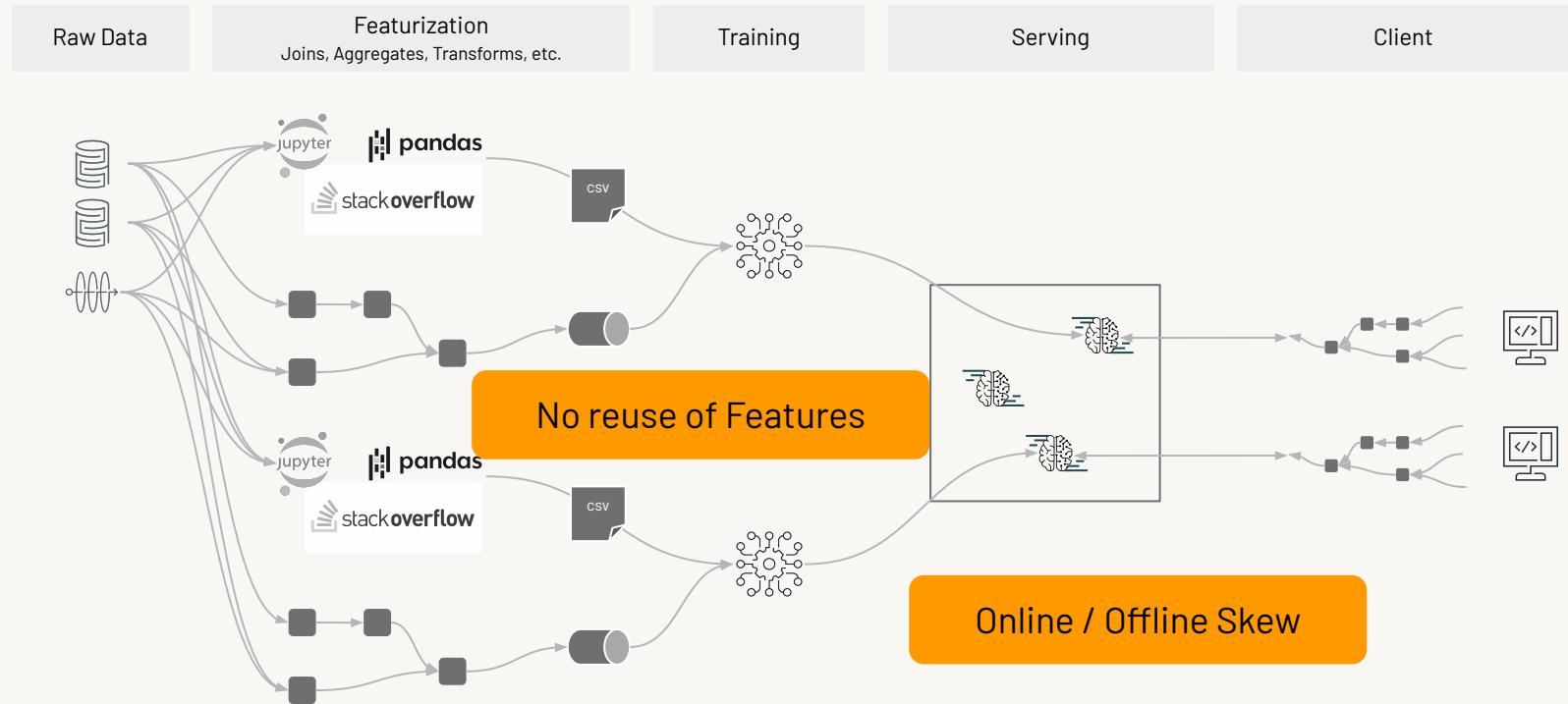


# Feature Store

The first Feature Store codesigned with a Data and MLOps Platform

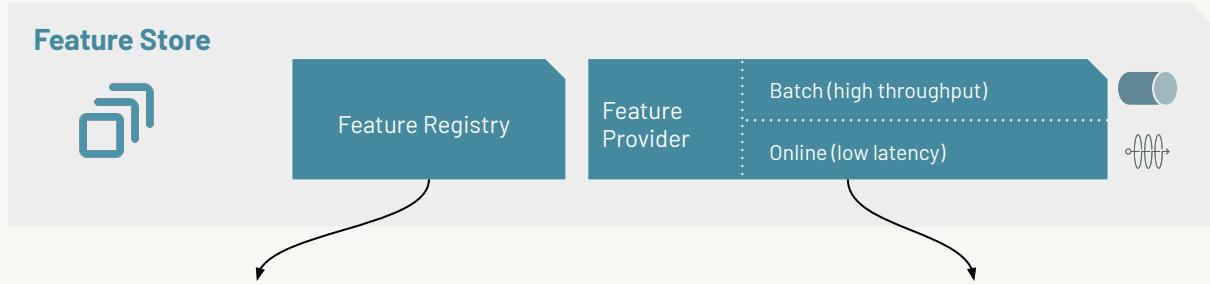


# A day (or 6 months) in the life of an ML model



# Feature Store

The first Feature Store codesigned with a Data and MLOps Platform



## Feature Registry

- Discoverability and Reusability
- Versioning
- Upstream and downstream Lineage

## Feature Provider

- Batch and online access to Features
- Feature lookup packaged with Models
- Simplified deployment process

Co-designed with  DELTA LAKE

- Open format
- Built-in data versioning and governance
- Native access through PySpark, SQL, etc.

