databricks

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

databricks

Register Now

Get skilled on Databricks

25% off on training,
75% off on certification

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.

- Full slate of free and paid hands-on training workshops across the spectrum of data lakehouse technologies, from Delta Lake to Apache Spark™ programming to managing ML models with MLflow.
- Get certified with new certification bundles that can help hone your skill set

DATA+AI SUMMIT 2022

HYBRID | JUNE 27-30, 2022

Register Now

Organized by
databricks

About Me

Div Saini, Solutions Architect, Databricks

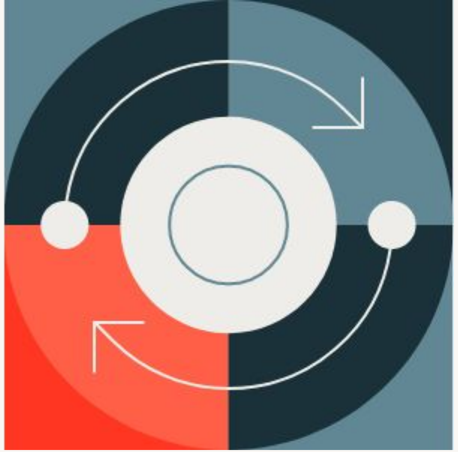
- Foodie (vegetarian)
- Love to explore new recipes (IG @milkandsaffron)
- Enjoy gardening, painting, music, traveling

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

April 27
9AM – 12PM

Databricks SQL

May 4
9AM – 12PM

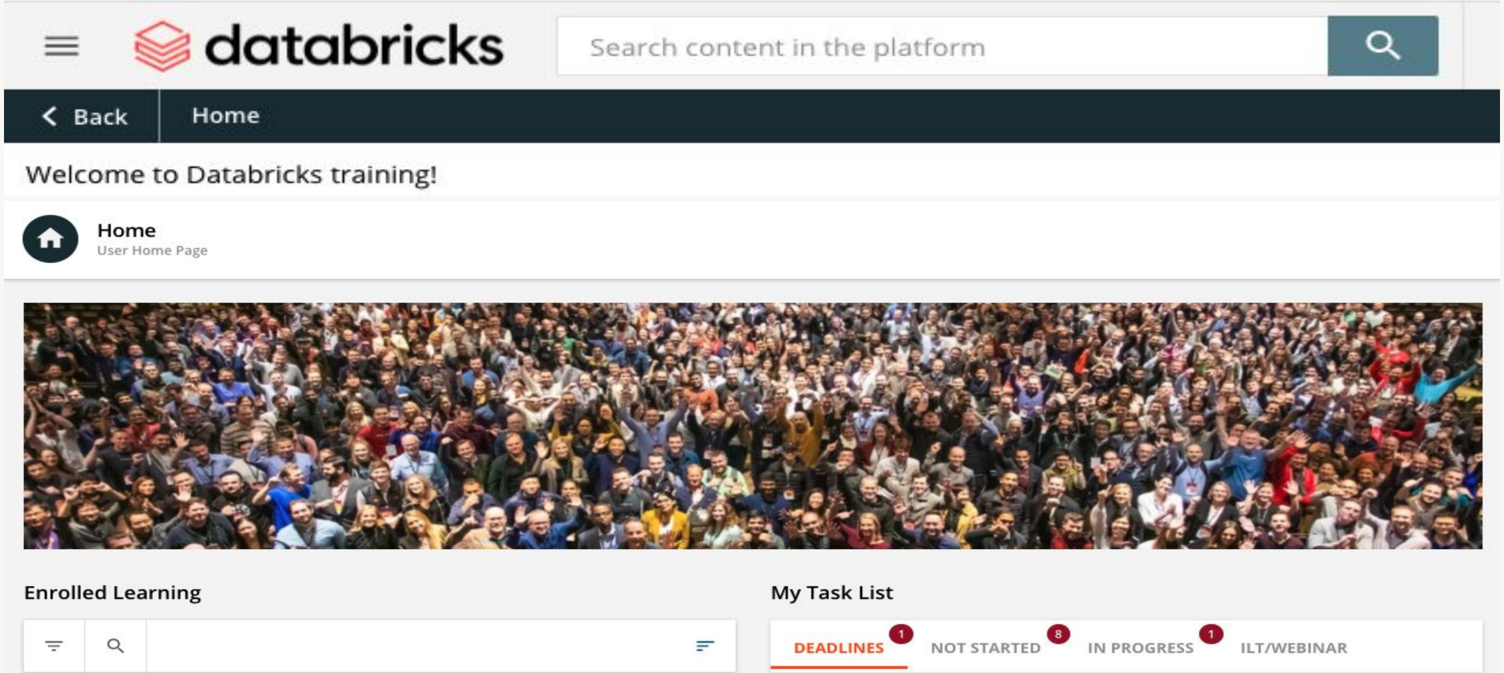
MLflow

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



Presentation Overview

What we'll cover

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

Hands on Overview

What we'll do

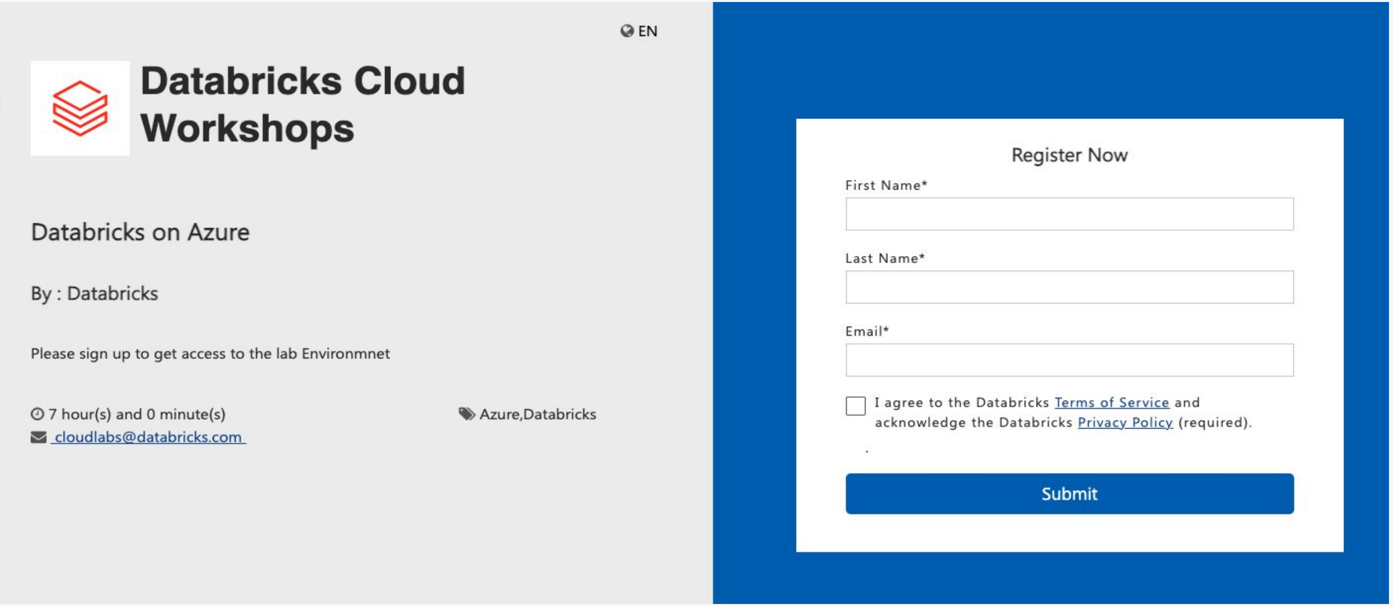
- Get hands-on with MLflow
- Track, register and serve models
- HyperParameter Tuning with Spark

Let’s get you set up

Step 1: Workspace Registration

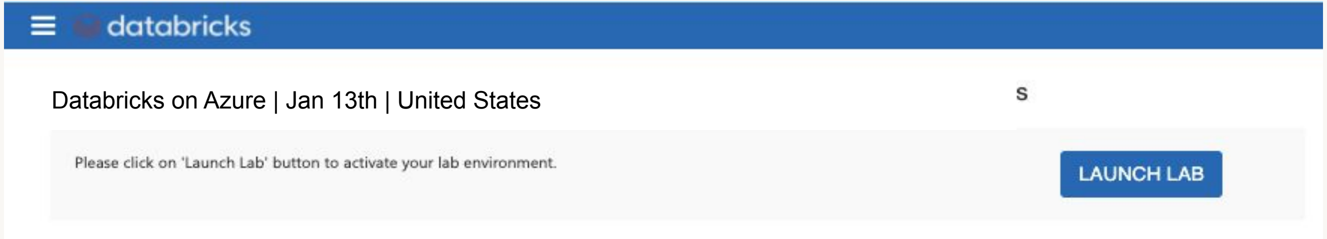
The temporary workspace you'll be using in today's workshop is administered by a third party vendor.

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



Step 2: Launching the workspace

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



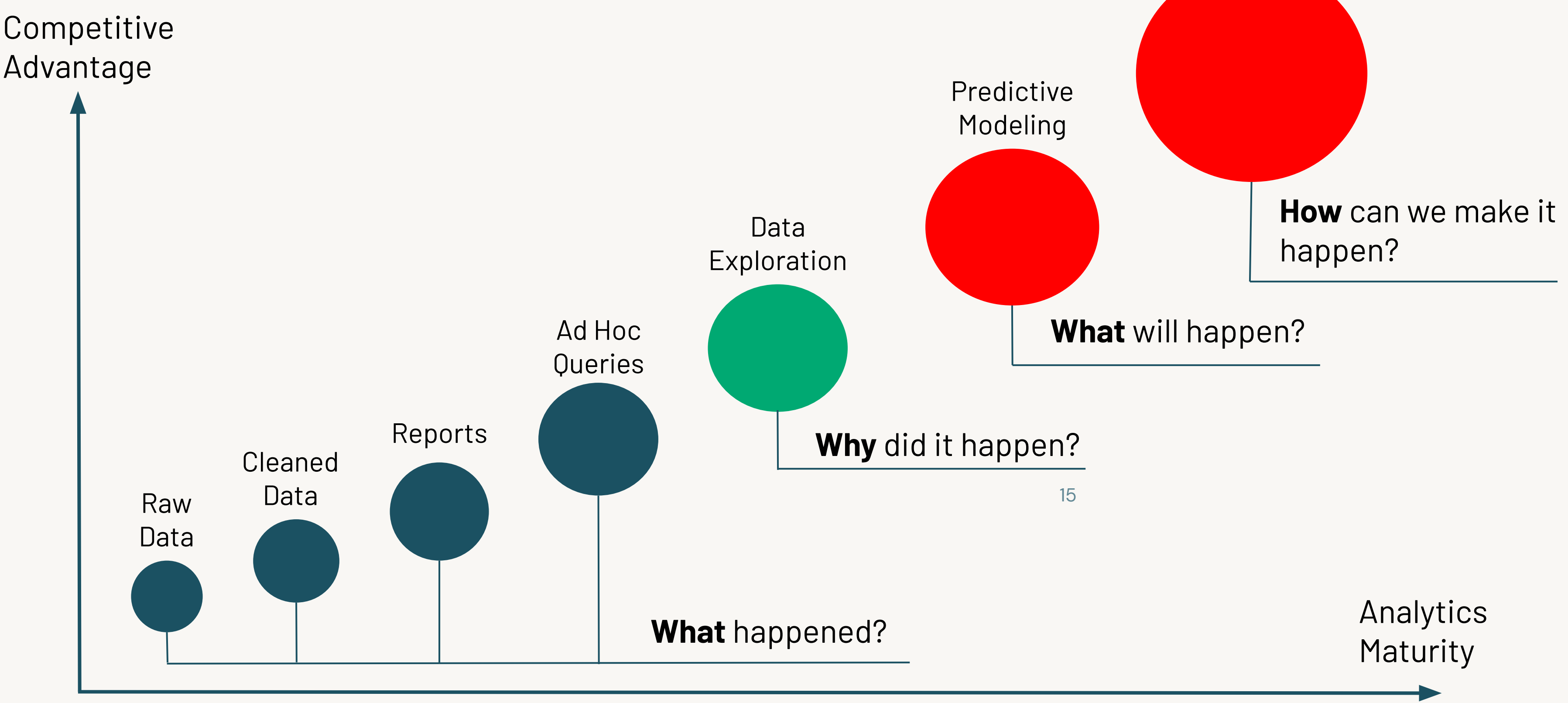
databricks

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



The future is here

...it's just not evenly distributed

83%

of CEOs say AI is a strategic priority

MITSloan Management Review

85%

of big data projects fail

Gartner

$3.9T

in business value created by AI in 2022

Gartner

87%[16]

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

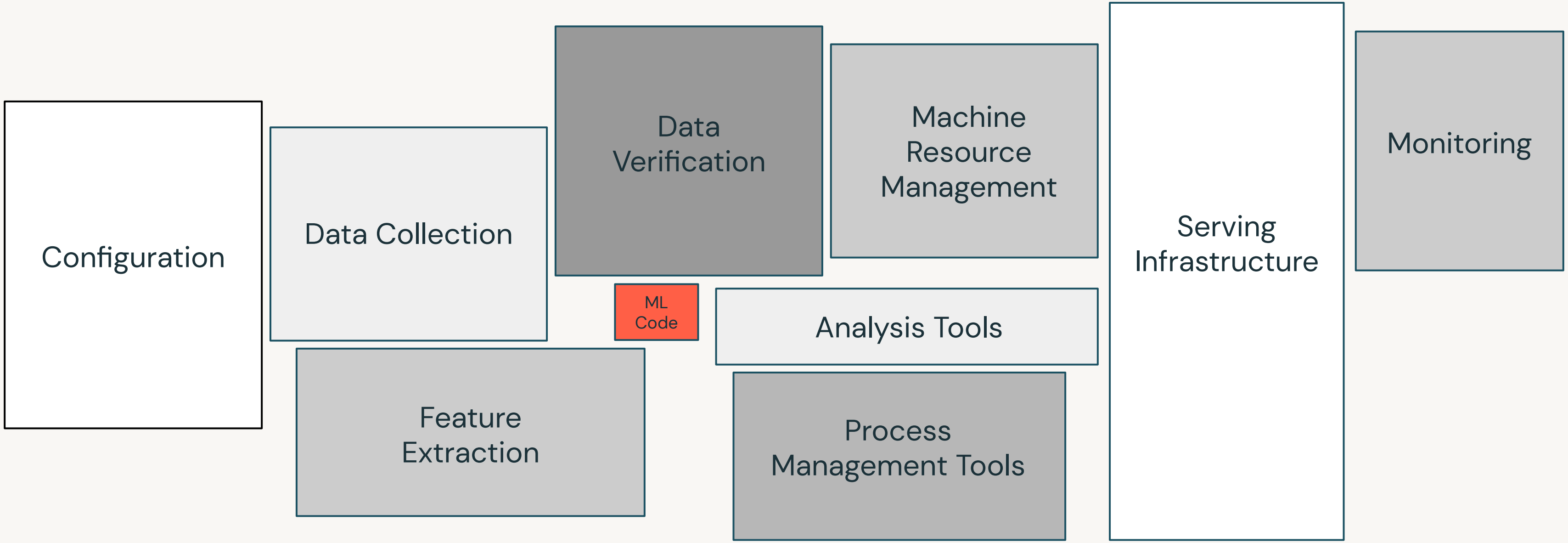


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small red box in the middle. The required surrounding infrastructure is vast and complex.

