



# Simplifying the Machine Learning Lifecycle

# Agenda

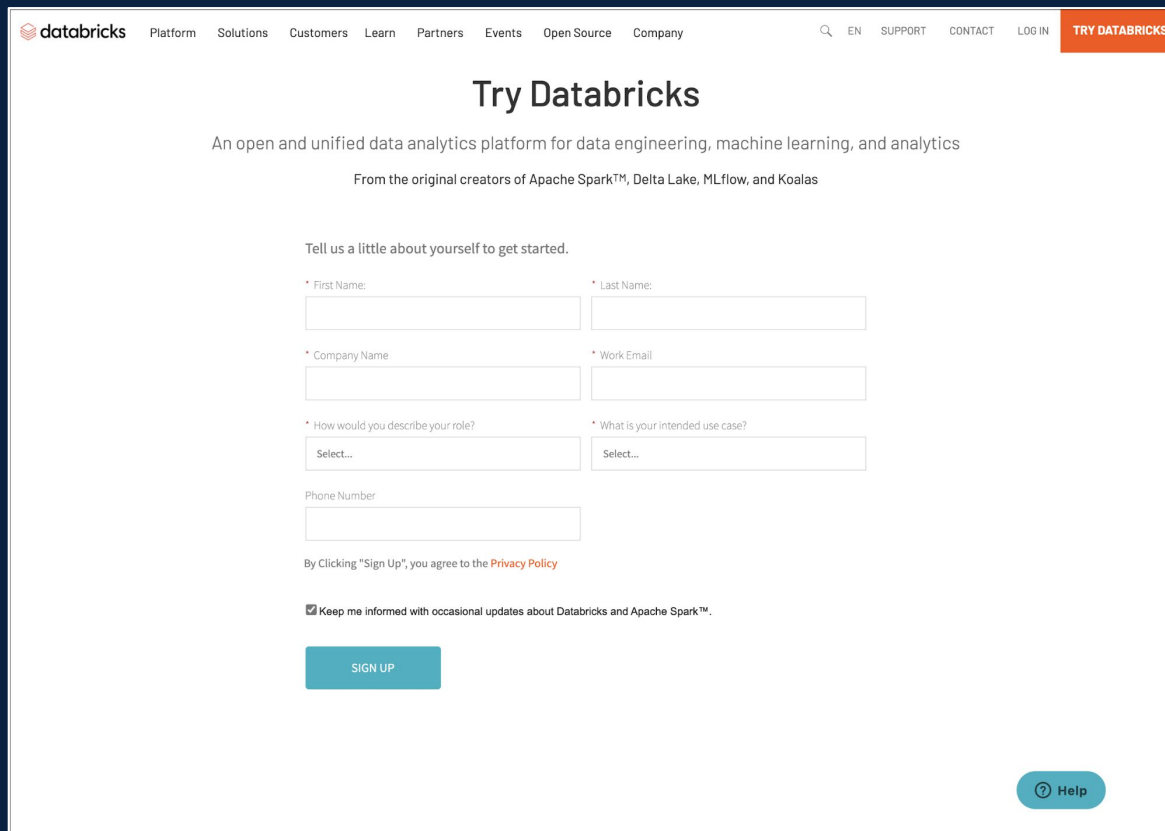
- / Broad Adoption of ML ... and its issues
- / The need for standardization
- / ML development challenges
- / How MLflow tackles these

# Login to Databricks Community Edition

<https://databricks.com/try>

- Sign up for Databricks Community Edition for free
- We will use this for the tutorial
- Once you sign up, you can continue to use it to learn and experiment on a dedicated data sciences engineering environment

# Go to databricks.com/try



The screenshot shows the 'Try Databricks' sign-up page. At the top, there is a navigation bar with the Databricks logo and links for Platform, Solutions, Customers, Learn, Partners, Events, Open Source, and Company. On the right side of the navigation bar, there are links for Search, EN, SUPPORT, CONTACT, LOGIN, and a prominent orange button labeled 'TRY DATABRICKS'. The main heading is 'Try Databricks', followed by the tagline 'An open and unified data analytics platform for data engineering, machine learning, and analytics' and the text 'From the original creators of Apache Spark™, Delta Lake, MLflow, and Koalas'. Below this, a section titled 'Tell us a little about yourself to get started.' contains a form with the following fields: 'First Name' and 'Last Name' (text input), 'Company Name' and 'Work Email' (text input), 'How would you describe your role?' and 'What is your intended use case?' (dropdown menus with 'Select...' options), and 'Phone Number' (text input). A checkbox is checked, with the text 'Keep me informed with occasional updates about Databricks and Apache Spark™'. Below the form is a blue 'SIGN UP' button. In the bottom right corner, there is a 'Help' button with a question mark icon.