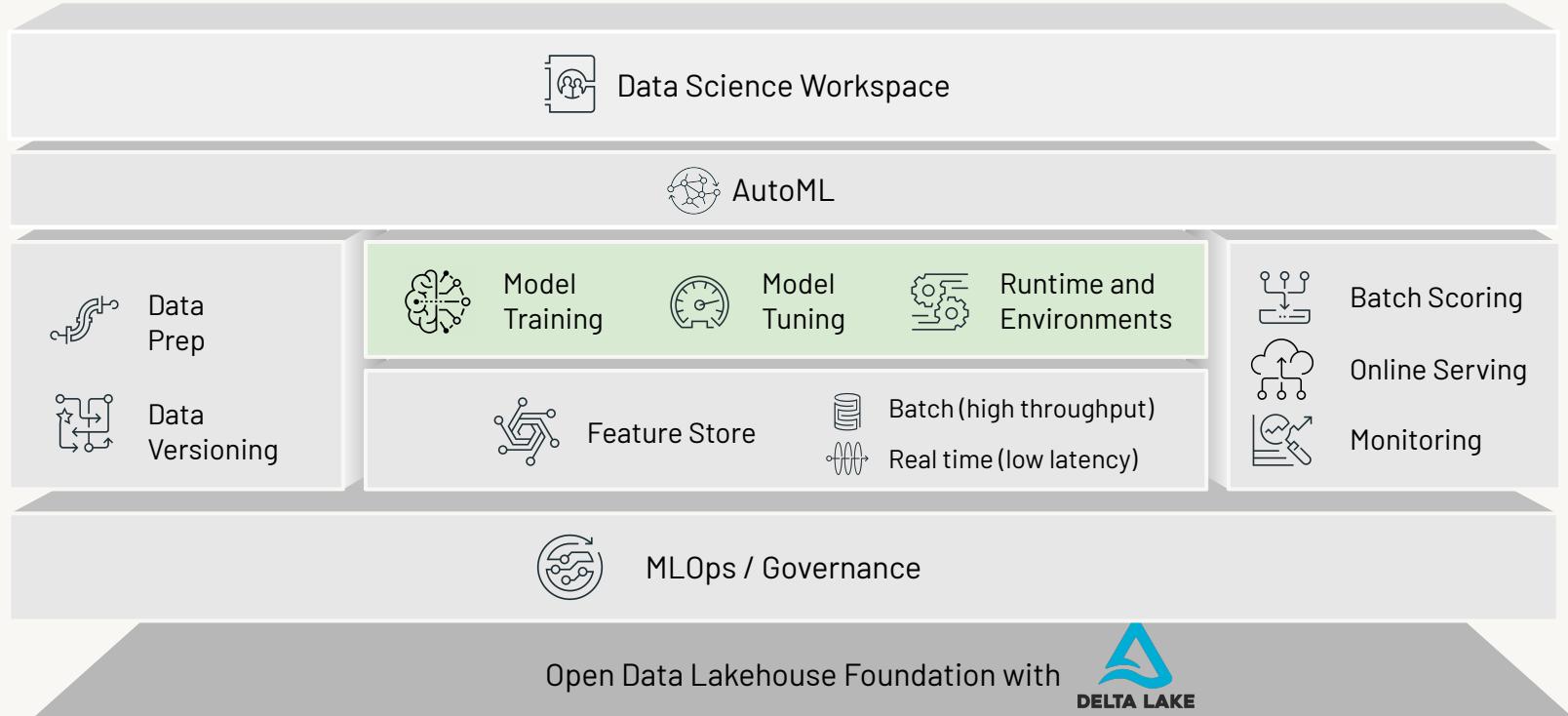
Co-designed with  mlflow™

- Open model format that supports all ML frameworks
- Feature version and lookup logic hermetically logged with Model

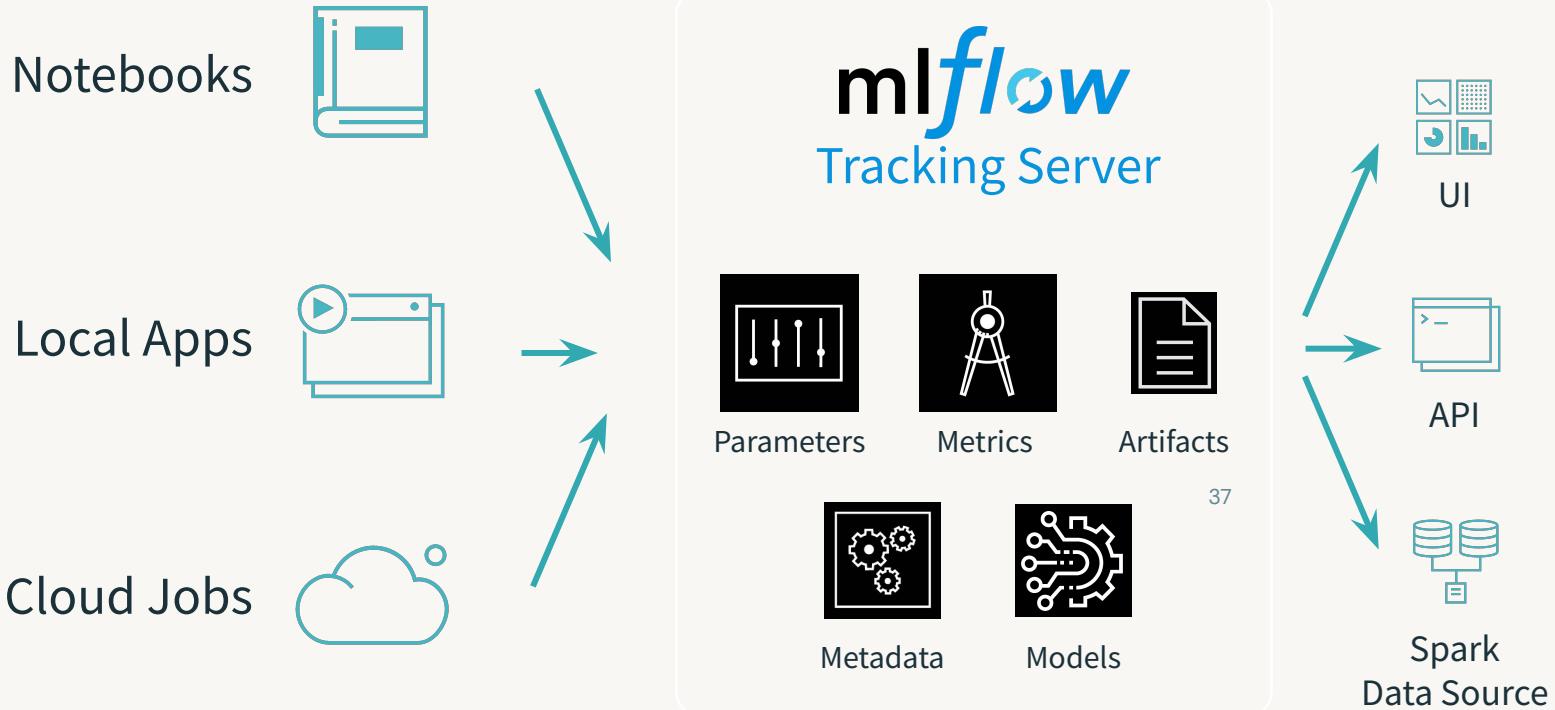


# Databricks Machine Learning

A data-native and collaborative solution for the full ML lifecycle



# mlflow Tracking



# Key Concepts in Tracking

**Parameters:** key-value inputs to your code

**Metrics:** numeric values (can update over time)

**Artifacts:** arbitrary files, including models

**Source:** what code ran?

The screenshot shows a Databricks experiment tracking interface. At the top, there's a URL bar with the path /Users/alexey.ott@databricks.com/Demos/MLOps Workshop/Azure MLOps demo/train\_model and a 'Preview' button. Below it, the 'Experiment ID' is listed as 3479448620313166. A navigation bar includes 'Description' and 'Edit' buttons, along with 'Refresh', 'Compare', 'Delete', 'Download CSV', and time filters ('Start Time' and 'All time'). There are also 'Columns', 'Only show differences' (which is turned off), a search bar containing the query 'metrics.rmse < 1 and params.model = "tree"', and a 'Filter' button. The main area displays a table titled 'Showing 4 matching runs'. The table has columns: Start Time, Duration, Run Name, User, Source, Version, Models, and mae. The data rows are: 1 day ago (10.9s, -), 2 days ago (9.0s, -), 9 months ago (4.6s, -), and 1 year ago (8.4s, -). All runs were performed by user alexey.ott@... using 'train\_mod' source and 'sklearn' models, with an mae of 0.617. A 'Metrics' button is at the top right of the table. At the bottom right of the table area, there's a 'Load more' button.

	Start Time	Duration	Run Name	User	Source	Version	Models	mae
<input type="checkbox"/>	1 day ago	10.9s	-	alexey.ott@...	train_mod	-	aott-wine-.../3	0.617
<input type="checkbox"/>	2 days ago	9.0s	-	alexey.ott@...	train_mod	-	aott-wine-.../2	0.617
<input type="checkbox"/>	9 months ago	4.6s	-	alexey.ott@...	train_mod	-	aott-wine-.../1	0.617
<input type="checkbox"/>	1 year ago	8.4s	-	alexey.ott@...	train_mod	-	sklearn	0.617

# mlflow Tracking

Parameters

```
2020-04-06 10:14:06 PDT
batch_size: 64
class_weight: None
epochs: 100
epsilon: 1e-07
initial_epoch: 0
learning_rate: 0.001
max_queue_size: 10
num_layers: 2
optimizer_name: Adam
sample_weight: None
shuffle: True
steps_per_epoch: None
use_multiprocessing: False
validation_freq: 1
validation_split: 0.2
validation_steps: None
workers: 1
```

loss: 1118149.3  
val\_loss: 890218

Points:

Off

Line Smoothness

0.00

X-axis:

- Step
- Time (Wall)
- Time (Relative)

Y-axis:

loss x val\_loss x

- loss
- val\_loss



Artifacts

model

- data
- MLmodel
- conda.yaml
- cluster\_spec.json
- conda.yml
- model\_report.pdf
- model\_summary.txt
- plot.png
- validation.csv

Models

Artifacts

Experiment Tracking



XGBoost



APACHE Spark



LightGBM

# "Manual" Logging

```
with mlflow.start_run() as run:  
    lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio)  
    lr.fit(train_x, train_y)  
    predictions = lr.predict(test_x)  
    (rmse, mae, r2) = eval_metrics(test_y, predictions)  
    mlflow.log_param("alpha", alpha)  
    mlflow.log_metric("rmse", rmse)  
    mlflow.sklearn.log_model("model", lr)
```

...