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

Apache Spark Delta Lake

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

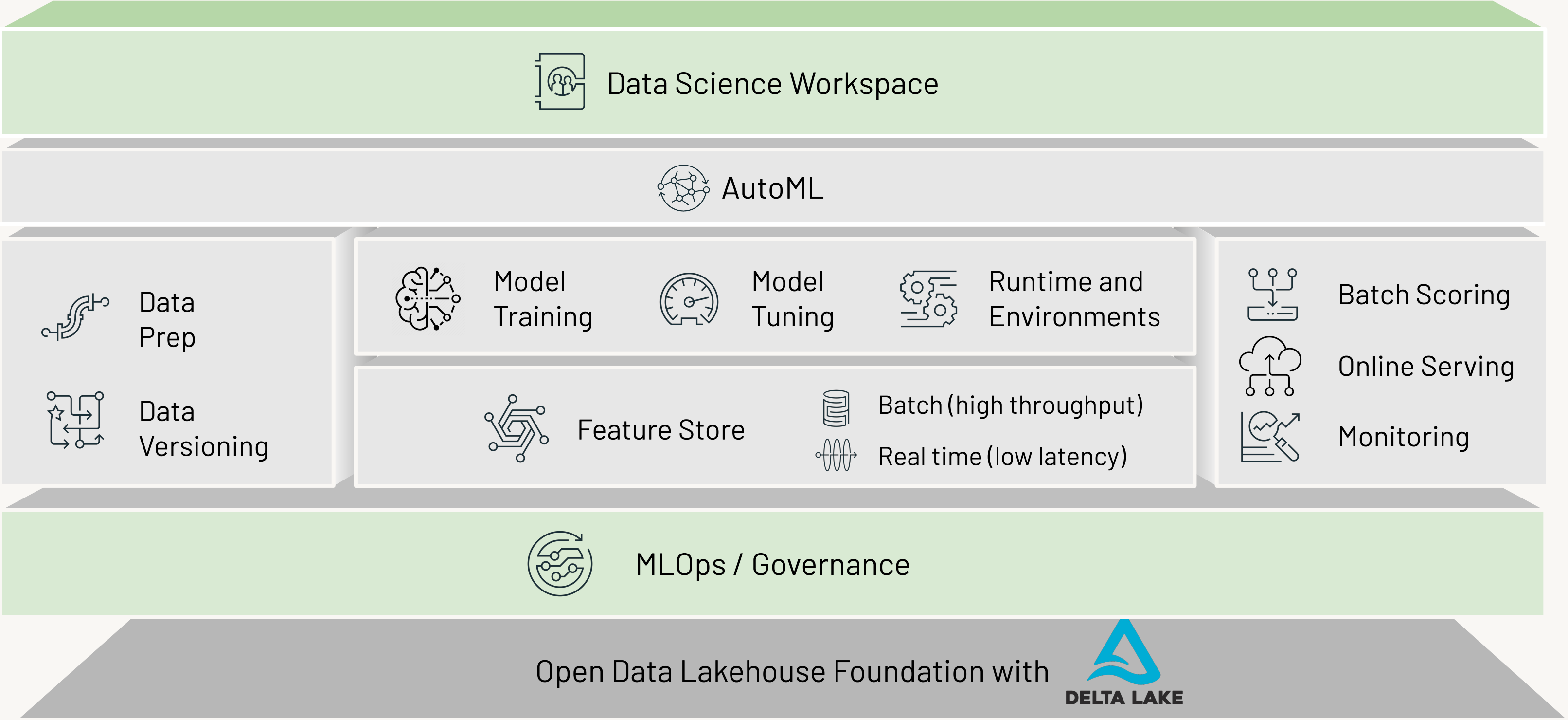
3 Standardize the full ML lifecycle

mlflow

Seamlessly and securely move models from experimentation to production

Databricks Machine Learning

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



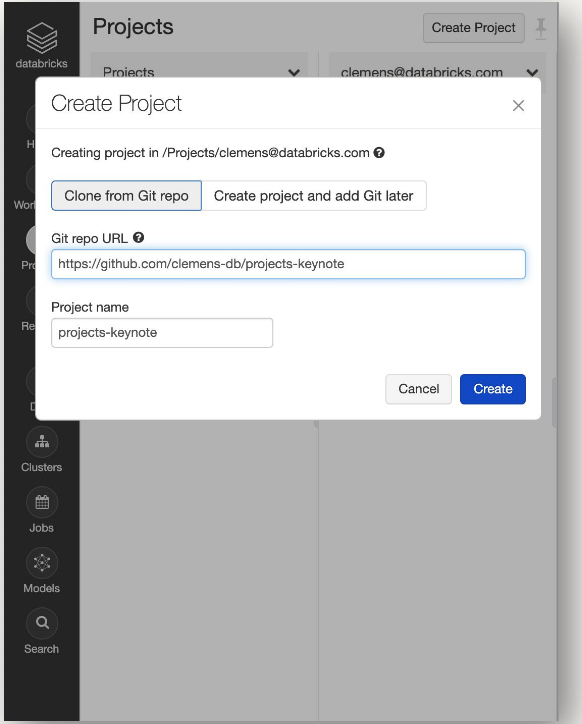
Full ML Lifecycle: MLOps for Data Teams

MLOps = DataOps + DevOps + ModelOps

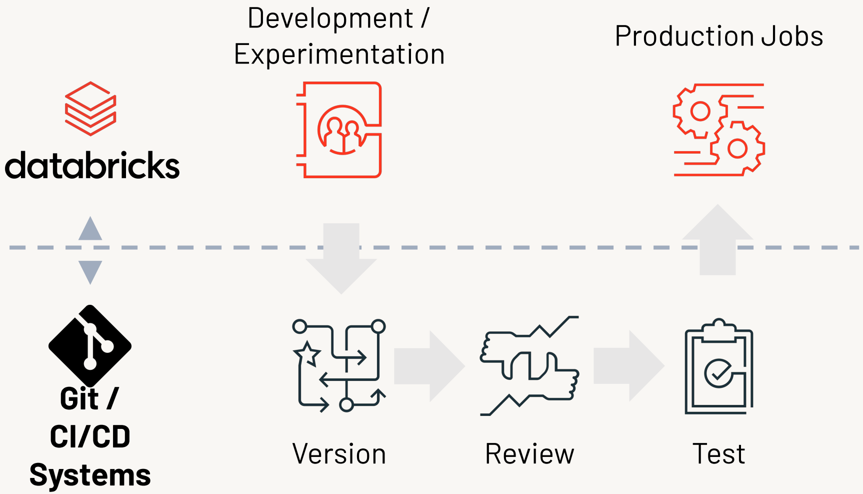
DataOps	DevOps	ModelOps
DELTA LAKE	Repos	mlflow
Data Versioning with Time Travel	Code Versioning with Git Integration	Model Lifecycle Management with Model Registry

MLOps / Governance

(Git-based) Projects



CI/CD Integration



Supported Git Providers

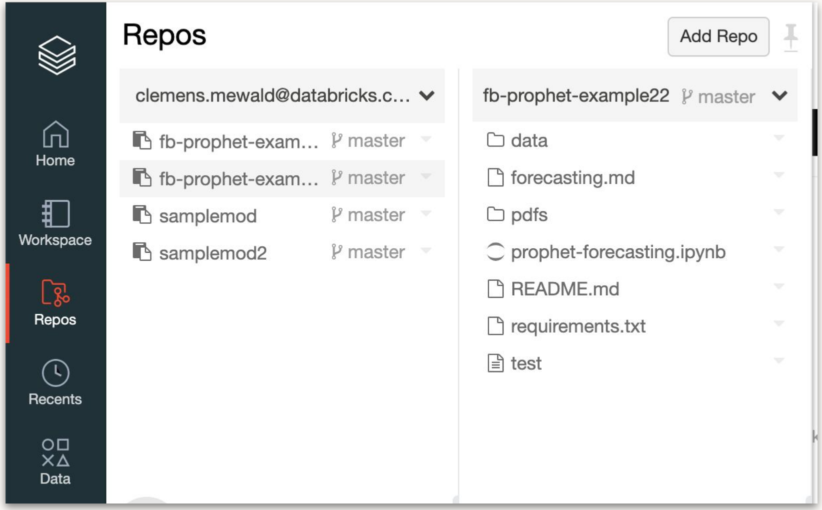
Azure DevOps GitHub GitLab Bitbucket

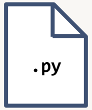
In Public Preview

Repos

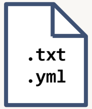
File support within Repos

Seamlessly bring any workload to Databricks





- Portability of code
- Library files

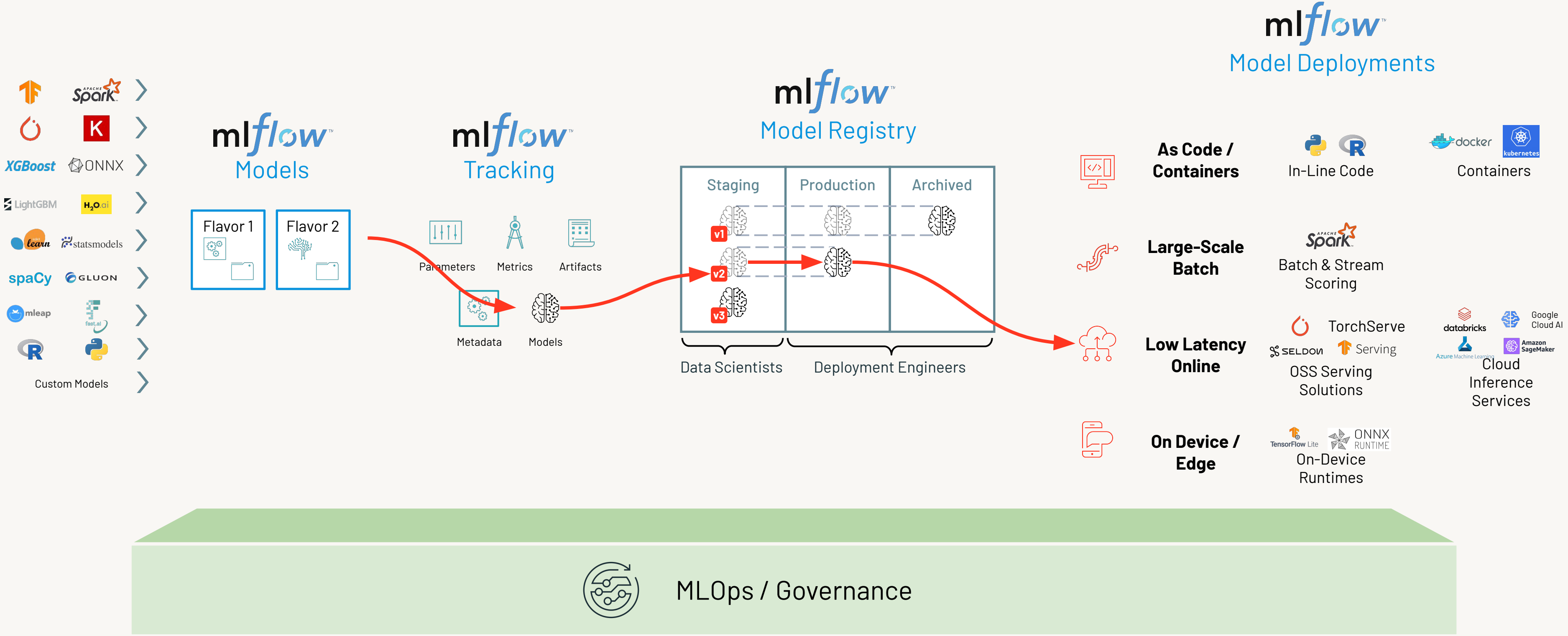


- Environment specification

- Small data ease of use
- Relative imports

- Automate workflows with Repos API

Full ML Lifecycle: MLOps for Data Teams

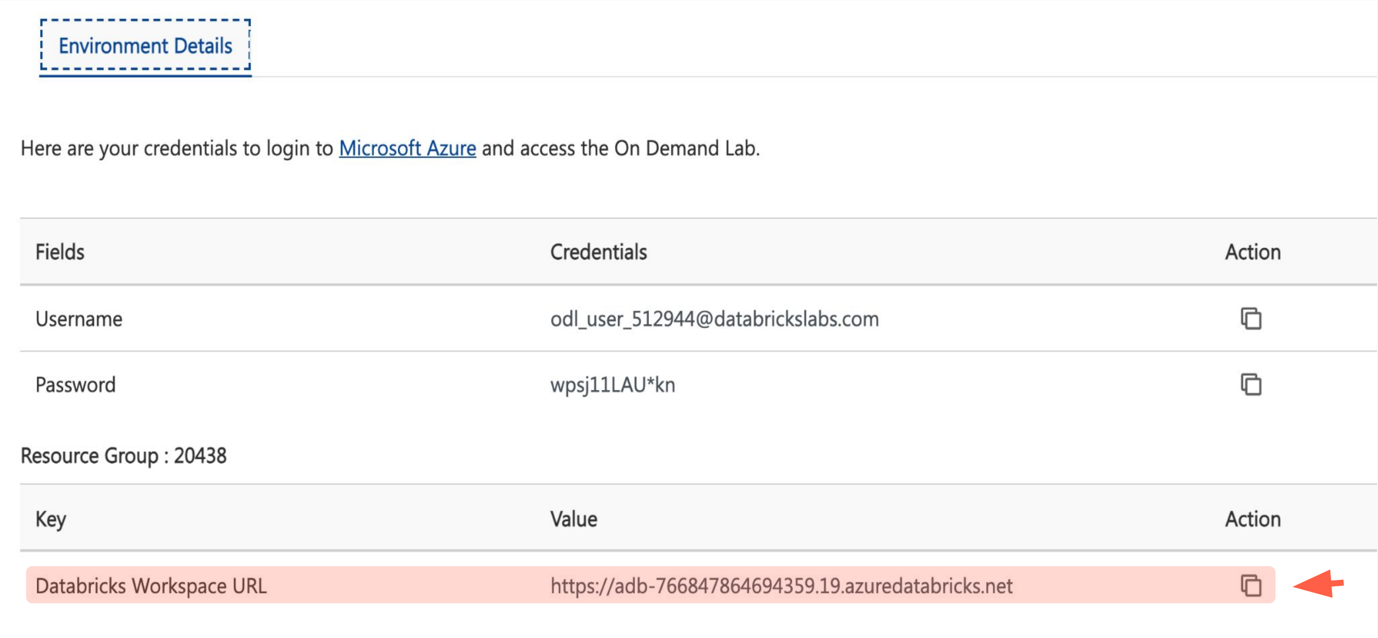


Exercise #1: Workspace Access

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

Here are your credentials to login to Microsoft Azure and access the On Demand Lab.