databricks Platform Solutions Customers Learn Partners Events Open Source Company

EN SUPPORT CONTACT LOGIN TRY DATABRICKS

## Try Databricks

An open and unified data analytics platform for data engineering, machine learning, and analytics

From the original creators of Apache Spark™, Delta Lake, MLflow, and Koalas

Tell us a little about yourself to get started.

\* First Name:

\* Last Name:

\* Company Name:

\* Work Email:

\* How would you describe your role?

\* What is your intended use case?

Phone Number:


By Clicking "Sign Up", you agree to the [Privacy Policy](#)

Keep me informed with occasional updates about Databricks and Apache Spark™.

SIGN UP

Help

# Sign up for Community Edition

 Platform Solutions Customers Learn Partners Events Open Source Company EN SUPPORT CONTACT LOGIN **TRY DATABRICKS**

## Try Databricks

An open and unified data analytics platform for data engineering, machine learning, and analytics

From the original creators of Apache Spark™, Delta Lake, MLflow, and Koalas

Select a platform

### DATABRICKS PLATFORM - FREE TRIAL

For businesses

- Collaborative environment for Data teams to build solutions together
- Unlimited clusters that can scale to any size, processing data in your own account
- Job scheduler to execute jobs for production pipelines
- Fully collaborative notebooks with multi-language support, dashboards, REST APIs
- Native integration with the most popular ML frameworks (scikit-learn, TensorFlow, Keras,...), Apache Spark™, Delta Lake, and MLflow
- Advanced security, role-based access controls, and audit logs
- Single Sign On support
- Integration with BI tools such as Tableau, Olik, and Looker
- 14-day full feature trial (excludes cloud charges)

### COMMUNITY EDITION



For students and educational institutions

- Single cluster limited to 6GB and no worker nodes
- Basic notebooks without collaboration
- Limited to 3 max users
- Public environment to share your work

[GET STARTED](#)

By clicking "Get Started" for the Community Edition, you agree to the [Databricks Community Edition Terms of Service](#).

**GET STARTED ON**

 **Azure** OR  **aws**

Please note that Azure Databricks is provided by Microsoft and is subject to Microsoft's terms.

By clicking on the "AWS" button to get started, you agree to the [Databricks Terms of Service](#).


[Help](#)


# Sign up for Community Edition

Login - Databricks Community Edition X

https://community.cloud.databricks.com/login.html

Important Notice: [Acceptable use and unused account termination policy](#) and [Terms of Use](#) update.

 databricks

 Sign In to Databricks

[Forgot Password?](#)

[Sign In](#)

New to Databricks? [Sign Up](#).

[Privacy Policy](#) | [Terms of Use](#)

# Log into DBCE

The screenshot shows the Databricks Community Edition web interface. The browser address bar displays `https://community.cloud.databricks.com/?o=57901`. The page features a dark sidebar on the left with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area is titled "Welcome to databricks" and includes three primary action cards: "Explore the Quickstart Tutorial", "Import & Explore Data", and "Create a Blank Notebook". Below these are three sections: "Common Tasks" with links for New Notebook, Upload Data, Create Table, New Cluster, New Job, Import Library, and Read Documentation; "Recents" with a link for Video Identification of Suspicious Behavior; and "What's new in v2.100" with a link for Databricks Light GA and a link to View latest release notes.

Databricks Community Edition

Home

Workspace

Recents

Data

Clusters

Jobs

Search

Upgrade ?

## Welcome to databricks

Drop files or [click to browse](#)

### Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

### Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.

### Create a Blank Notebook

Create a notebook to start querying, visualizing, and modeling your data.