40

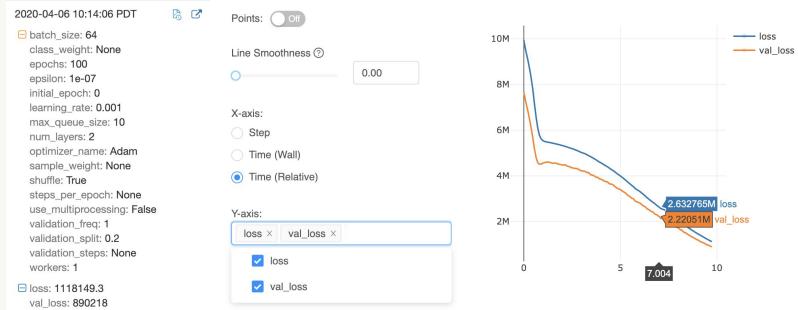


# Auto-Logging

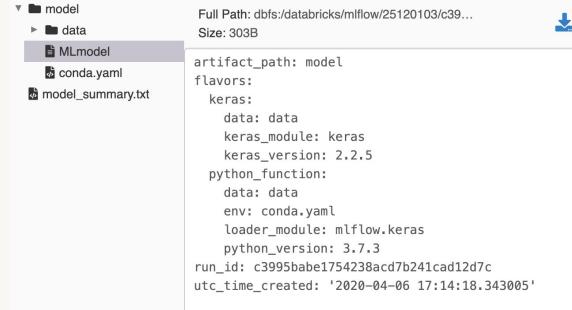
Auto-logging for ML Frameworks: A single line of code logs parameters, metrics, and artifacts.

```
mlflow.autolog() # or: mlflow.keras.autolog()
```

Parameters and (a time series of) metrics

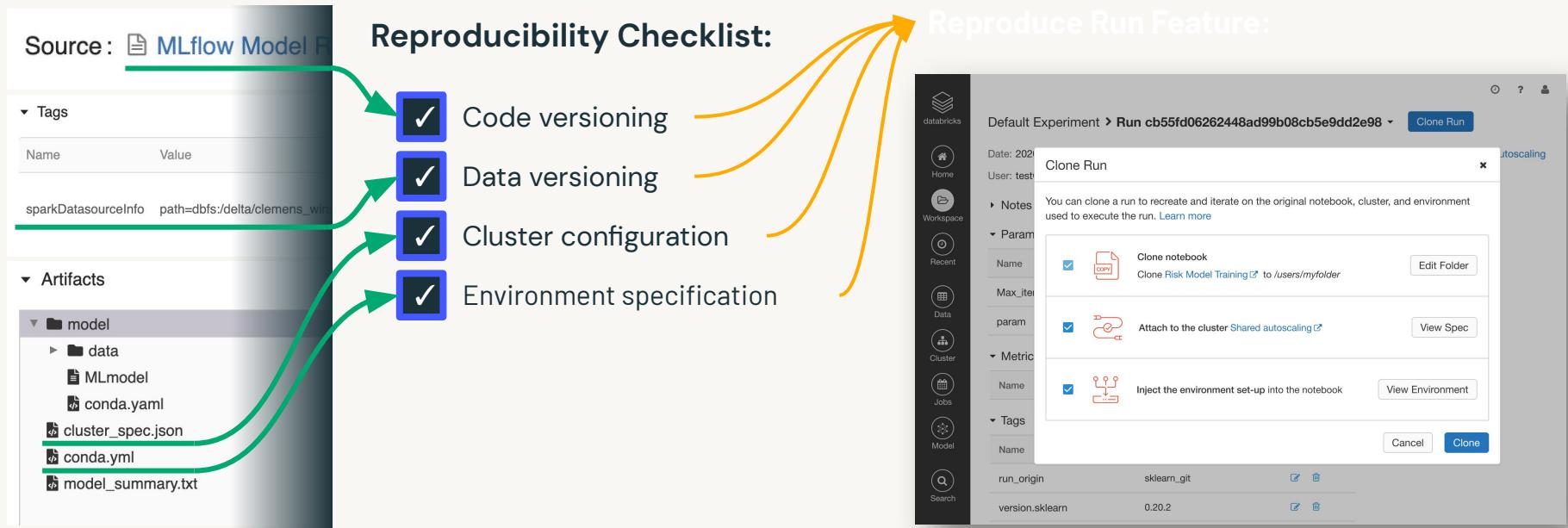


Artifacts (including model)



# Auto-Logging for Reproducibility

## Auto-Logging of Cluster Specification and Environment Dependencies

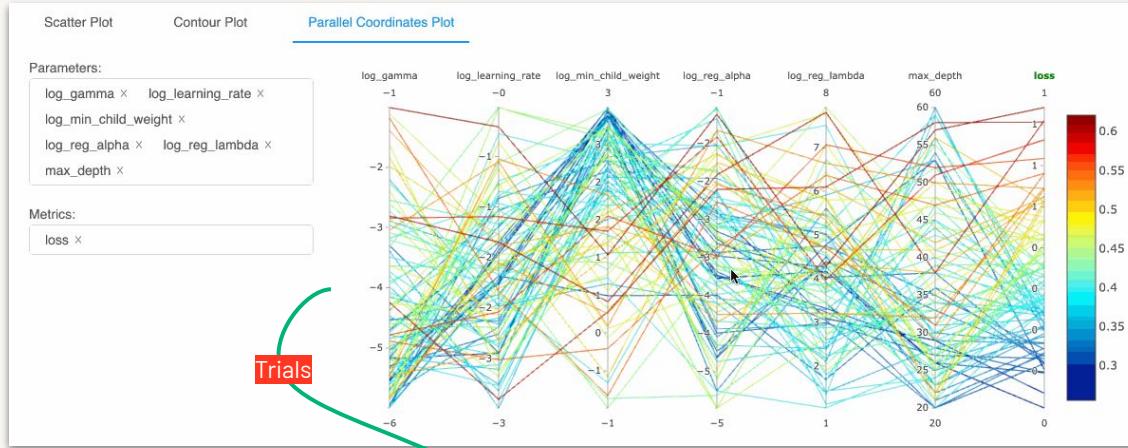


# Hyperparameter tuning with HyperOpt

- **HyperOpt**: Python library, Bayesian optimizer for assessing a model's accuracy over a space of hyperparameters
- Integrates with Spark for parallel hyperparameter search
- Integrates with MLflow for tracking
- Included in [ML runtime](#)
- [Tuning your Model with HyperOpt](#)