Fields	Credentials	Action
Username	odl_user_512944@databrickslabs.com	
Password	wpsj11LAU*kn	

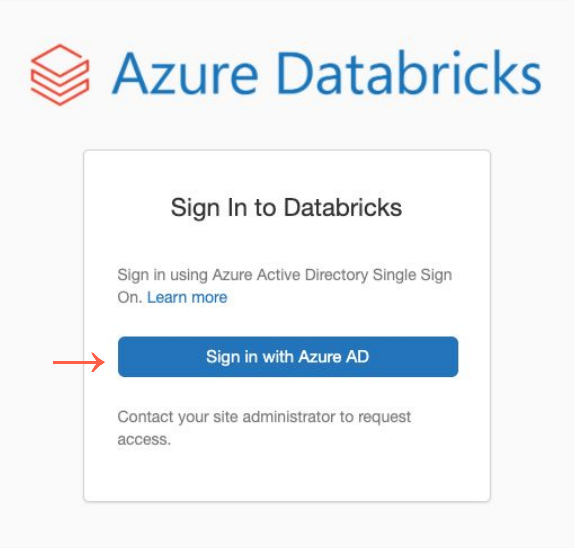
Resource Group : 20438

Key	Value	Action
Databricks Workspace URL	https://adb-766847864694359.19.azuredatabricks.net	