#### Common Tasks

- [New Notebook](#)
- [Upload Data](#)
- [Create Table](#)
- [New Cluster](#)
- [New Job](#)
- [Import Library](#)
- [Read Documentation](#)

#### Recents

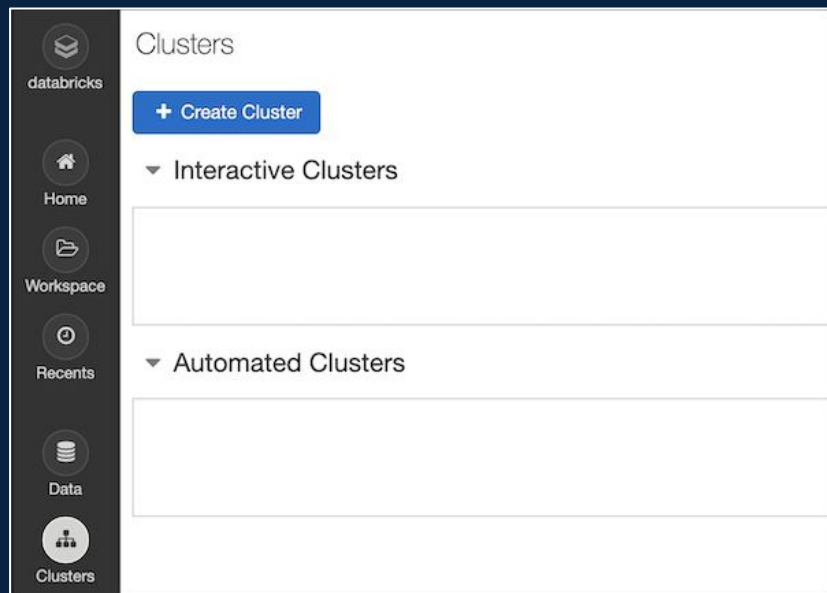
- [Video Identification of Suspicious Behavior](#)

#### What's new in v2.100

- [Databricks Light GA](#)

[View latest release notes](#)

# Create a Cluster on DBCE





# Create a Cluster on Databricks

Create Cluster

New Cluster Cancel Create Cluster **0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU**  
**1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU**

Cluster Name

Databricks Runtime Version **?**  
Runtime: 5.5 LTS (Scala 2.11, Spark 2.4.3) | v

Python Version **?**  
3 | v

Instance

Free 6GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances Spark

Availability Zone **?**  
us-west-2c | v

# Create a Cluster on DBCE

datadricks

Home

Workspace

Recents

Data

Clusters

Jobs

Search

## Create Cluster

### New Cluster

Cancel Create Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU  
1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU

Cluster Name  
delta-rocks

Databricks Runtime Version  
Runtime: 6.1 Beta (Scala 2.11, Spark 2.4.4)

**Databricks Runtime**

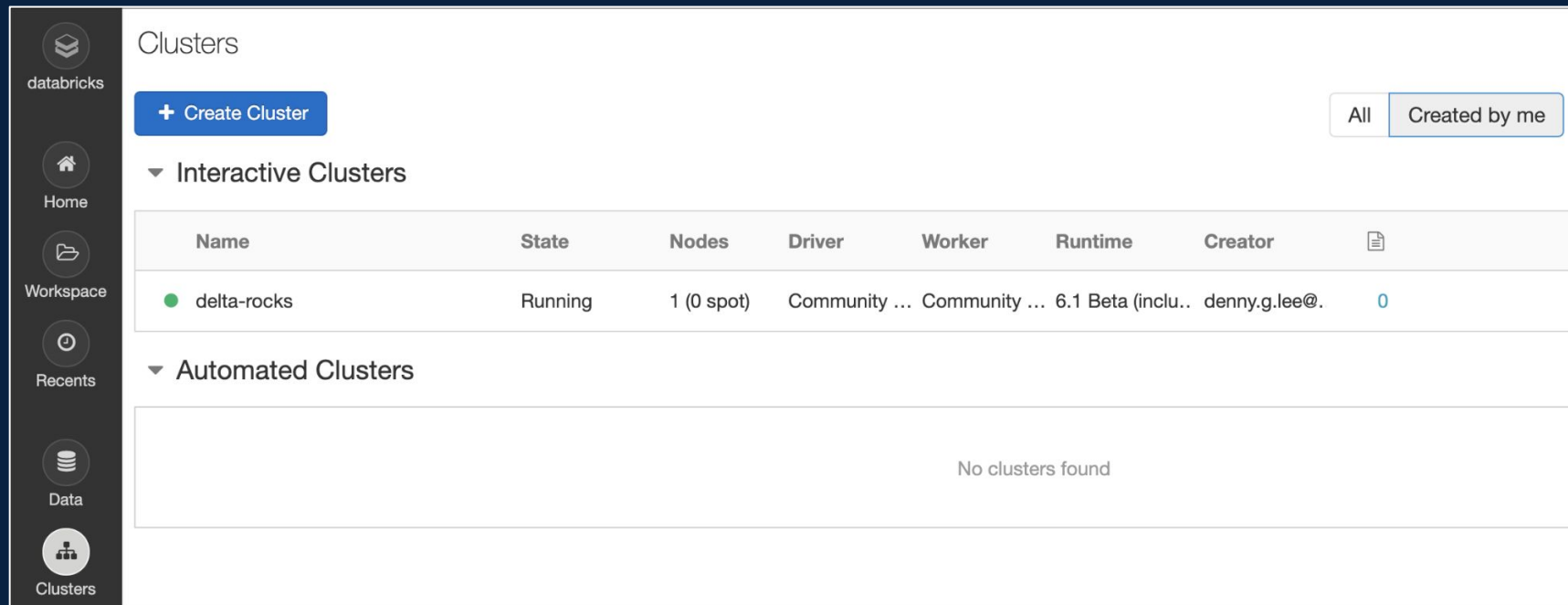
- 6.1 Beta Scala 2.11, Spark 2.4.4
- 6.1 ML Beta GPU, Scala 2.11, Spark 2.4.4
- 6.1 ML Beta Scala 2.11, Spark 2.4.4
- 20 more