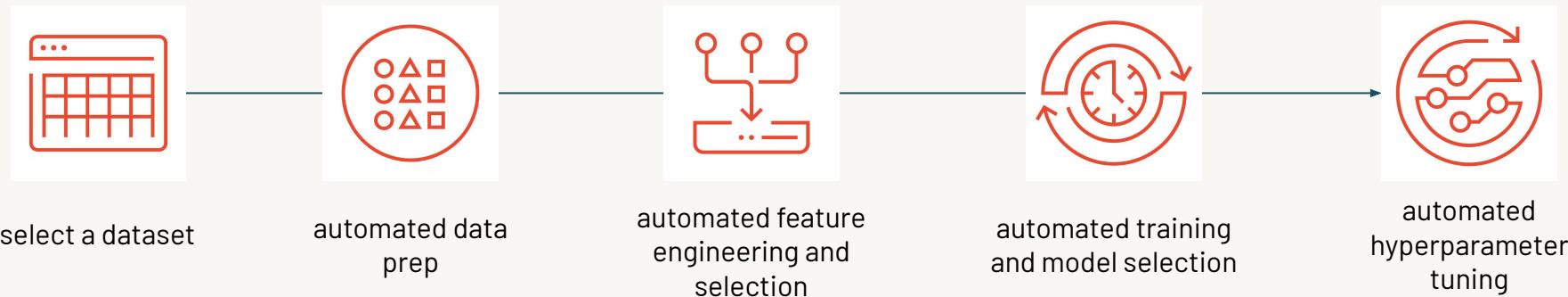
# Automated capture of Hyperparameter Search



Experiment Tracking

# What is AutoML?

**Automated machine learning (AutoML)** is a fully-automated model development solution seeking to “democratize” machine learning. While the scope of the automation varies, AutoML technologies usually automate the ML process from data to model selection.



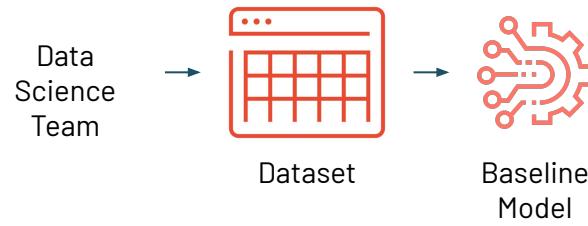
# AutoML solves two key pain points for data scientists

## Quickly Verify the Predictive Power of a Dataset



*"Can this dataset be used to predict customer churn?"*

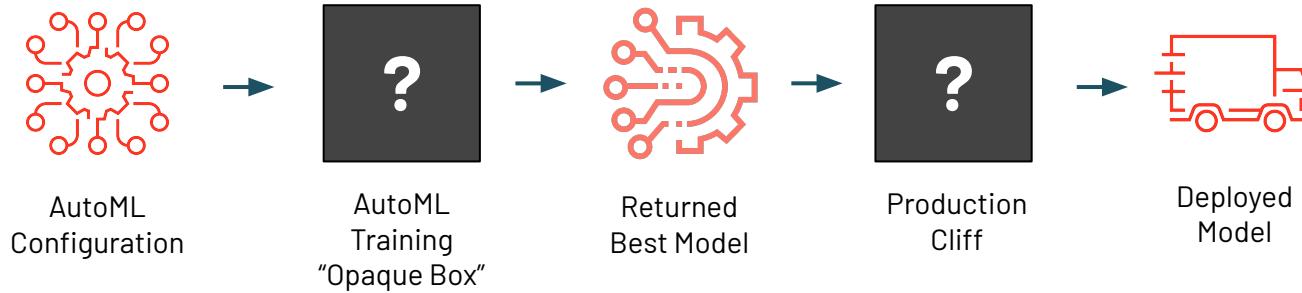
## Get a Baseline Model to Guide Project Direction



*"What direction should I go in for this ML project and what benchmark should I aim to beat?"*

# Problems with existing AutoML solutions

## Opaque-Box and Production Cliff Problems in AutoML



Problem	Result / Pain Points
<ol style="list-style-type: none"><li>1. A “production cliff” exists where data scientists need to modify the returned “best” model using their domain expertise before deployment</li><li>2. Data scientists need to be able to explain how they trained a model for regulatory purposes (e.g., FDA, GDPR, etc.) and most AutoML solutions have “opaque box” models</li></ol>	<ul style="list-style-type: none"><li>• The “best” model returned is often not good enough to deploy</li><li>• Data scientists must spend time and energy reverse engineering these “opaque-box” returned models so that they can modify them and/or explain them</li></ul>



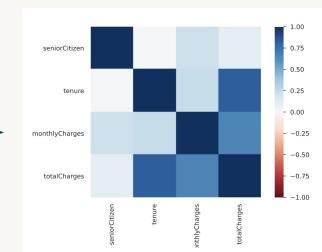
# Databricks AutoML

A glass-box solution that empowers data teams without taking away control

UI and API to start AutoML training

The screenshot shows the 'Configure AutoML experiment' page. It includes a navigation bar with 'Configure', 'Preview', and 'Profile' tabs. Below the tabs, it says 'Experiments > Configure AutoML experiment'. There are three steps: '1 Configure', '2 Augment', and '3 Train'. Step 1 is selected. Under 'AutoML Experiment Configuration', there is a section for 'Compute' with a dropdown menu showing 'dais\_mir\_8\_new'. A note at the bottom says 'Select an existing cluster with a Databricks Runtime for ML 8.0+ or later'.

Start Time	Run Name	User	Source
2021-05-05 1	logistic_r...	kase...	Notebo...
2021-05-05 1	logistic_r...	alkis...	21-05...
2021-05-05 1	logistic_r...	alkis...	21-05...
2021-05-05 1	logistic_r...	kase...	Notebo...
2021-05-05 1	logistic_r...	kase...	Notebo...
2021-05-05 1	decision...	kase...	Notebo...
2021-05-05 1	random_f...	kase...	Notebo...



## MLflow experiment

Auto-created MLflow Experiment to track models and metrics

Easily deploy to Model Registry

The screenshot shows a 'Generated Trial Notebook (Python)' titled 'Generated Trial Notebook (Python)'. It contains code for 'Random Forest training' including 'Load Data', 'Preprocessors', 'Numerical columns', 'One-hot encoding', 'Feature standardization', and 'Train classification mo...'. The code uses Jupyter Notebook syntax with cells containing code and output.

## Reproducible trial notebooks

Generated notebooks with source code for every model

Understand and debug data quality and preprocessing

Iterate further on models from AutoML, adding your expertise



## Configure

Configure AutoML experiment [Preview](#) [Provide Feedback](#)

Experiments > Configure AutoML experiment

1 Configure — 2 Augment — 3 Train — 4 Evaluate

AutoML Experiment Configuration

\* Compute  
dais\_mir\_8\_new

Select an existing cluster with a Databricks Runtime for ML 8.0+ or [create a new cluster](#)

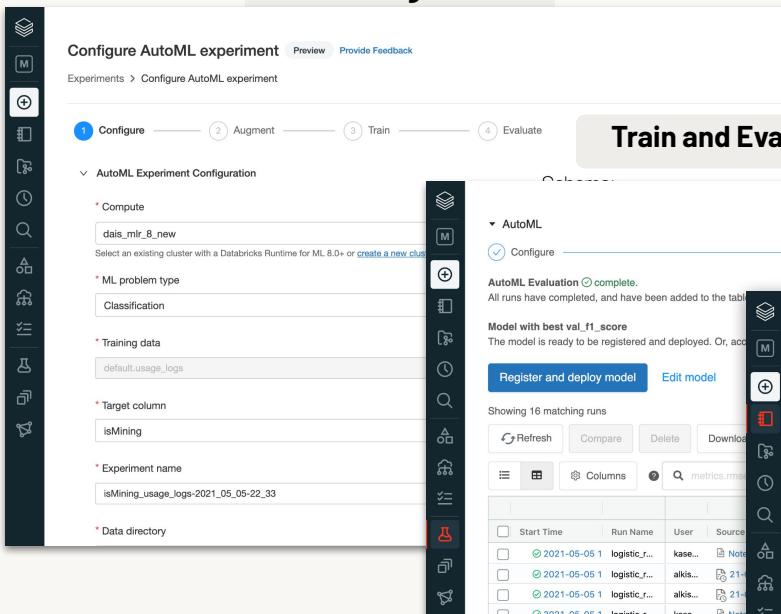
\* ML problem type  
Classification

\* Training data  
default\_usage\_logs

\* Target column  
isMining

\* Experiment name  
isMining\_usage\_logs-2021\_05\_05-22\_33

\* Data directory



## Train and Evaluate

AutoML Evaluation [complete](#).  
All runs have completed, and have been added to the table below.