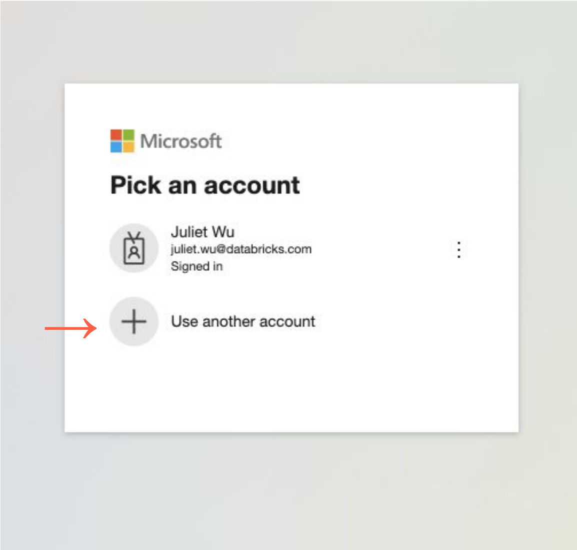
Step 4-5: Signing in on Azure

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

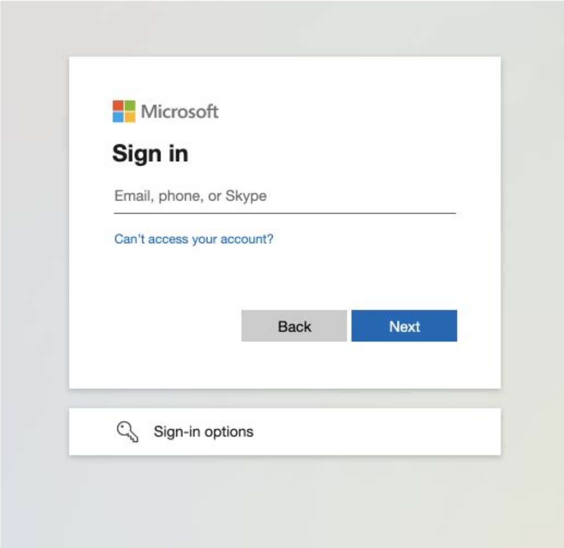
Select **Sign in with Azure AD**



Select **Use another account**



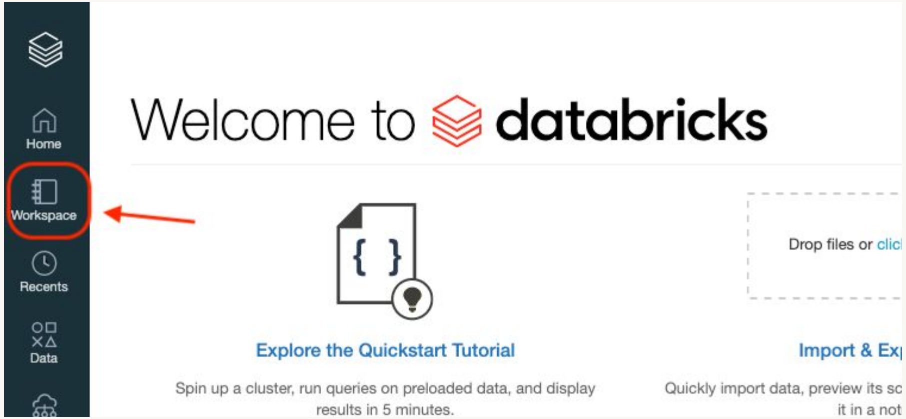
Copy-paste **provided credentials**



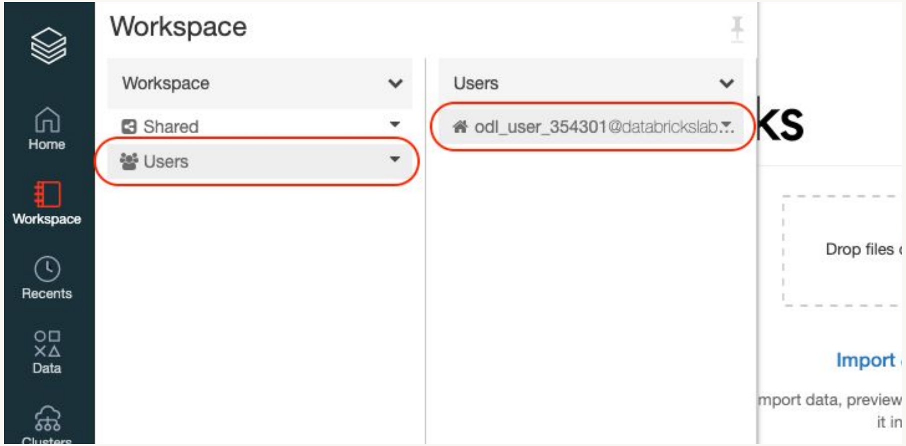
Arrive in your Databricks Workspace

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



Break – return in 5 mins

MLFlow Part 1

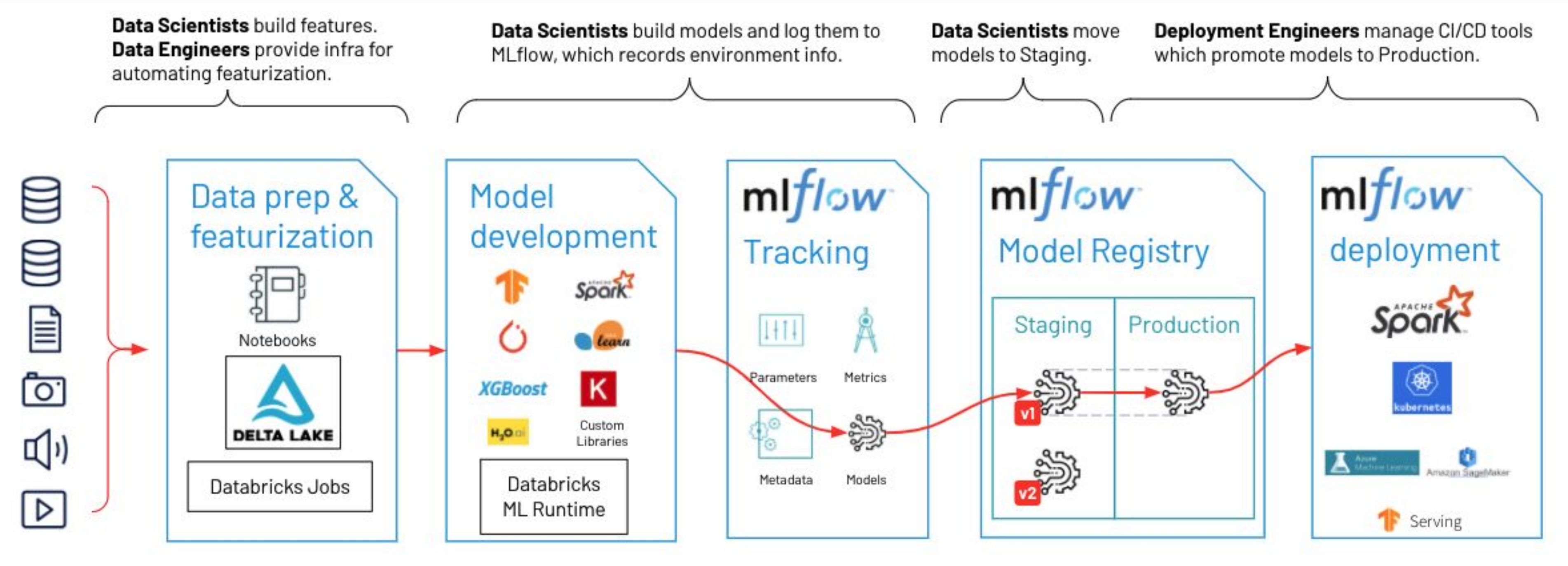
Feature Store

Tracking

Hyperparameter Tuning

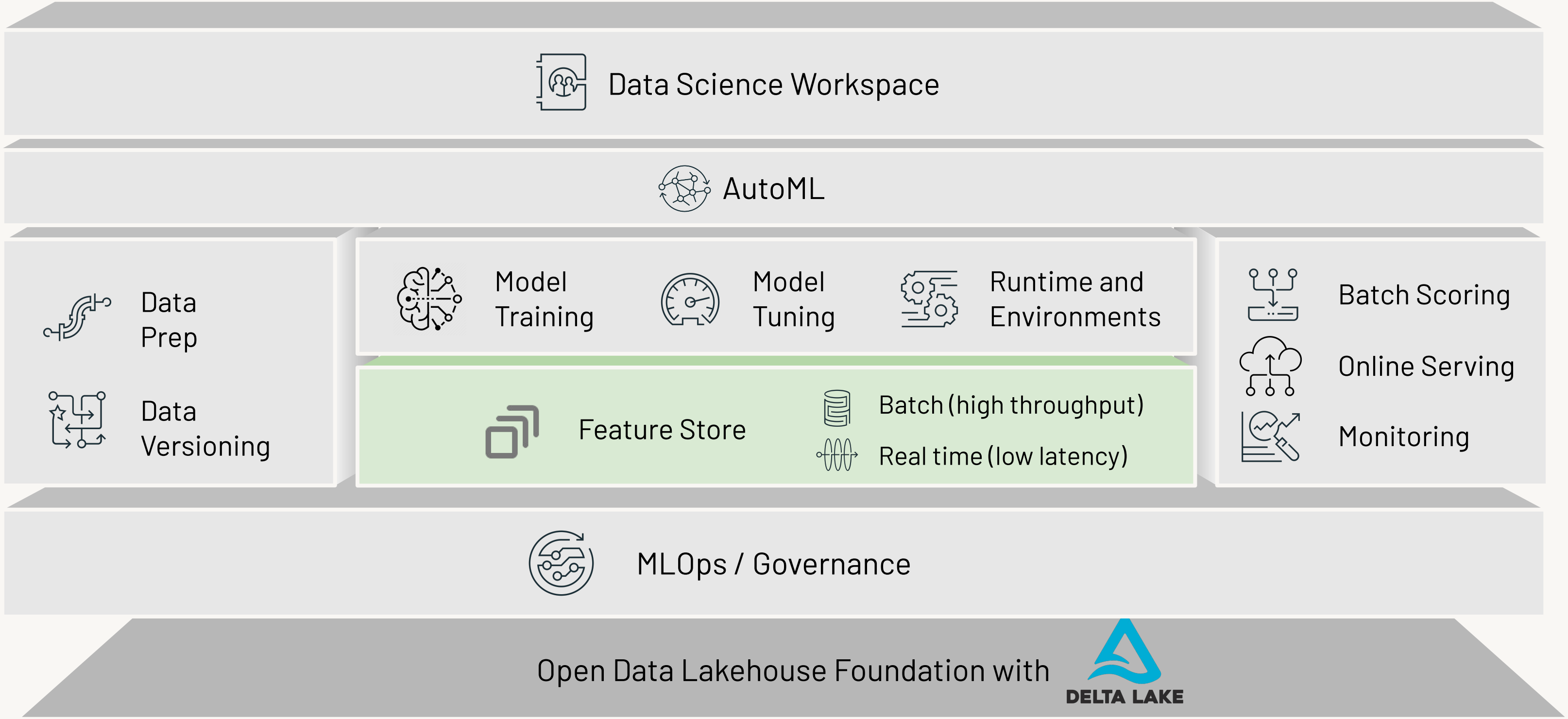
AutoML

The Full ML Lifecycle



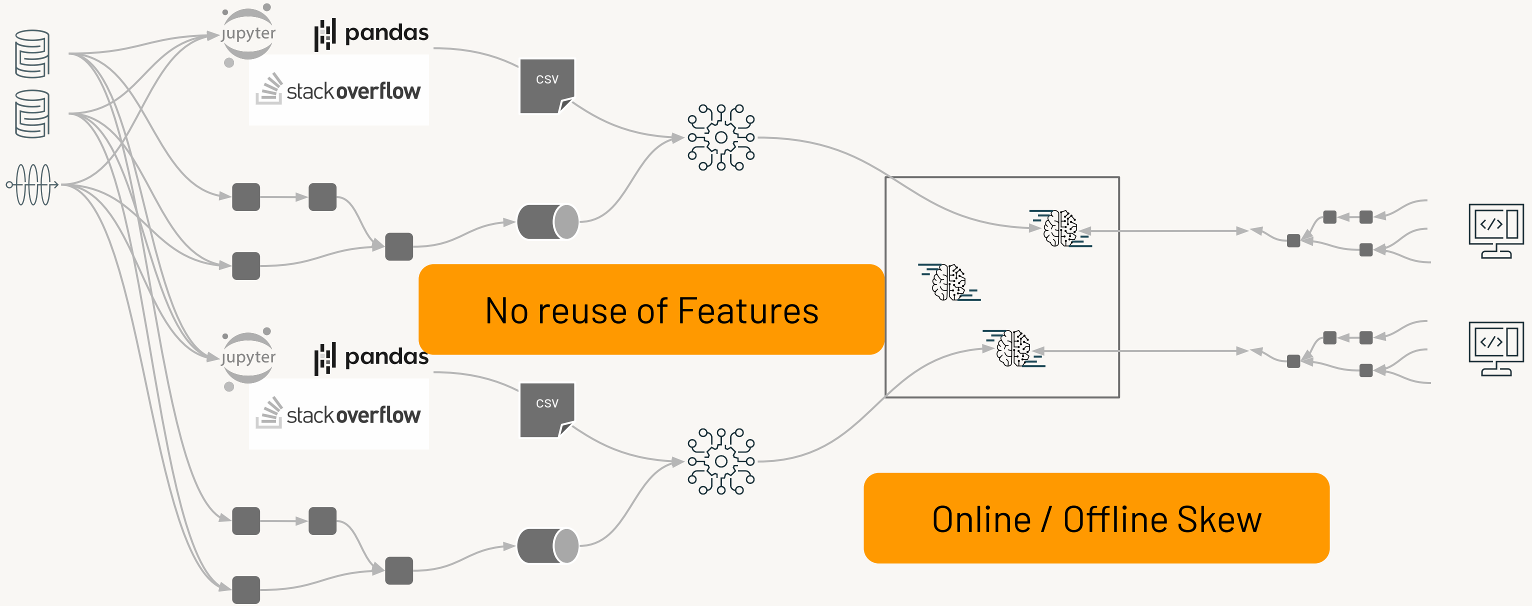
Feature Store

The first Feature Store codesigned with a Data and MLOps Platform



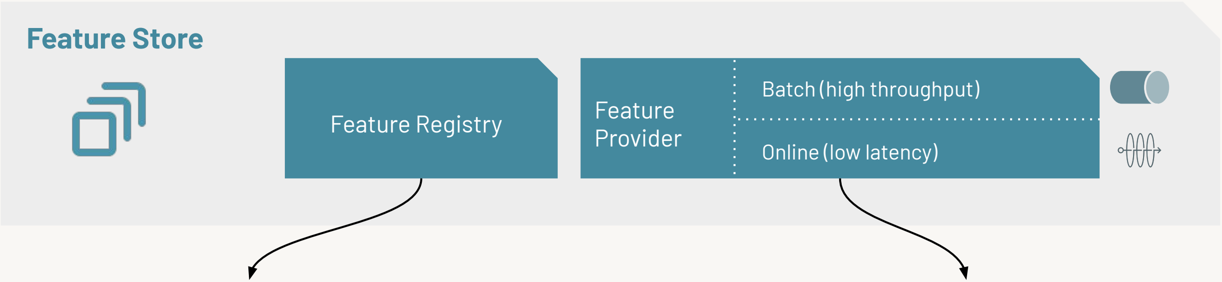
A day (or 6 months) in the life of an ML model

Raw Data	Featurization Joins, Aggregates, Transforms, etc.	Training	Serving	Client



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

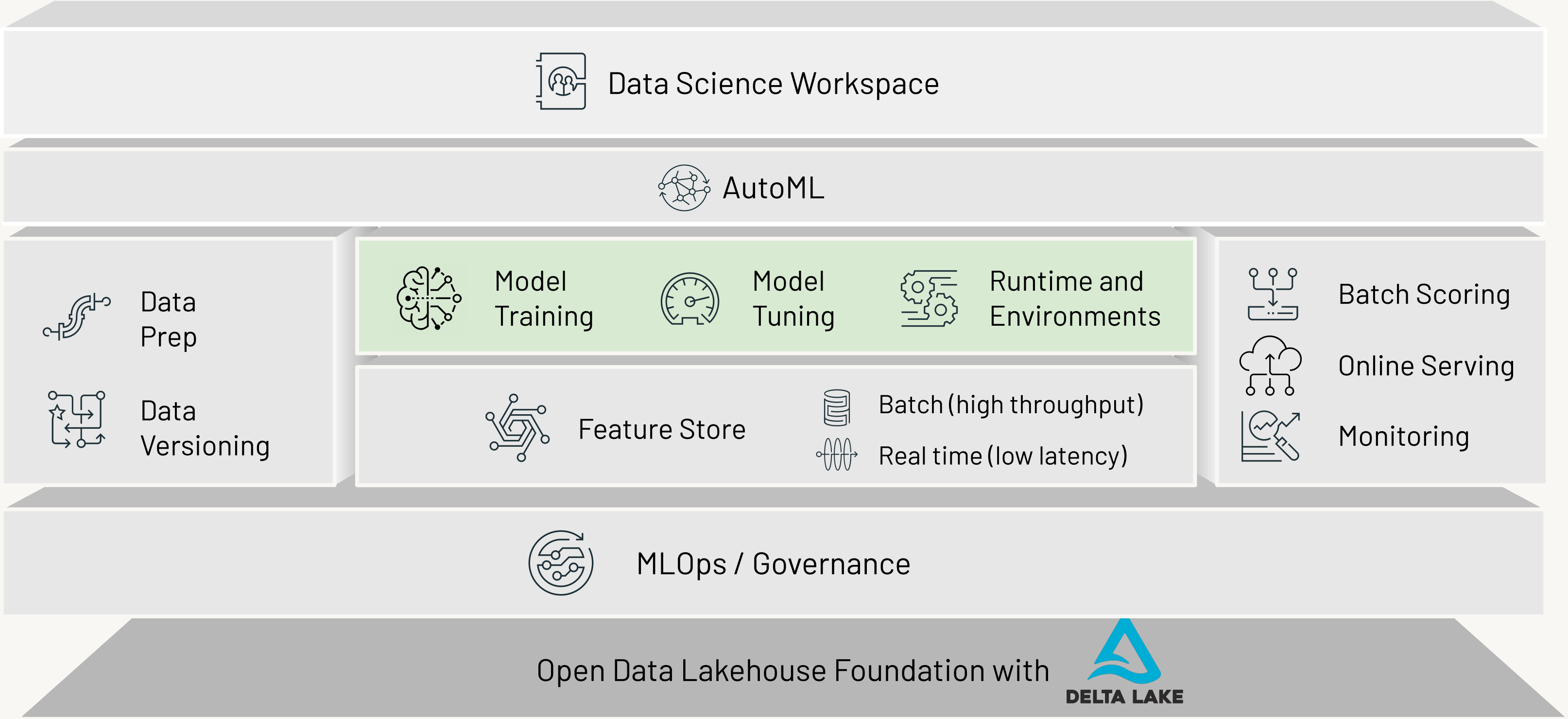
- Open format
- Built-in data versioning and governance
- Native access through PySpark, SQL, etc.

Co-designed with mlflow

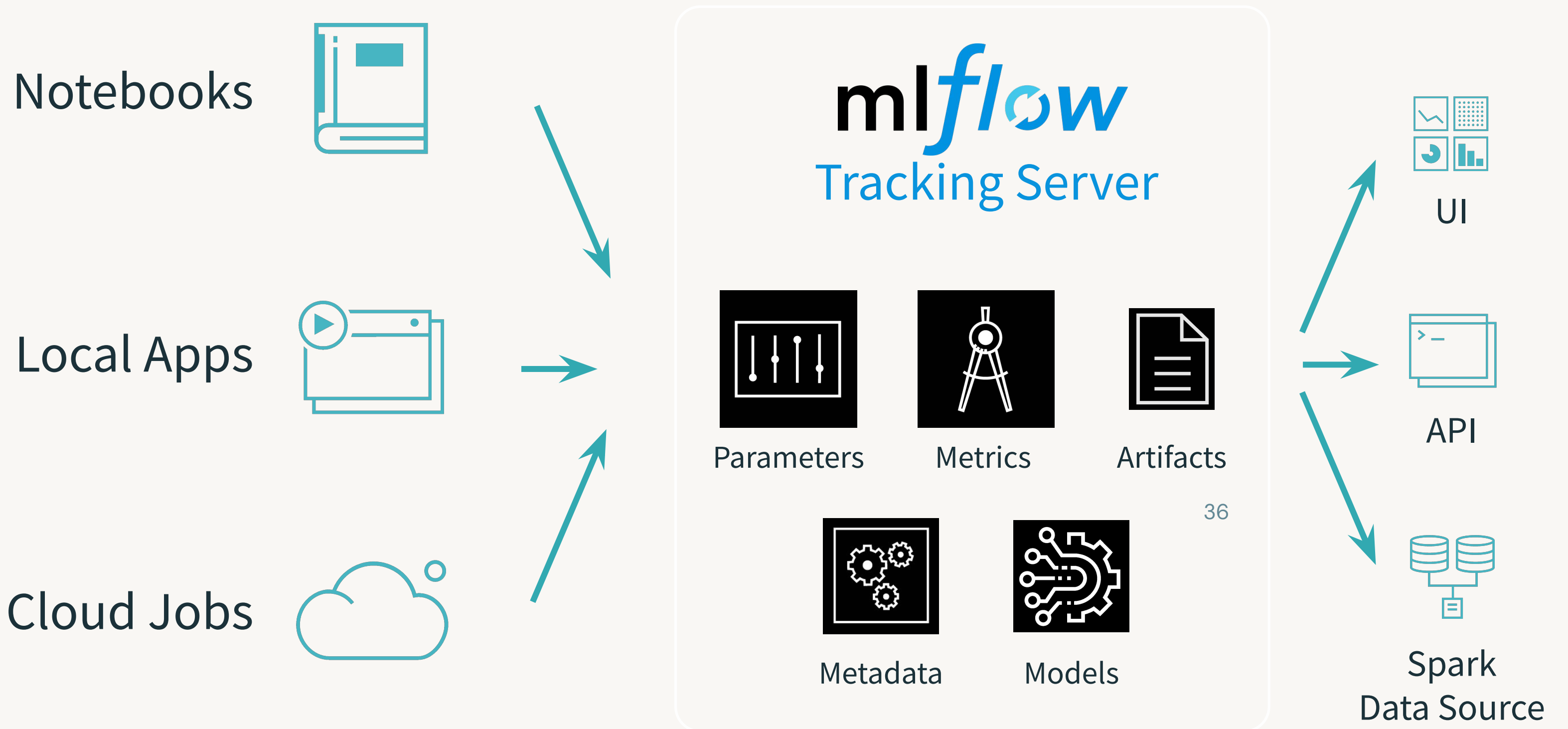
- Open model format that supports all ML frameworks
- Feature version and lookup logic hermetically logged with Model