Automatically terminate after an idle period of two hours. subscription.

Instances Spark

Availability Zone  
us-west-2c

# Create a Cluster on DBCE

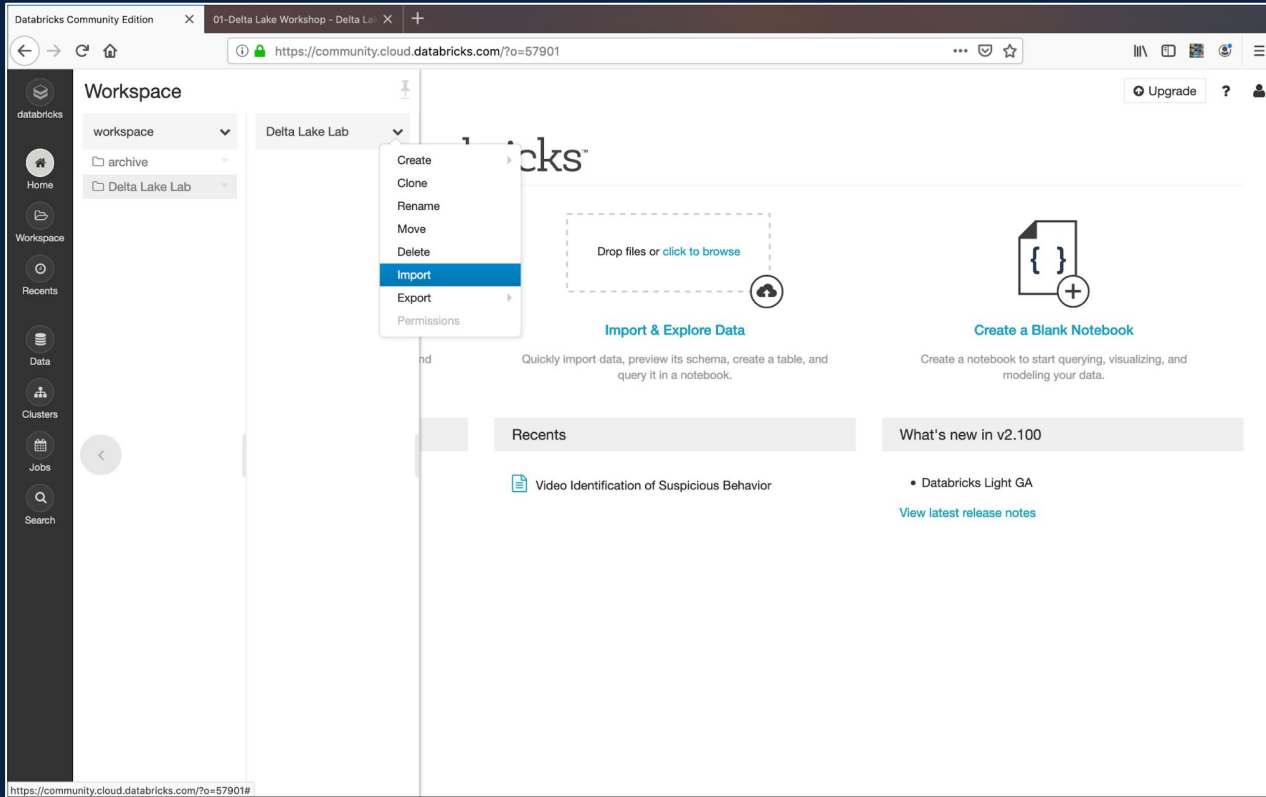


The screenshot displays the Databricks Clusters management interface. On the left is a dark sidebar with navigation icons for Databricks, Home, Workspace, Recents, Data, and Clusters. The main content area is titled 'Clusters' and features a '+ Create Cluster' button. Below