Model with best val\_f1\_score

The model is ready to be registered and deployed. Or, acc...

Register and deploy model [Edit model](#)

Showing 16 matching runs

Start Time	Run Name	User	Source
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	alkis...	21...
2021-05-05 1	logistic_r...	alkis...	21...
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	decision...	kase...	Notebook
2021-05-05 1	random_f...	kase...	Notebook
2021-05-05 1	decision...	kase...	Notebook
2021-05-05 1	random_f...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook

Generated Trial Notebook (Python)

```
# Choose any prediction to explain, or sample multiple examples for more thorough results.  
example = X_val.sample(n=25)  
  
# Use Kernel SHAP to explain feature importance on example  
predict = lambda x: model.predict(x).DataFrame().columns  
explainer = KernelExplainer(predict, X_train)  
shap_values = explainer.shap_values[example]  
summary_plot(shap_values, X_train, example)  
except Exception as e:  
    print(f"An unexpected error occurred: {e}")
```

Details Serving

Status: Ready - Stop Cluster: milflow-model-dais\_demo

Model Versions

Version 2 Ready

Model URL: [https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais\\_demo/2/invocations](https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais_demo/2/invocations)  
[https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais\\_demo/2/Production/invocations](https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais_demo/2/Production/invocations)

Call The Model

Browser Curl Python

Request

```
[{"notebookLanguage": "python", "cpu": "3", "ip_address": "1587714725935792, "account_id": "1614074788455, "memberStartTime": "2021-05-06 03:18:46 +0000"}]
```

Response

```
[true,true,true,true]
```

Send Request Show Example

Logs Version Events

```
$(GUNICORN_CMD_ARGS) --milflow-pyfunc.scoring._server.wsgi:app  
[2021-05-06 03:18:46 +0000] [8143] [INFO] Starting gunicorn 20.1.0
```

# "Glass-Box" AutoML with a UI

# “Glass-Box” AutoML with an API

The screenshot illustrates the Databricks AutoML workflow. On the left, the AutoML interface shows a completed evaluation with 16 matching runs listed in a table. A blue arrow points from the 'Train' button in the AutoML interface down to the 'Generated Trial Notebook' section on the right.

`databricks.automl.classify(df, target_col='label', timeout_minutes=60)`

AutoML Evaluation complete.  
All runs have completed, and have been added to the table below. Click a specific run to view details or review the [data exploration notebook](#).

Model with best val\_f1\_score  
The model is ready to be registered and deployed. Or, access the source code for the model training to make modifications by clicking a notebook under the Source code section.

Register and deploy model Edit model

Showing 16 matching runs

Start Time	Run Name	User	Source	Version	Models	Parameters >	Metrics >
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi...	-	pyfunc	Logi...	0.0... - 1 1 1 0.00
2021-05-05 1	logistic_r...	alkis...	21-05-05-18:56-Logisti...	-	pyfunc	Logi...	0.0... - 1 1 1 9.9...
2021-05-05 1	logistic_r...	alkis...	21-05-05-18:56-Logisti...	-	pyfunc	Logi...	0.0... - 1 1 1 0.00
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi...	-	pyfunc	Logi...	0.0... - 1 1 1 0.01
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi...	-	pyfunc	Logi...	0.0... - 1 1 1 0.00
2021-05-05 1	decision_...	kase...	Notebook: DecisionTree	-	pyfunc	Deci...	- - 0.999 0.999 0.00
2021-05-05 1	random_f...	kase...	Notebook: RandomForest	-	pyfunc	Ran...	- False 0.999 0.999 0.04
2021-05-05 1	decision_...	kase...	Notebook: DecisionTree	-	pyfunc	Deci...	- - 0.999 0.999 0.00

Generated Trial Notebook [Python]

```
# Choose any prediction to explain, or sample multiple examples for more thorough results.
example = X_val.sample(n=25)

# Use Kernel SHAP to explain feature importance on example from validation set
predict = lambda x: model.predict(pd.DataFrame(x, columns=X_train.columns))
explainer = KernelExplainer(predict, train_sample, link="identity")
shap_values = explainer.shap_values(example, l1_reg=False)
summary_plot(shap_values, example)

except Exception as e:
    print(f"An unexpected error occurred while plotting feature importance using SHAP: {e}")

100% |██████████| 25/25 [03:03<00:00, 7.36s/it]
```

SHAP value (impact on model output)

The SHAP summary plot displays the impact of various features on the model's output. The x-axis represents the SHAP value (impact on model output) ranging from -0.4 to 1.0, and the y-axis represents the Feature value. The plot includes numerous colored dots representing individual data points and a blue line representing the overall trend. Features listed on the x-axis include: \_\_ip\_address\_\_online\_feature\_lookup\_demo\_ip\_features\_\_country, \_\_account\_id\_\_online\_feature\_lookup\_demo\_account\_features\_\_email\_domain, \_\_account\_id\_\_online\_feature\_lookup\_demo\_account\_features\_\_event\_count, eventTime, \_\_ip\_address\_\_online\_feature\_lookup\_demo\_ip\_features\_\_isp, memberRecency, account\_id, memberStartTime, ip\_address, notebookLanguage, os, and cpu.



# Exercise #2



# Break – return in 5 mins



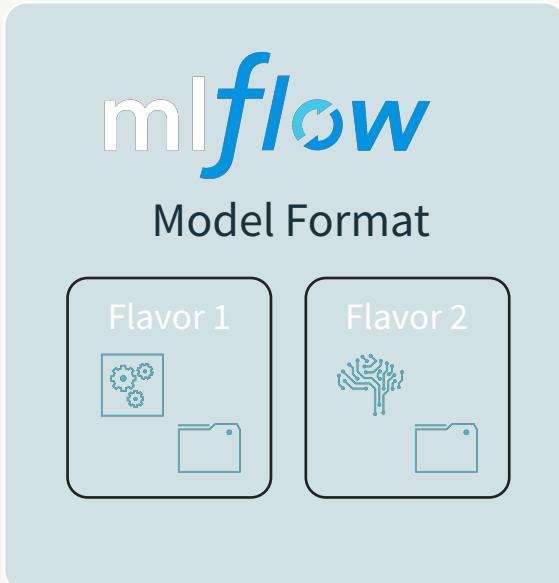
# MLflow Part 2

## MLflow Models

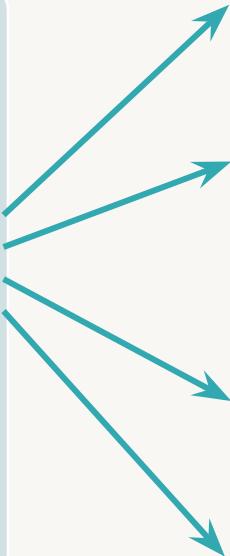
## MLflow Registry



# mlflow Models



Simple model flavors  
usable by many tools



In-Line Code



Containers



Batch & Stream Scoring



Cloud Inference Services

# Example MLflow Model

```
my_model/
    └── MLmodel
        ├── run_id: 769915006efd4c4bbd662461
        ├── time_created: 2018-06-28T12:34
        ├── flavors:
        │   ├── tensorflow:
        │   │   ├── saved_model_dir: estimator
        │   │   ├── signature_def_key: predict
        │   │   ├── python_function:
        │   │   │   └── loader_module: mlflow.tensorflow
        │   └── ...
        └── estimator/
            ├── saved_model.pb
            └── variables/
                ...
            ...
```