Databricks Machine Learning

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



mlflow Tracking



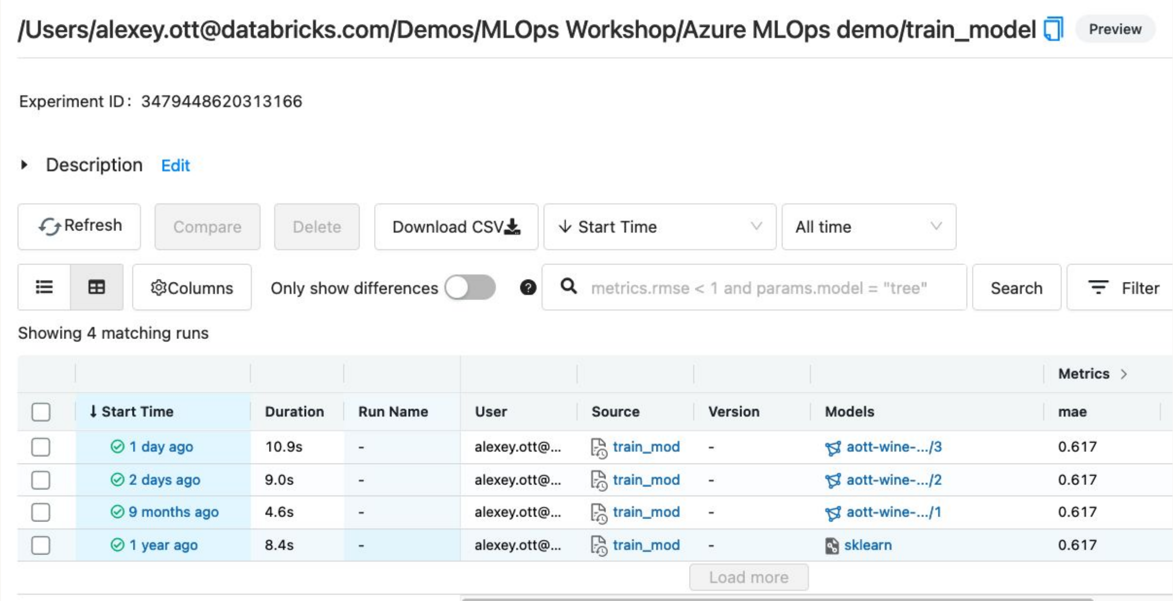
Key Concepts in Tracking

Parameters: key-value inputs to your code

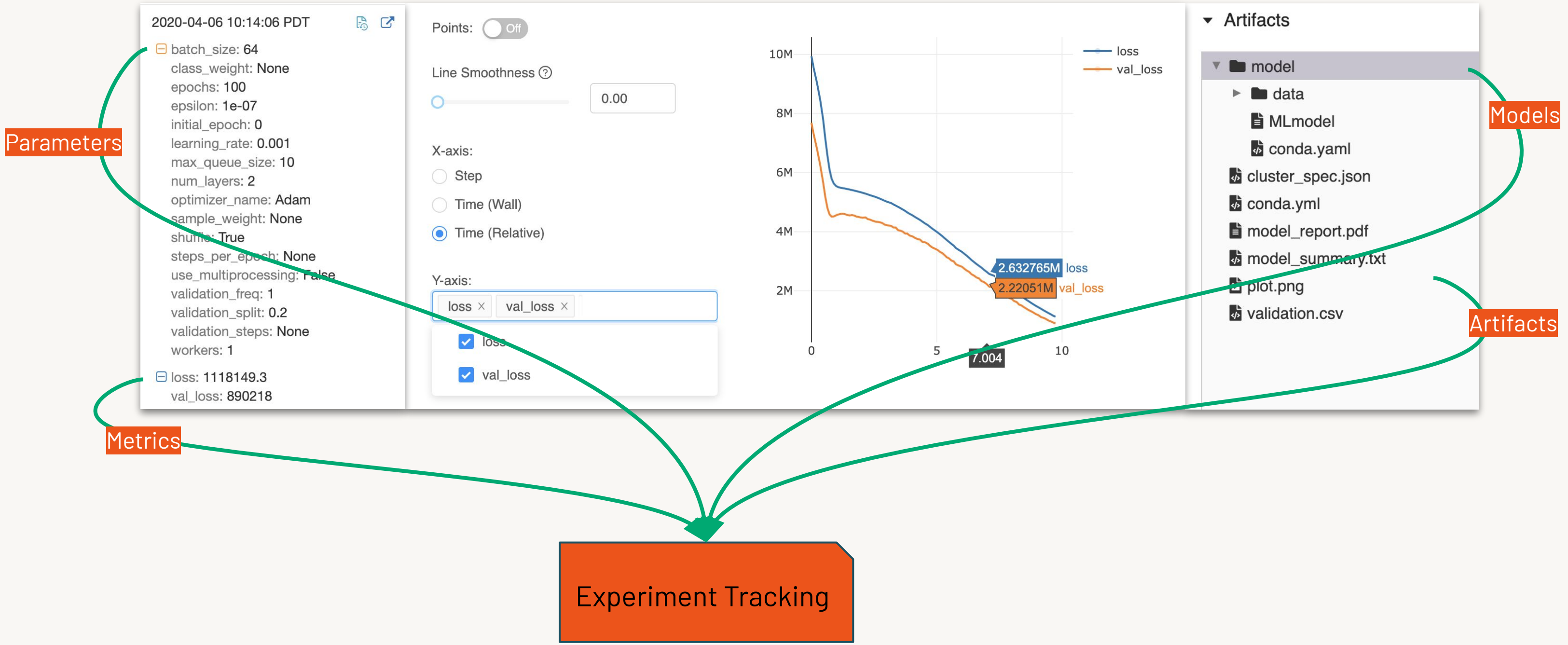
Metrics: numeric values (can update over time)

Artifacts: arbitrary files, including models

Source: what code ran?



mlflow Tracking



Powered by mlflow

“Manual” Logging

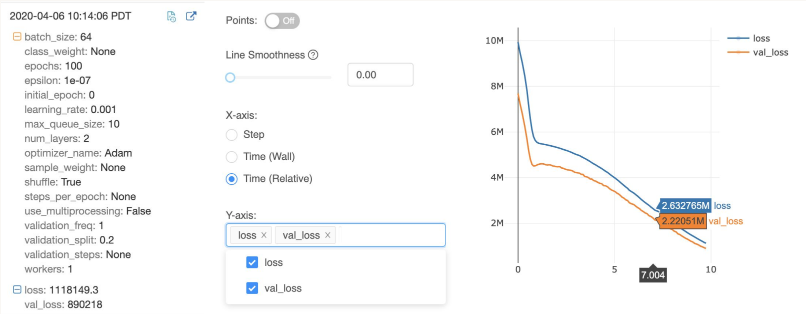
```
with mlflow.start_run() as run:
   lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio)
   lr.fit(train_x, train_y)
   predictions = lr.predict(test_x)
   (rmse, mae, r2) = eval_metrics(test_y, predictions)
   mlflow.log_param("alpha", alpha)
   mlflow.log_metric("rmse", rmse)
   mlflow.sklearn.log_model("model",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

   ...
```

Auto-Logging

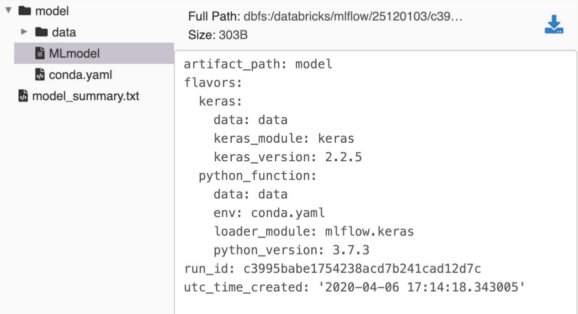
Auto-logging for ML Frameworks: A single line of code logs parameters, metrics, and artifacts.

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

Parameters and (a time series of) metrics

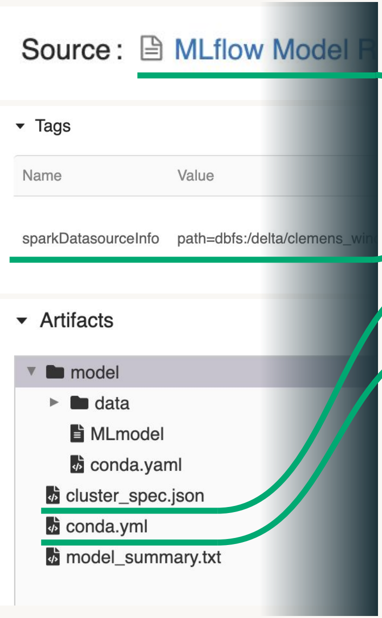


Artifacts (including model)



Auto-Logging for Reproducibility

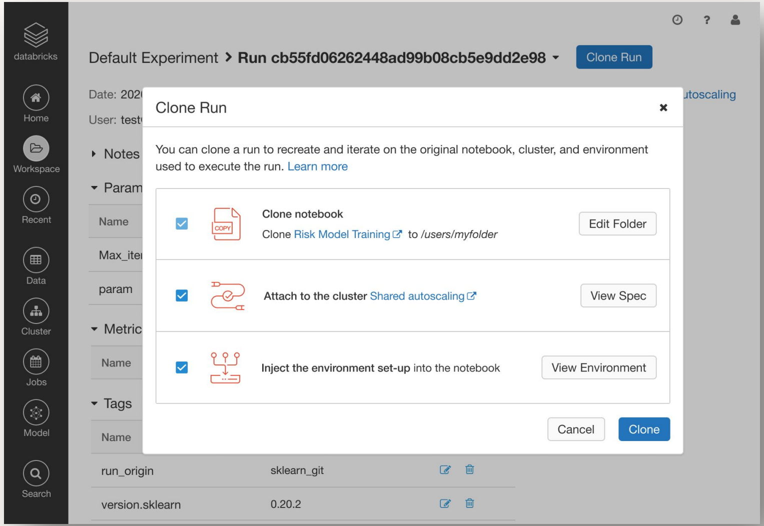
Auto-Logging of Cluster Specification and Environment Dependencies



Reproducibility Checklist:

- ✓ Code versioning
- ✓ Data versioning
- ✓ Cluster configuration
- ✓ Environment specification

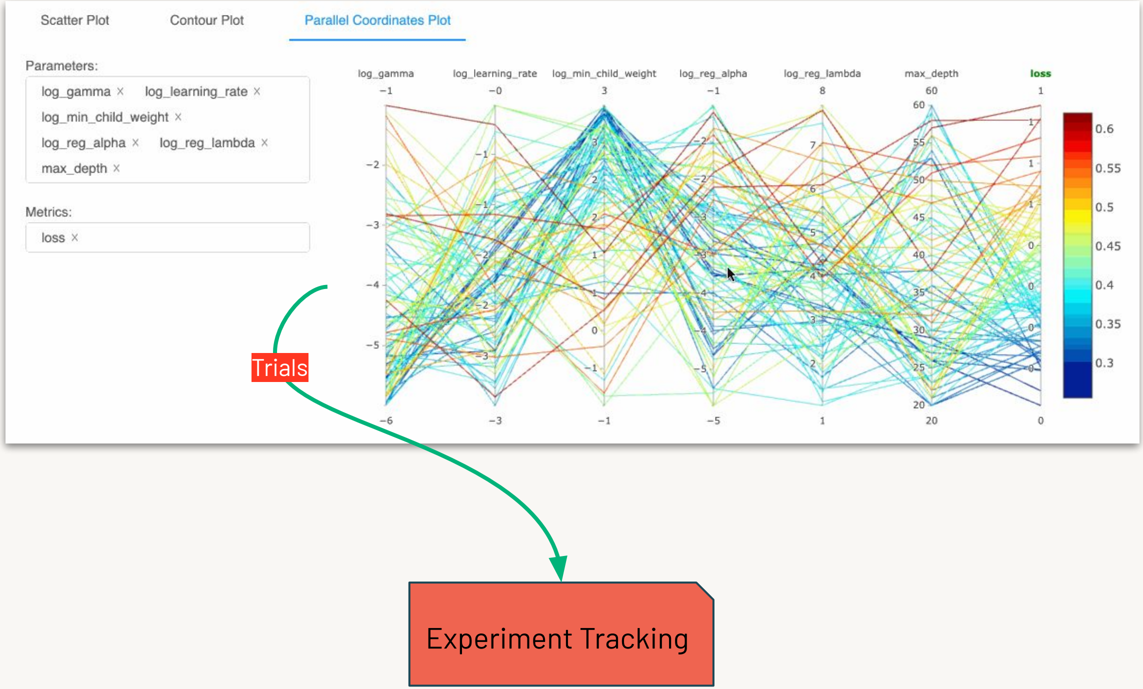
Reproduce Run Feature:



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

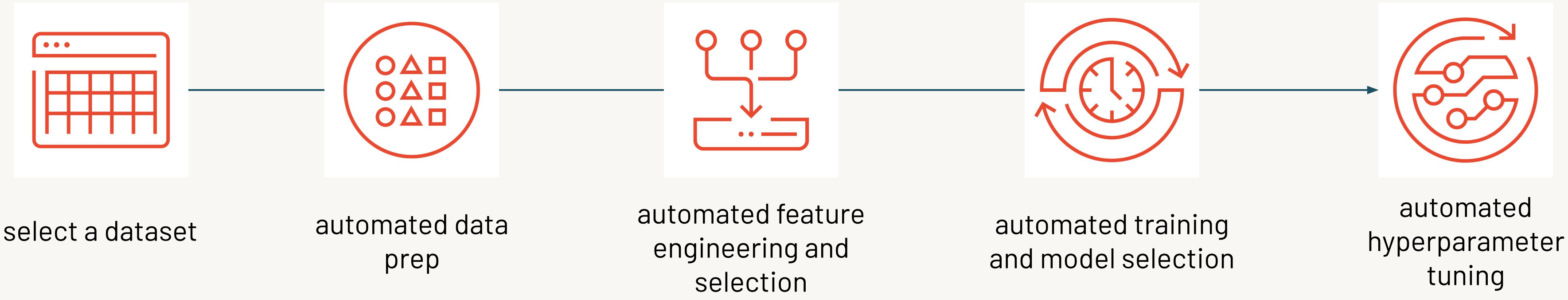
Automated capture of Hyperparameter Search



Powered by mlflow

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.

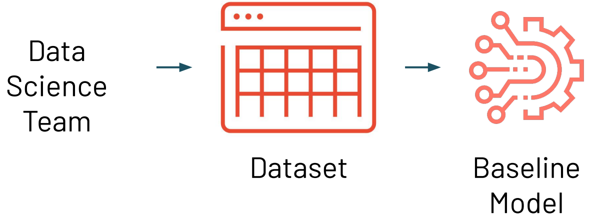


AutoML solves two key pain points for data scientists

Quickly Verify the Predictive Power of a Dataset

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

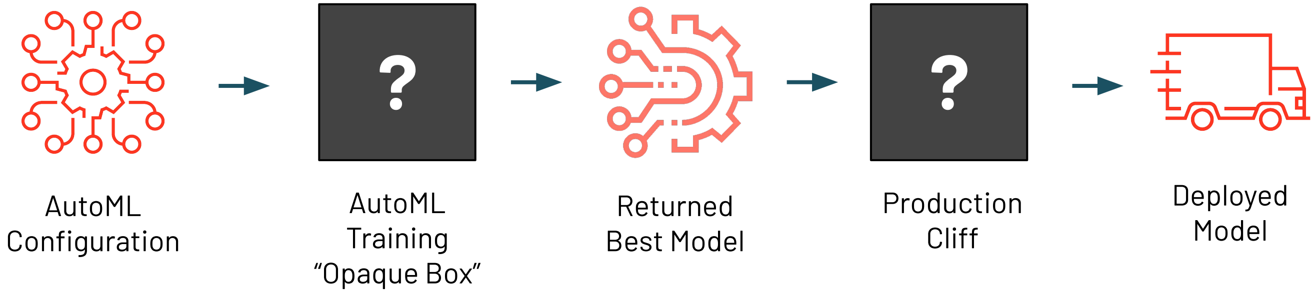
Get a Baseline Model to Guide Project Direction



"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

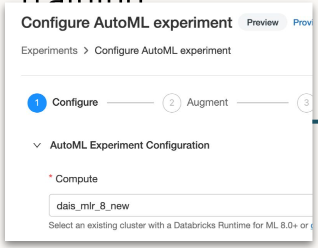


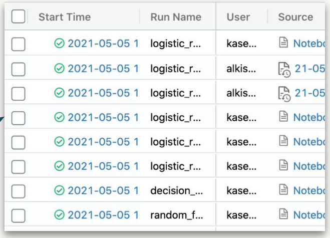
Problem	**Result / Pain Points**
1. A "production cliff" exists where data scientists need to modify the returned "best" model using their domain expertise before deployment 2. Data scientists need to be able to explain how they trained a model for regulatory purposes (e.g., FDA, GDPR, etc.) and most AutoML solutions have "opaque box" models	• The "best" model returned is often not good enough to deploy • Data scientists must spend time and energy reverse engineering these "opaque-box" returned models so that they can modify them and/or explain them