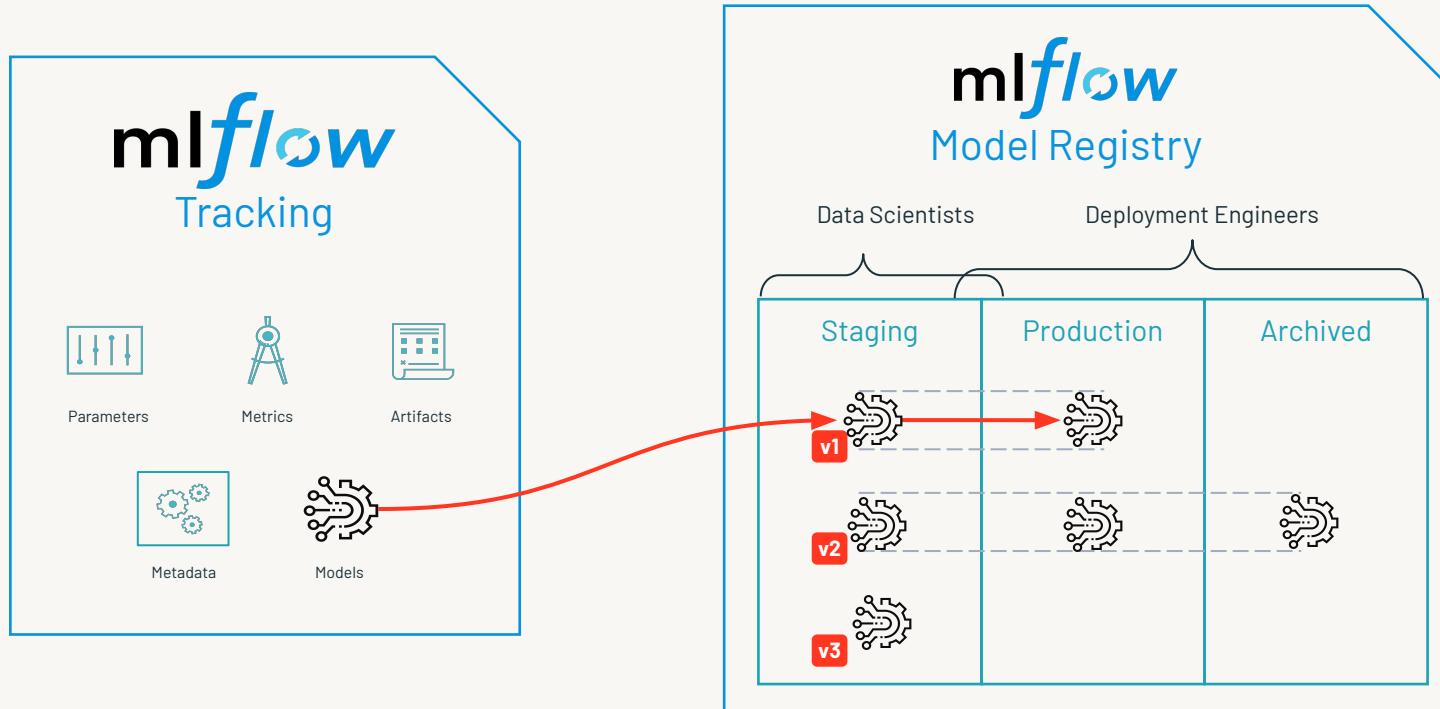
- } Usable by tools that understand TensorFlow model format
- } Usable by any tool that can run Python (Docker, Spark, etc!)

55



# mlflow Model Registry

VISION: Centralized and collaborative model lifecycle management



# One Collaborative Hub

The screenshot shows the Databricks interface for managing machine learning models. On the left, a vertical navigation bar lists various categories: Machine Learning, Create, Workspace, Repos, Recents, Search, Data, Compute, Jobs, Experiments, Feature Store, and Models. The 'Models' option is highlighted with a red circle and the number '1'. The main area is titled 'Registered Models' and contains a table with the following data:

Name	Latest Version	Staging	Production
airbnb_hawaii	Version 3	Version 3	Version 2
alexey_ott_dns_dga	Version 8	-	-
aott-wine-model	Version 3	Version 2	-
field_demos_customer_churn	Version 7	Version 7	-
mlver-model1	Version 12	-	-
mlver-model2	Version 8	-	-

A red box highlights the row for 'aott-wine-model', which is also circled with the number '2'. A red circle with the number '3' points to the search bar at the top right of the table.

## Pain Points

- Models are stored in random places, not discoverable
- No visibility into what models are in production, etc.

## Features

1. Left-nav menu: Models
2. Overview of registered models and their <sup>57</sup> versions that are in Staging and Production
3. Search and filter models by name, stage, etc.



# Manage the entire Model Lifecycle (ModelOps)

The screenshot shows the Databricks Model Registry interface for the model 'Airline\_Delay\_SparkML'. It displays two main sections: a list of active model versions and a stage transition interface.

**1. Model Versions:** A table showing active versions. Version 5 is in Production, and Version 6 is in Staging. A red box highlights this section, and a red arrow points from it to the Stage transition interface on the right.

Version	Registered at	Created by	Stage	Pending Requests
Version 5	2019-10-11 12:44:44	clemens.mewald@databricks.com	Production	-
Version 6	2019-10-16 03:15:56	clemens.mewald@databricks.com	Staging	1

**2. Stage Transition:** A panel titled 'Stage: Staging' showing pending requests to transition the model to Production. A red box highlights the 'Request transition to Production' button, which is also highlighted with a blue background. A red arrow points from the 'Pending Requests' in the version table to this transition panel.

## Pain Points

- No insight into the deployment stage of models
- Anyone can put models into production without any checks and balances

## Features

1. Overview of active model versions and their deployment stage
2. Request/Approval workflow for transitioning deployment stages



# Visibility

## Registered Models > Airline\_Delay\_SparkML > Version 5 ▾

Registered At: 2019-10-11 12:44:44

Creator: clemens.mewald@databricks.com Stage: Production ▾

Last Modified: 2019-10-22 09:03:28

Source Run: [Run 6151fe768a5e49d39076b07448e60d57](#)

### ▼ Description

Improved the Airline delay model using a GBDT. See run for improved metrics.

### 1 Pending Requests

#### Activities

 clemens.mewald@databricks.com applied a stage transition None → Production 11 days ago

What can go wrong?

 clemens.mewald@databricks.com requested a stage transition Production → None 8 days ago

 clemens.mewald@databricks.com rejected a stage transition → None 8 days ago

## Pain Points

- No visibility into who put a model into production

## Features

1. Full activity log of stage transition requests, approvals, etc.

59



# Governance and Auditability

The screenshot shows three views of the Databricks interface illustrating model provenance:

- 1.a**: Registered Models > Airline\_Delay\_SparkML > Version 5. Shows details like Registered At: 2019-10-11 12:44:44, Creator: clemens.mewald@databricks.com, Stage: Production, and Source Run: Run 6151fe768a5e49d39076b07448e60d57.
- 1.b**: /Users/clemens.mewald@databricks.com/Airline Demo/... > Run 6151fe768a5e49d39076b07448e60d57. Shows Date: 2019-10-11 12:20:44, User: clemens.mewald@databricks.com, Duration: 12.8s, and Source: 02.2 Model Search.
- 1.c**: 02.2 Model Search (Python). Shows you are viewing a notebook revision from Oct 11 2019, 10:09 AM PDT. The code editor shows a training function definition:

```
def train(params):
    if params['type'] == 'spark_rf':
        regressor = RandomForestRegressor(featuresCol="features",
                                         labelCol="ArrDelay". maxBins=348. seed=42.
```

## Pain Points

- For any given production model, it is impossible to find out where it came from

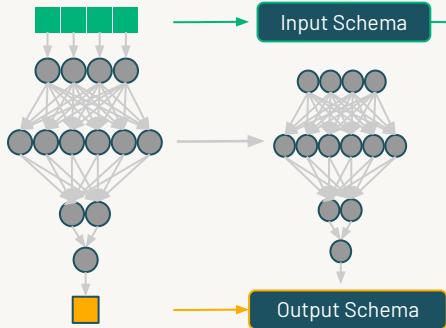
## Features

- Full provenance from Model marked production in the Registry to ...
  - Run that produced the model
  - Notebook that produced the run
  - Exact revision history of the notebook that produced the run



# Model Schema

- Schema support for all supported Models
- Schema validation
- Usability improvements



Registered Models > clemens-windfarm-signature > Comparing 2 Versions		
Run ID:	2494e060ef8547a589db131815177cbf	812fab4cba18458ea3c9a67c5ad3aec6
Model Version:	2	3
Run Name:		
Start Time:	2020-08-25 12:15:20	2020-08-25 12:23:31
Parameters		<input checked="" type="checkbox"/> Show diff only
Parameters are identical		
<input type="checkbox"/> Ignore column ordering		<input checked="" type="checkbox"/> Show diff only
Schema		
<input type="checkbox"/> Inputs		
inputs [9]	-	temperature_24: double
inputs [10]	-	wind_direction_24: double
inputs [11]	-	wind_speed_24: double
<input type="checkbox"/> Outputs		
outputs [0]	double	prediction: double
Metrics		
loss	1317002.5	994066.8
val_loss	1063741.3	768447.4