Databricks AutoML

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

UI and API to start AutoML training

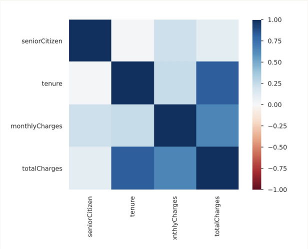




	Start Time	Run Name	User	Source
	2021-05-05 1	logistic_r...	kase...	Notebo
	2021-05-05 1	logistic_r...	alkis...	21-05
	2021-05-05 1	logistic_r...	alkis...	21-05
	2021-05-05 1	logistic_r...	kase...	Notebo
	2021-05-05 1	logistic_r...	kase...	Notebo
	2021-05-05 1	logistic_r...	kase...	Notebo
	2021-05-05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

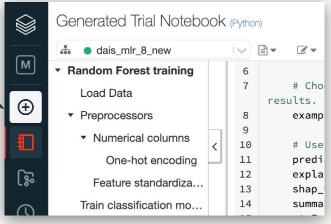
MLflow experiment
Auto-created MLflow Experiment to track models and metrics

Easily deploy to Model Registry



Data exploration notebook
Generated notebook with feature summary statistics and distributions

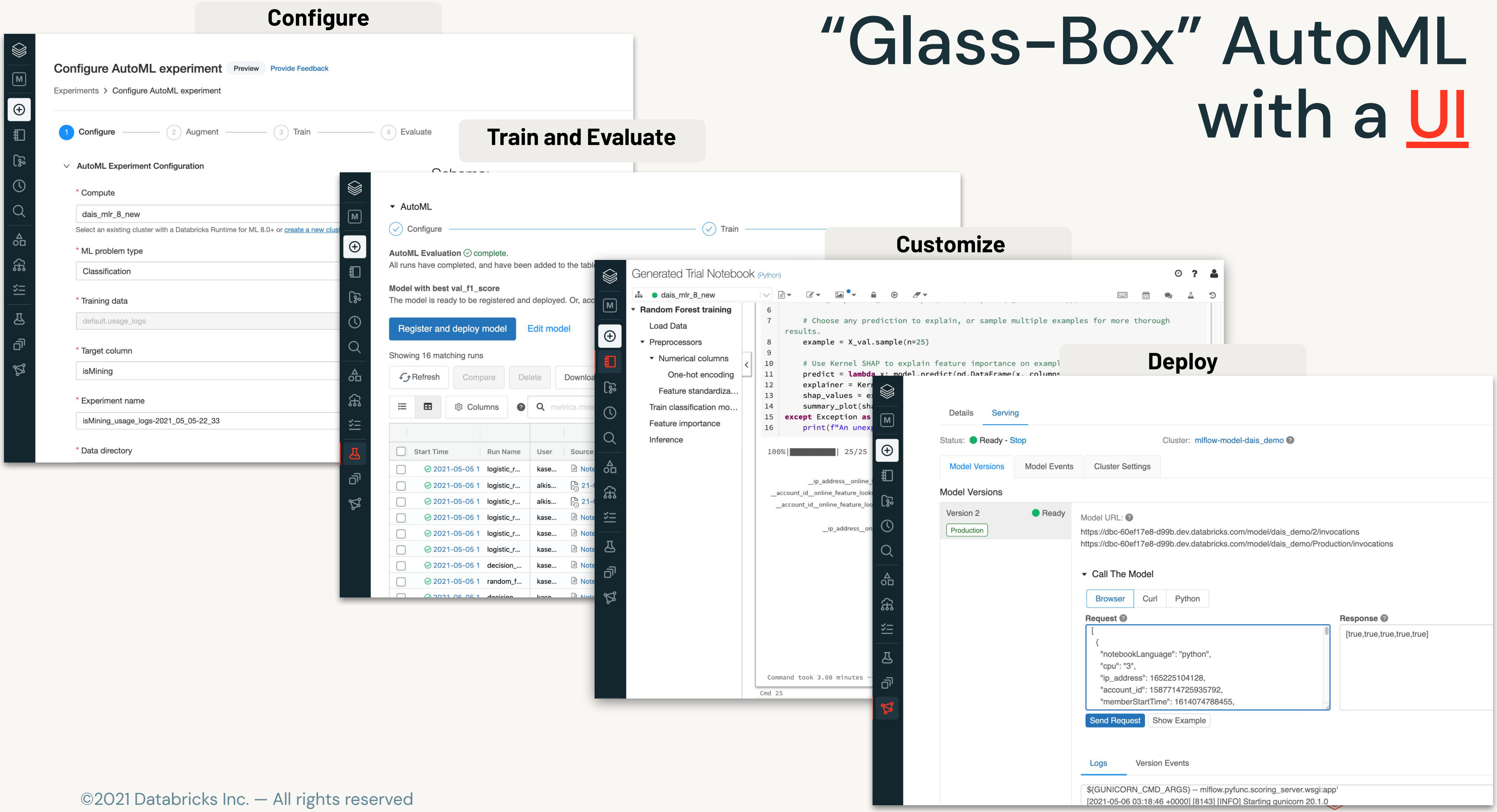
Understand and debug data quality and preprocessing



Reproducible trial notebooks
Generated notebooks with source code for every model

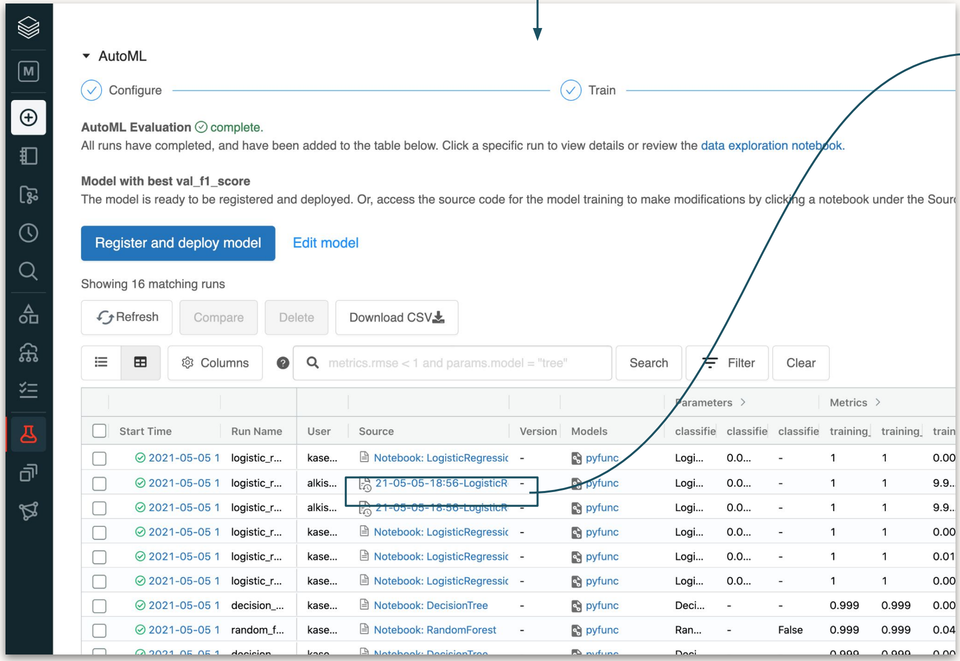
Iterate further on models from AutoML, adding your expertise

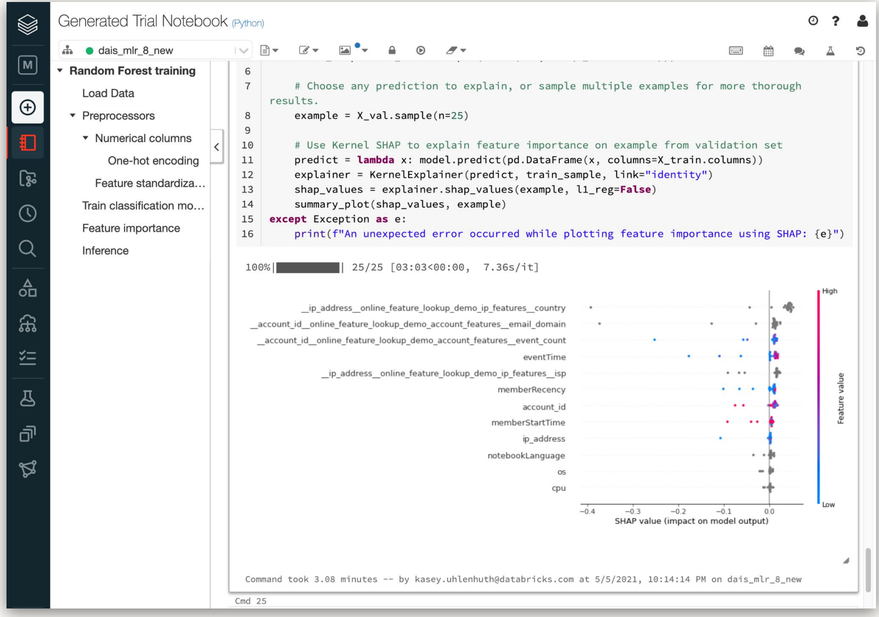
"Glass-Box" AutoML with a UI



“Glass-Box” AutoML with an API

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



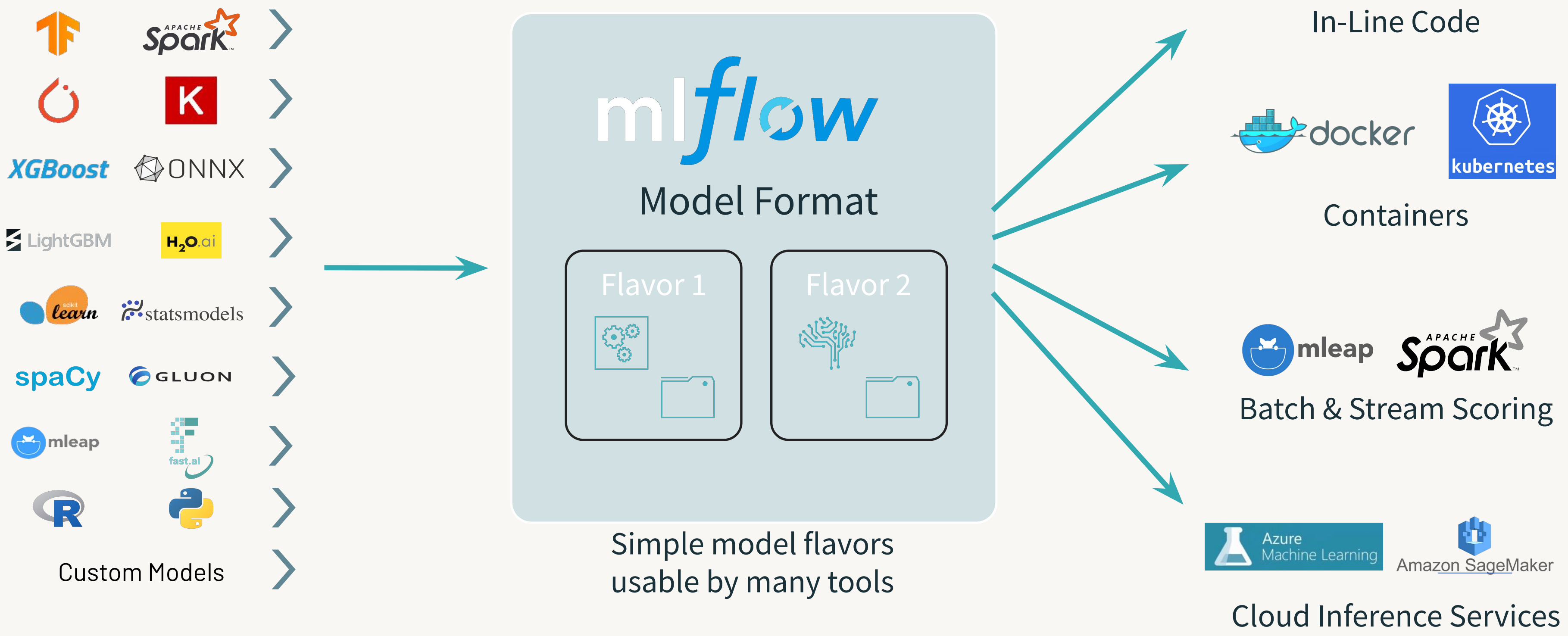


Exercise #2

Break – return in 5 mins

MLflow Part 2
MLflow Models
MLflow Registry

mlflow Models



Example MLflow Model

```
my_model/
├── MLmodel
│
│
│
│
│
└── estimator/
    ├── saved_model.pb
    └── variables/
        ...
```