# Exercise #3



# MLflow Part 3

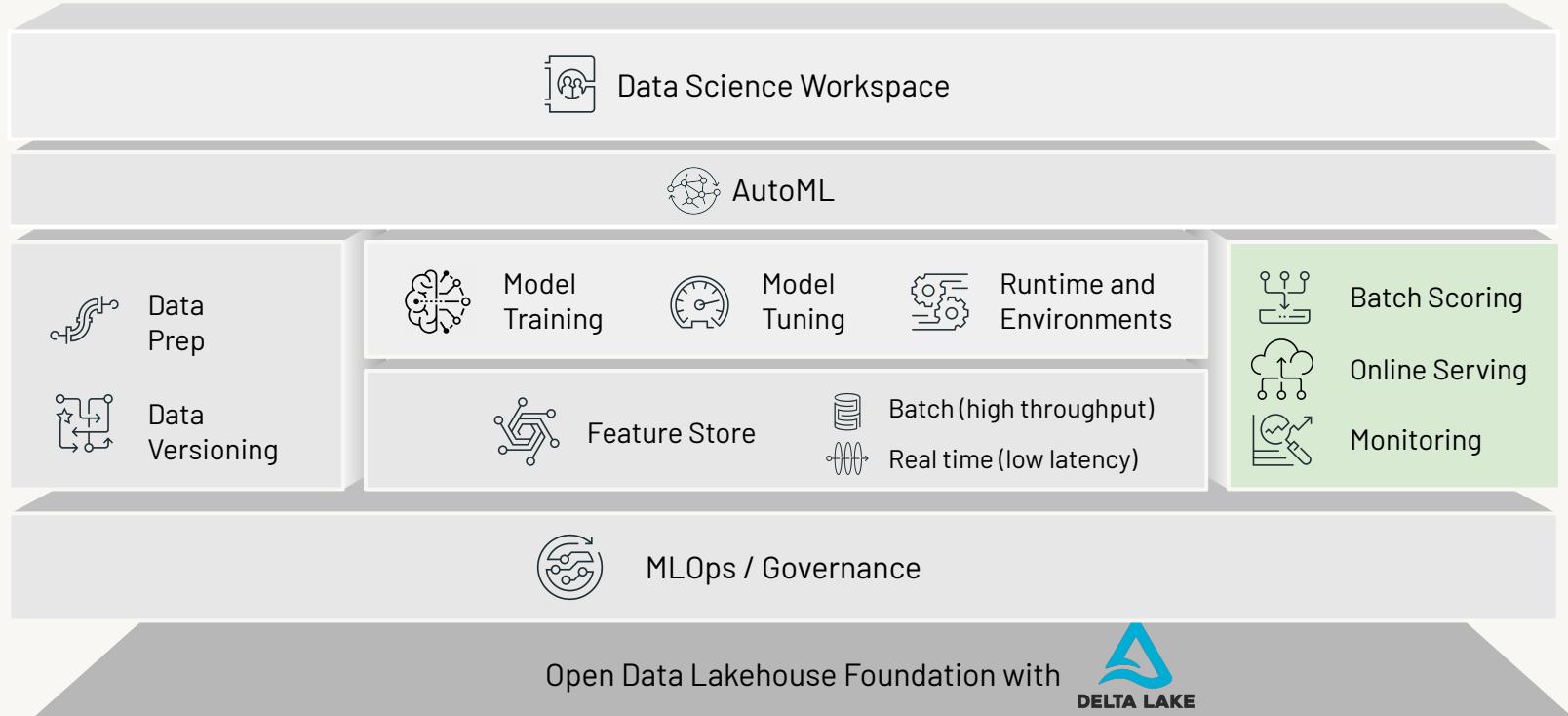
## MLflow Deployment

## Monitoring



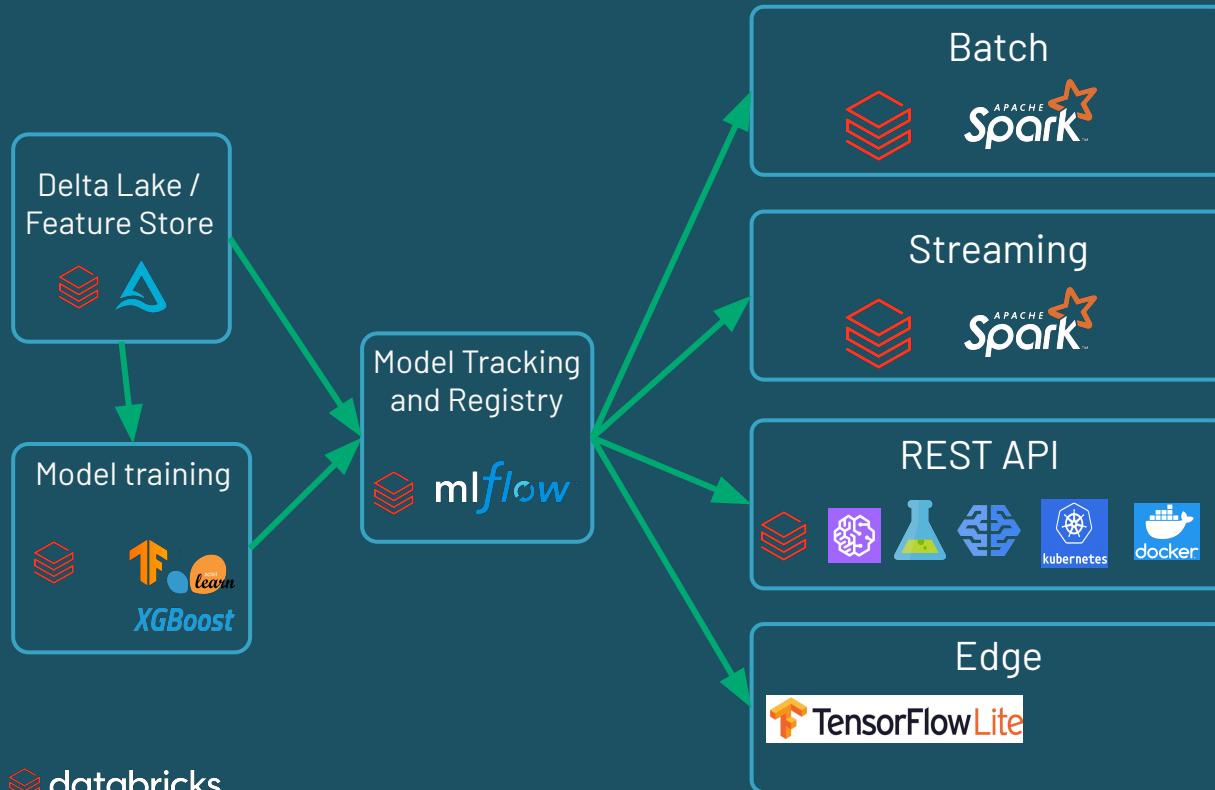
# Databricks Machine Learning

A data-native and collaborative solution for the full ML lifecycle



# Deployment models with MLflow

*Flexible Versatile Unified*



Feed ML predictions into...

- Business applications
- Data pipelines
- Web applications
- Embedded apps
- and more

Optimize deployments with...

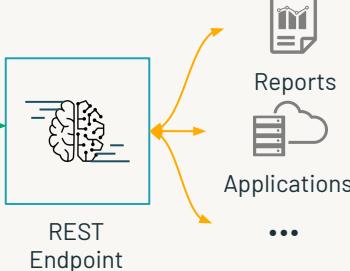
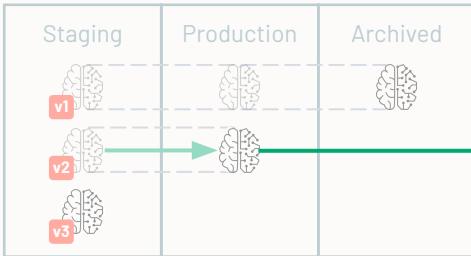
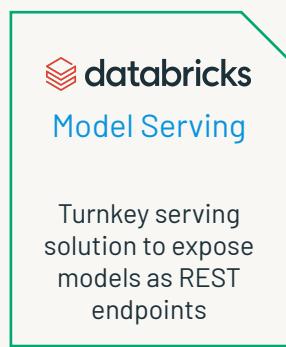
- A range of cost / latency trade-offs
- Databricks services, 3rd-party services, Feature Store model packages, and embedded applications

# How to choose deployment paradigm?

	Latency	Relative Cost
Batch (offline)	Minutes	Low
Streaming (offline)	Sec - Min	Low - Med
Real Time (online)	<1 Sec	High
Edge	varies	varies

# Model Serving

1-click deployment of MLflow models as REST endpoints



clemens-windfarm-signature ▾

Notify me about ⓘ Activity on versions I follow ⓘ

Registered Models > clemens-windfarm-signature

Status: ● Ready - Stop Cluster: mlflow-model-clemens-windfarm-signature ⓘ

Details Serving

Model Versions Model Events Cluster Settings

Model Versions

Version	Status	Model URL:
Version 2	● Ready (Production)	https://dogfood.staging.cloud.databricks.com/model/clemens-windfarm-signature/2/invocations
Version 3	● Ready (Staging)	https://dogfood.staging.cloud.databricks.com/model/clemens-windfarm-signature/Production/invocations
Version 4	● Ready	
Version 5	● Ready	
Version 6	● Ready	
Version 7	● Ready	
Version 8	● Ready	

Call The Model

Browser Curl Python

Request ⓘ

```
[{"temperature_00": 4.702021725972501, "wind_direction_00": 106.74258999999999, "wind_speed_00": 4.743291999999999, "temperature_08": 7.189482116699223, "wind_direction_08": 100.41638,}
```

Response ⓘ

```
[{"0":247.133056640625}, {"0":1836.6710205078125}, {"0":5532.92724609375}, {"0":2987.140869140625}, {"0":1318.53076171875}, {"0":3106.295166015625}, {"0":376.26806640625}, {"0":3493.88427734375}, {"0":2995.70556640625}, {"0":3201.224853515625}]
```

Send Request Show Example



# Model Deployment / Serving Options

Deploy any ML model at large scale AND low latency

Use model for inference

One click model deployment directly from the Model Registry

## Large-scale batch scoring

Configure model inference

Select either batch inference or real-time inference.

Batch inference      Real-time

This will generate a notebook in your home folder that you can edit.

\* Model version  
Production (Version 4)

\* Input table  
clemens.windfarm

Browse

\* Output table location  
/FileStore/batch-inference/ 2020-10-05-db\_automl-clemens\_windfar

Cancel      Use model for batch inference

## Low-latency online serving

Configure model inference

Select either batch inference or real-time inference.

Batch inference      Real-time

Enable realtime model serving behind a REST API interface. This will launch a single-node cluster that will host all active versions of this model. [Learn more](#).

Cancel      Enable Serving

Classic - Available in All Clouds

©2021 Databricks Inc. — All rights reserved

Serverless – Private Preview in AWS

## New Serverless Model Endpoints



Low latency REST endpoints



High availability SLAs



Serverless and autoscaling



Built-in observability and connectors



# Offline Scoring with Spark + MLflow

## When and Why?

- High Latency
- High throughput
- Low cost
- Easy to implement

## How?

- Data Ingestion
  - Spark batch
  - [Spark structured streaming](#)
- Model management
  - [MLflow Model Registry](#)
  - Support [built-in](#) and [custom flavor](#)
- Prediction
  - Distribute with [Pandas\\_udfs](#)
- Orchestration
  - Databricks [Jobs](#)

## Resources

[Deep Learning Model Inference Examples](#)



# Offline Scoring with Spark + MLflow

```
model_uri = "models:/{{model_registry_name}}/production"
```

## Score with native flavor

```
model = mlflow.sklearn.load_model(model_uri)
predictions = model.predict(data)
```

*- returns a single node sklearn model*

## Score with pyfunc flavor

```
model = mlflow.pyfunc.load_model(model_uri)
predictions = model.predict(data)
```

*- returns a mlflow pyfunc model*

## Score with Pandas UDF

```
udf = mlflow.pyfunc.spark_udf(spark,
model_uri)
predictions = data.withColumn("prediction",
udf(*data.columns))
- returns vectorized UDF and applies to a
spark dataframe in parallel
```

## Score with SQL UDF

```
udf = mlflow.pyfunc.spark_udf(spark, model_uri)
spark.udf.register("predict", udf)
%sql
select *, predict(*) as prediction from data
```



# Exercise #4



# Complete the Survey



# Thank you for attending!