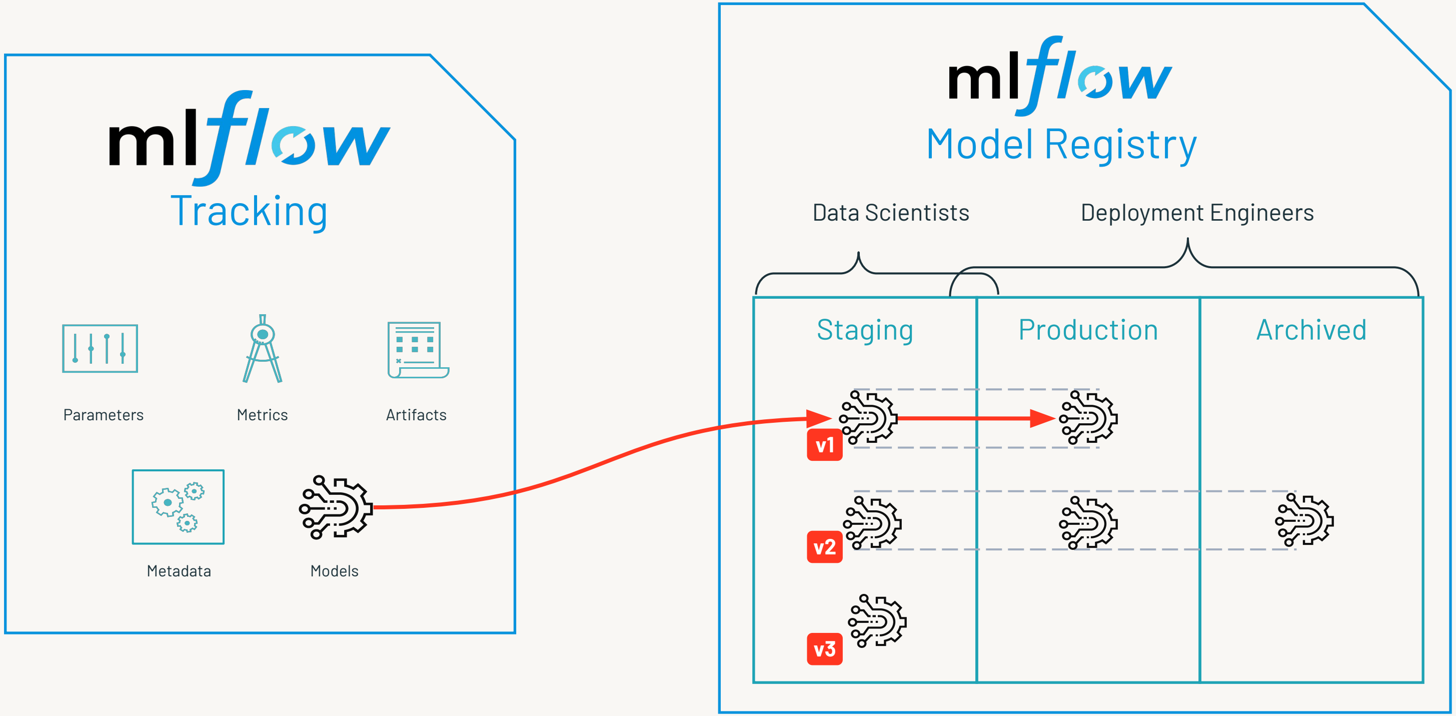
```
run_id: 769915006efd4c4bbd662461
time_created: 2018-06-28T12:34
flavors:
  tensorflow:
    saved_model_dir: estimator
    signature_def_key: predict
  python_function:
    loader_module: mlflow.tensorflow
```

Usable by tools that understand TensorFlow model format

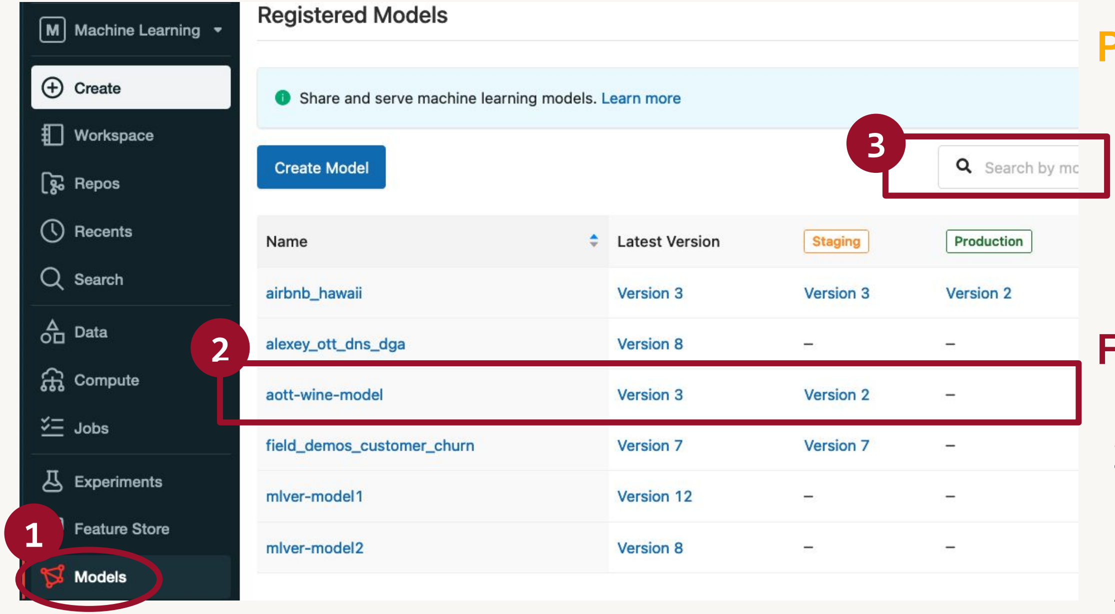
Usable by any tool that can run Python (Docker, Spark, etc!)

mlflow Model Registry

VISION: Centralized and collaborative model lifecycle management



One Collaborative Hub



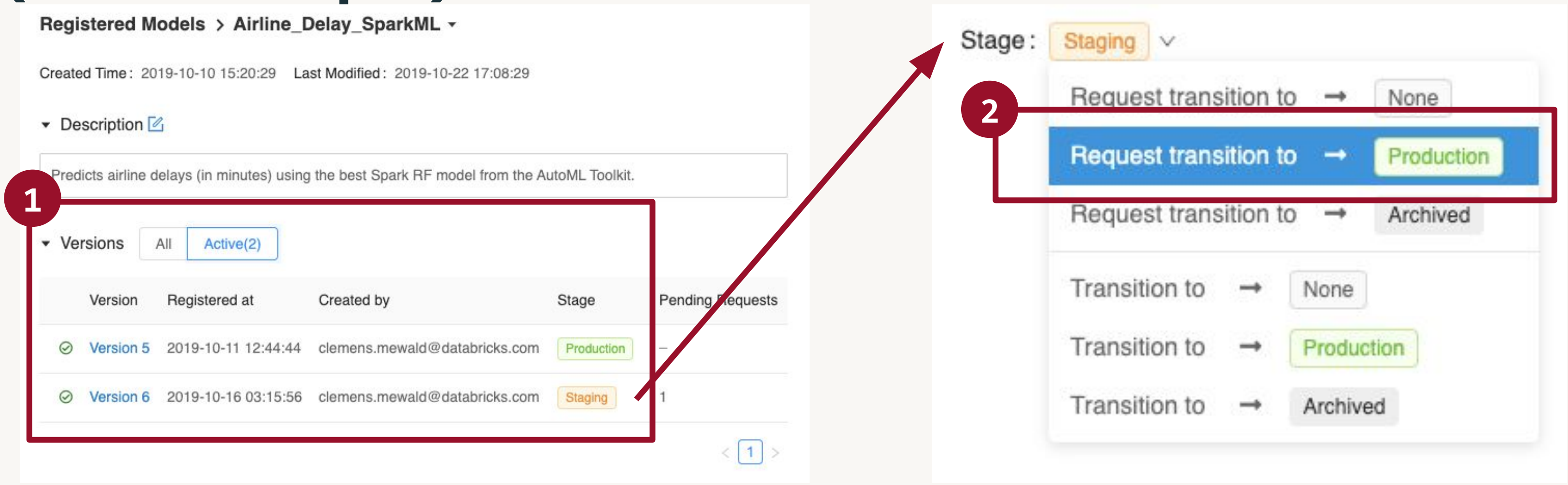
Pain Points

- 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 versions that are in Staging and Production
3. Search and filter models by name, stage, etc.

Manage the entire Model Lifecycle (ModelOps)



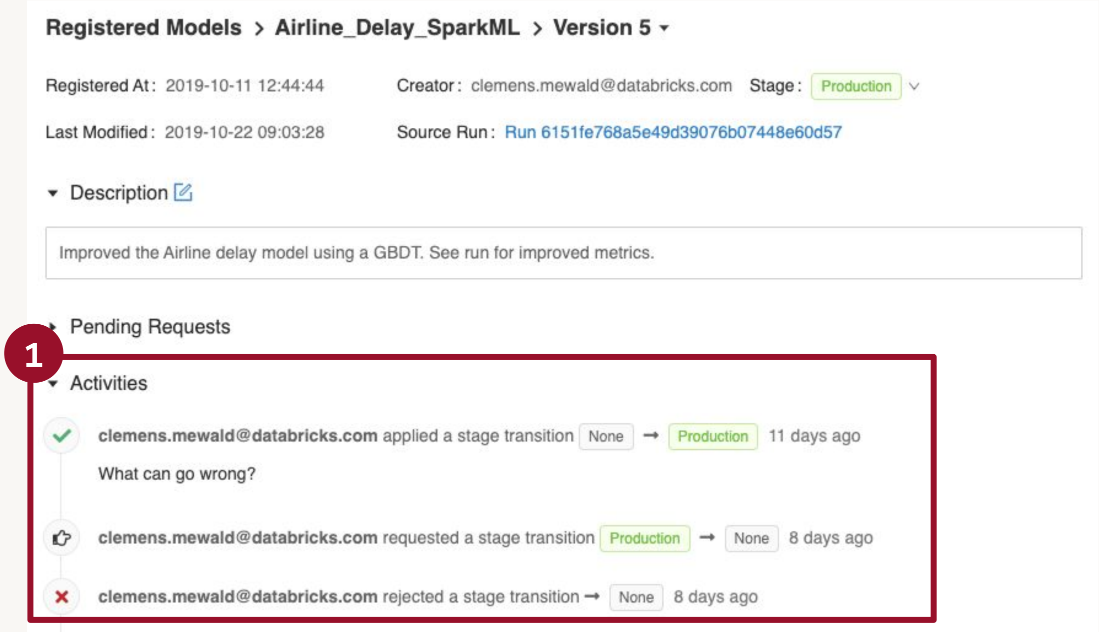
Pain Points

- 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



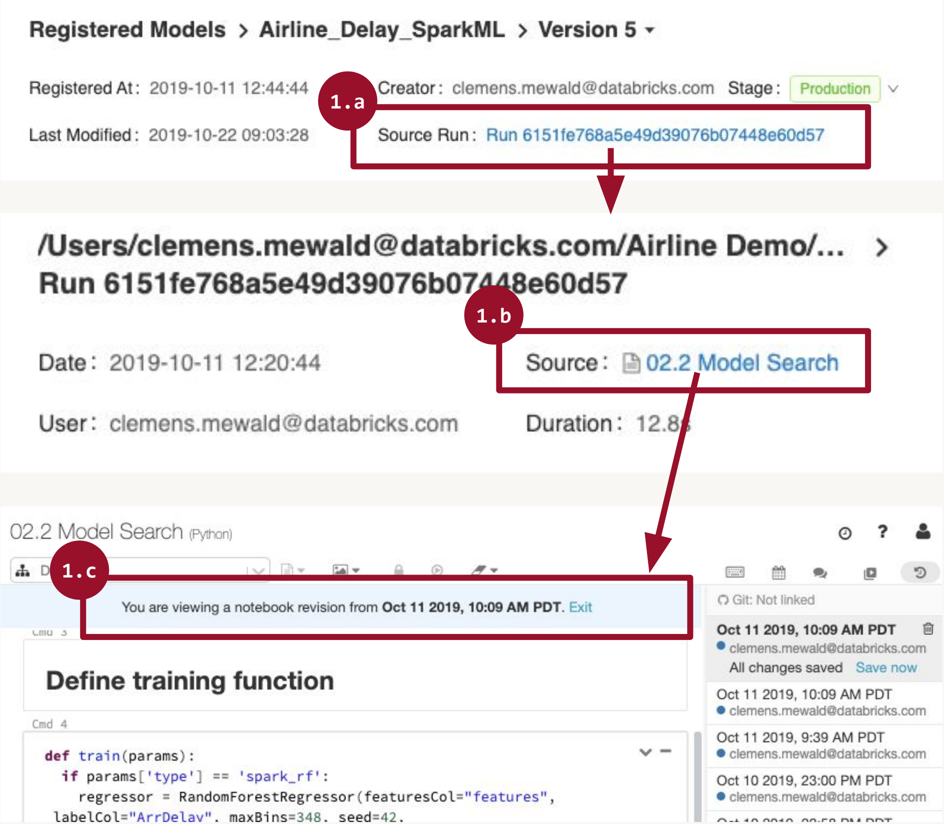
Pain Points

- No visibility into who put a model into production

Features

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

Governance and Auditability



Pain Points

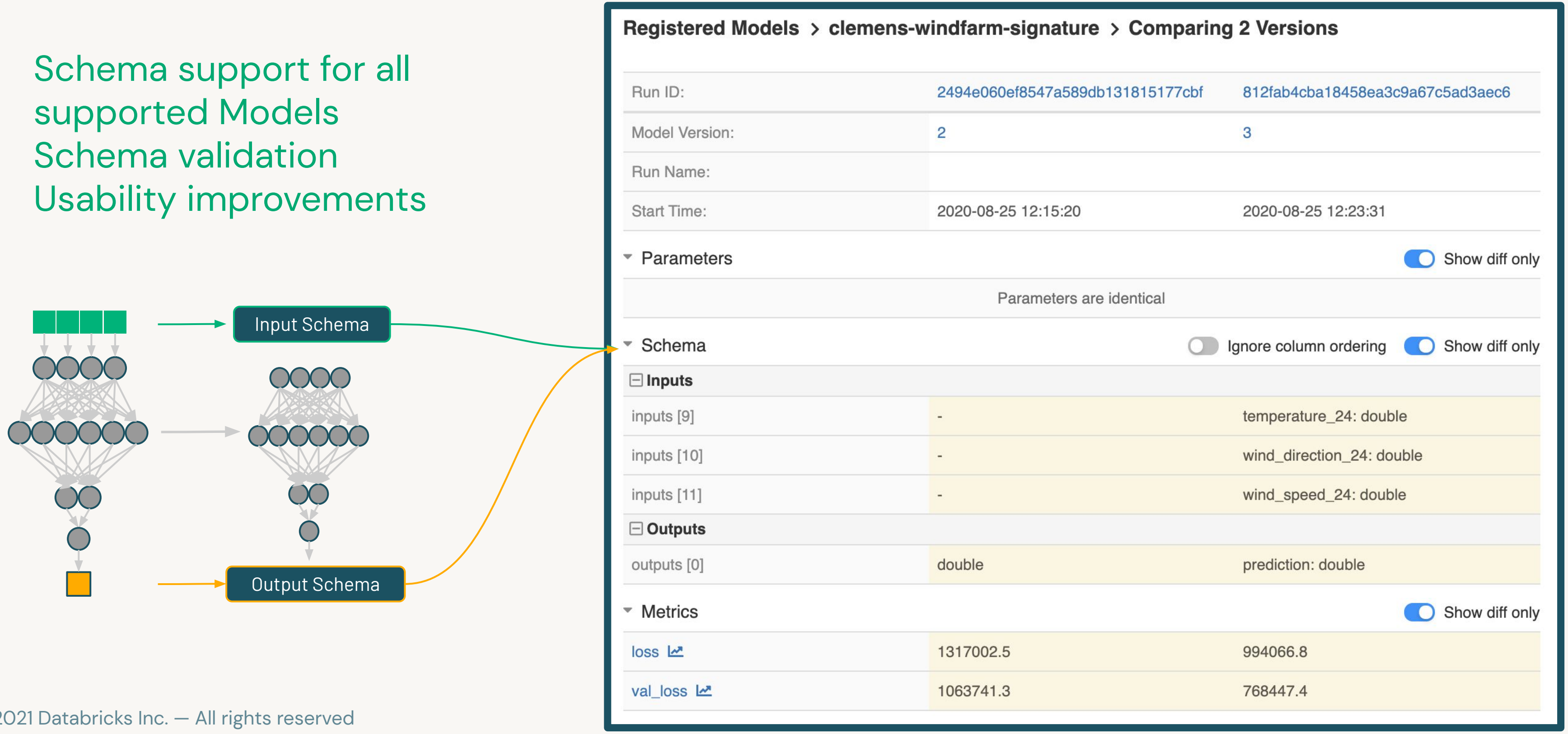
- For any given production model, it is impossible to find out where it came from

Features

1. Full provenance from Model marked production in the Registry to …
    a. Run that produced the model
    b. Notebook that produced the run
    c. Exact revision history of the notebook that produced the run

Model Schema

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

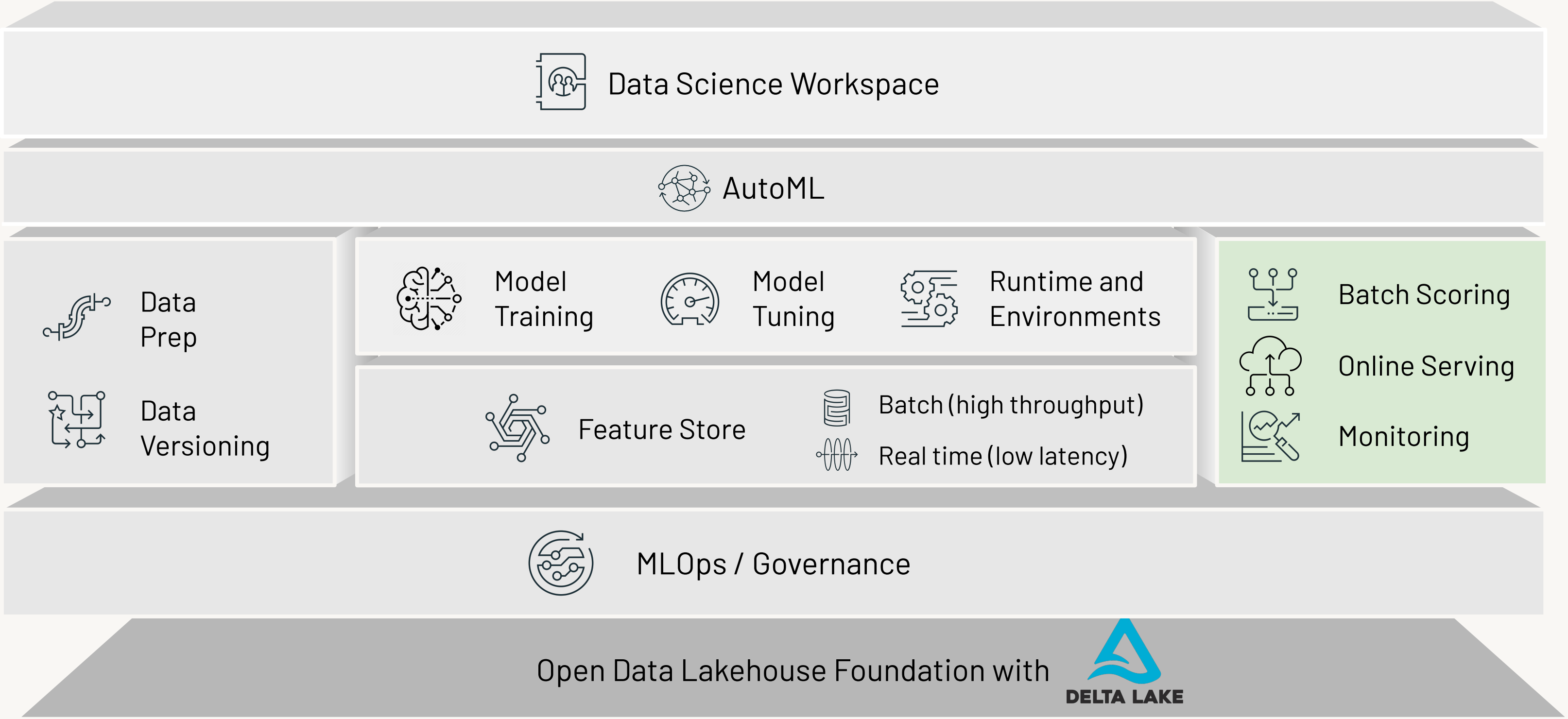


Exercise #3

MLflow Part 3
MLflow Deployment Monitoring

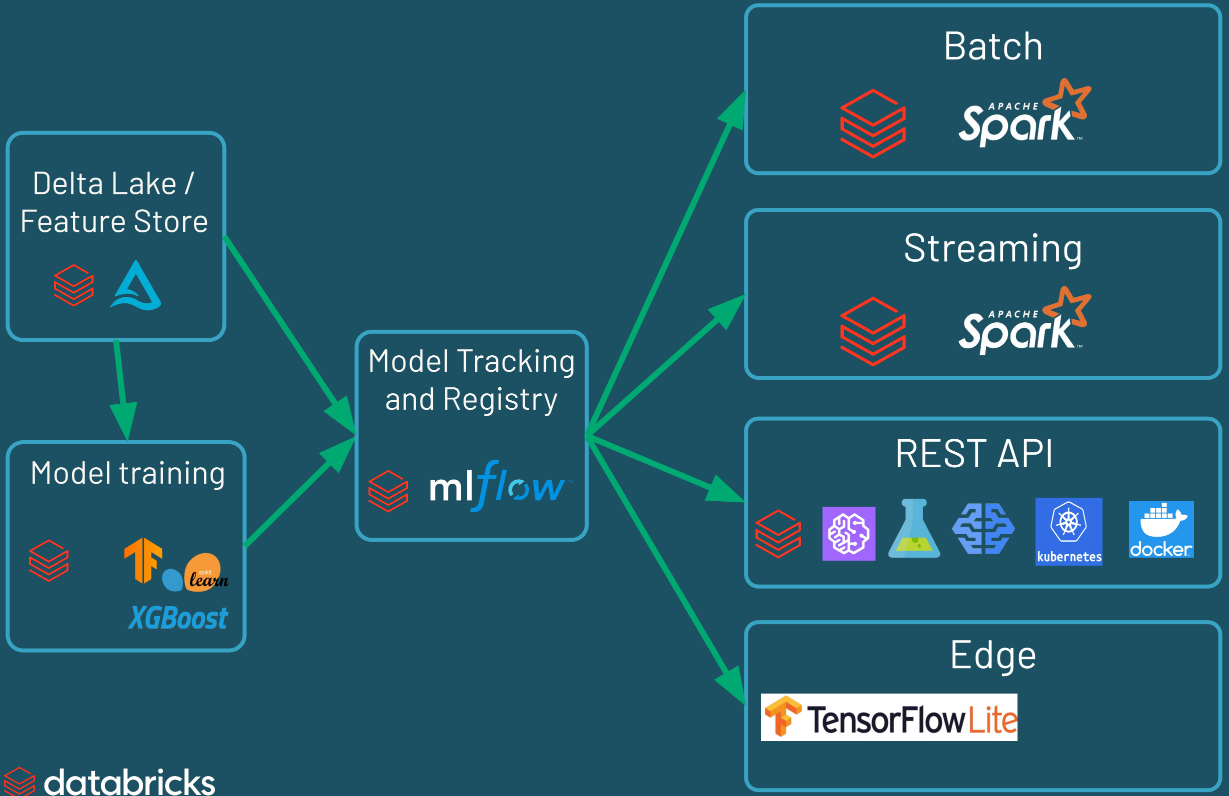
Databricks Machine Learning

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...

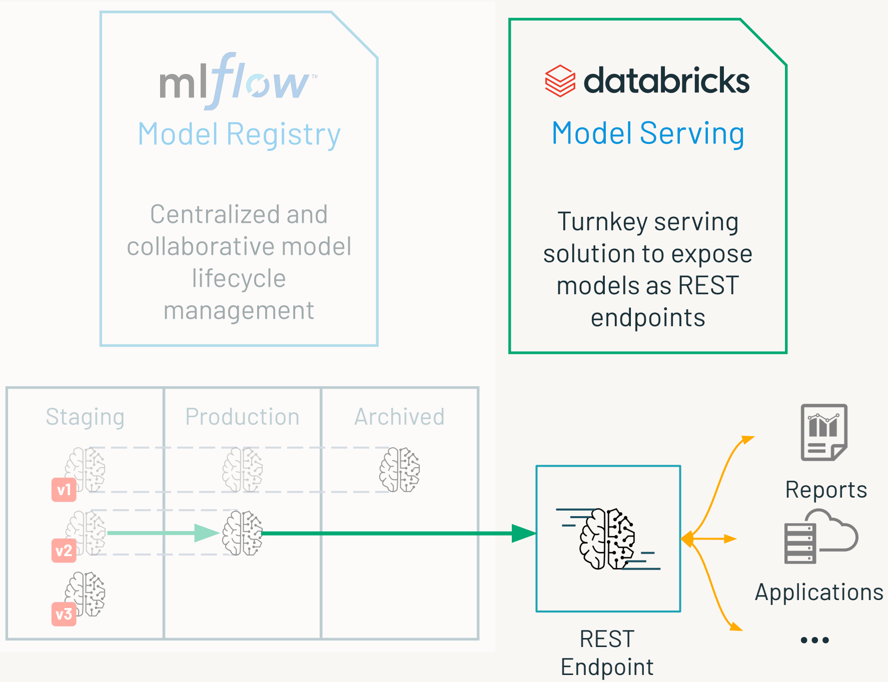
- A range of cost / latency trade-offs
- Databricks services, 3rd-party services, Feature Store model packages, and embedded applications

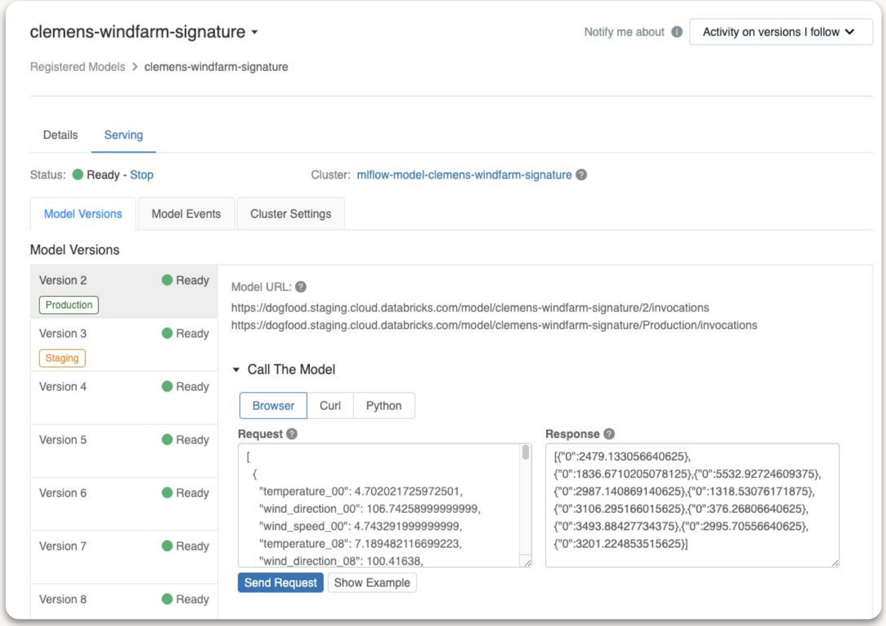
How to choose deployment paradigm?

	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



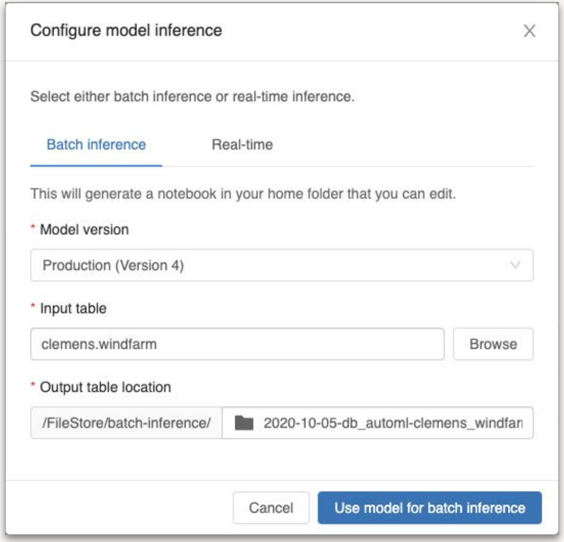


Model Deployment / Serving Options

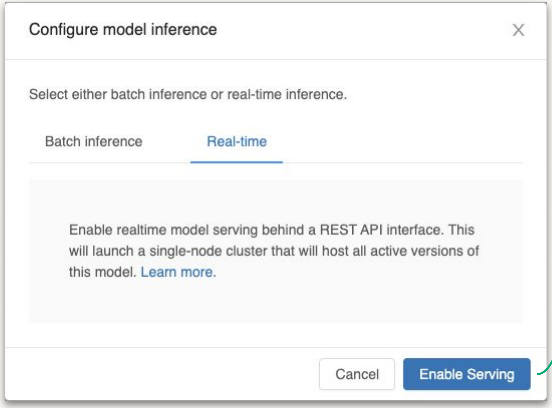
Deploy any ML model at large scale AND low latency

Use model for inference — One click model deployment directly from the Model Registry

Large-scale batch scoring



Low-latency online serving



Classic - Available in All Clouds

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

Resources

Deep Learning Model Inference Examples

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

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!