this, there are two sections: 'Interactive Clusters' and 'Automated Clusters'. The 'Interactive Clusters' section contains a table with one cluster entry.

Name	State	Nodes	Driver	Worker	Runtime	Creator	
delta-rocks	Running	1 (0 spot)	Community ...	Community ...	6.1 Beta (inclu..	denny.g.lee@.	0

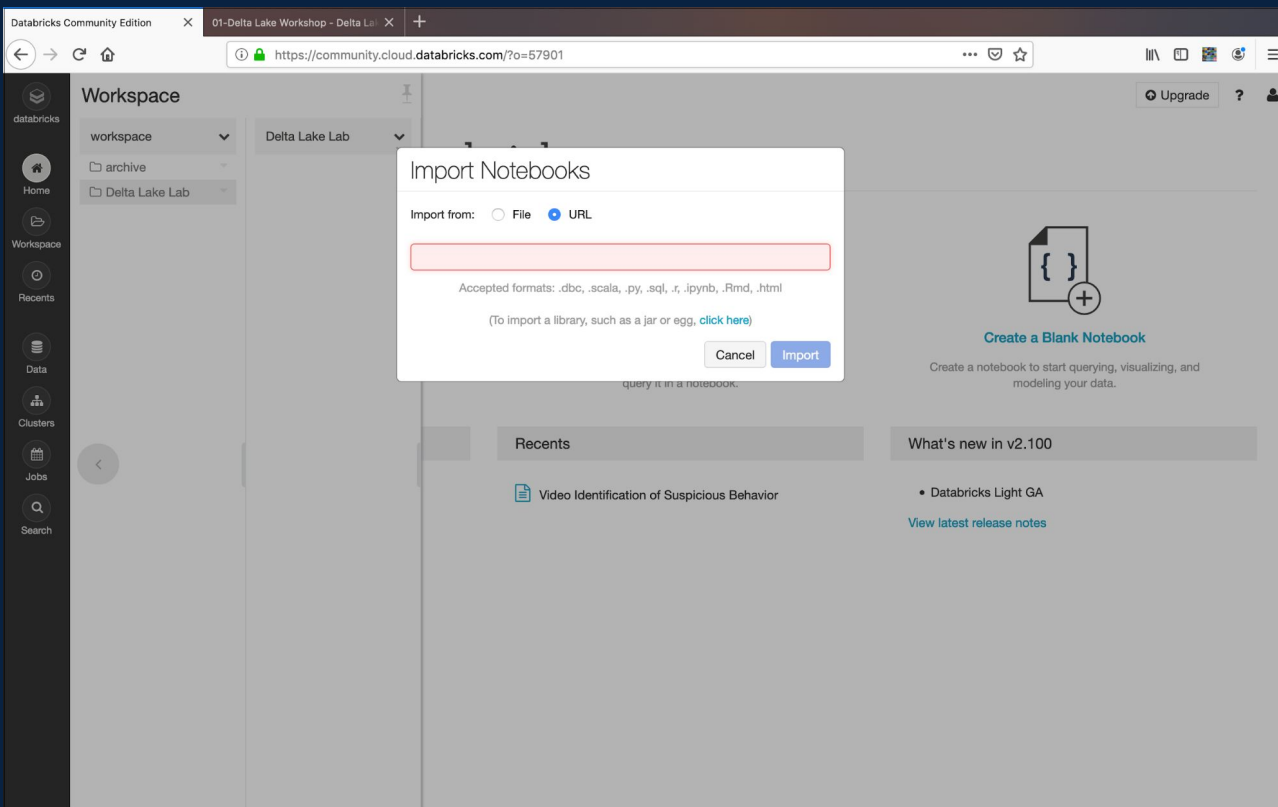
The 'Automated Clusters' section is currently empty, displaying the message 'No clusters found'.

# Attach a Notebook to your Cluster



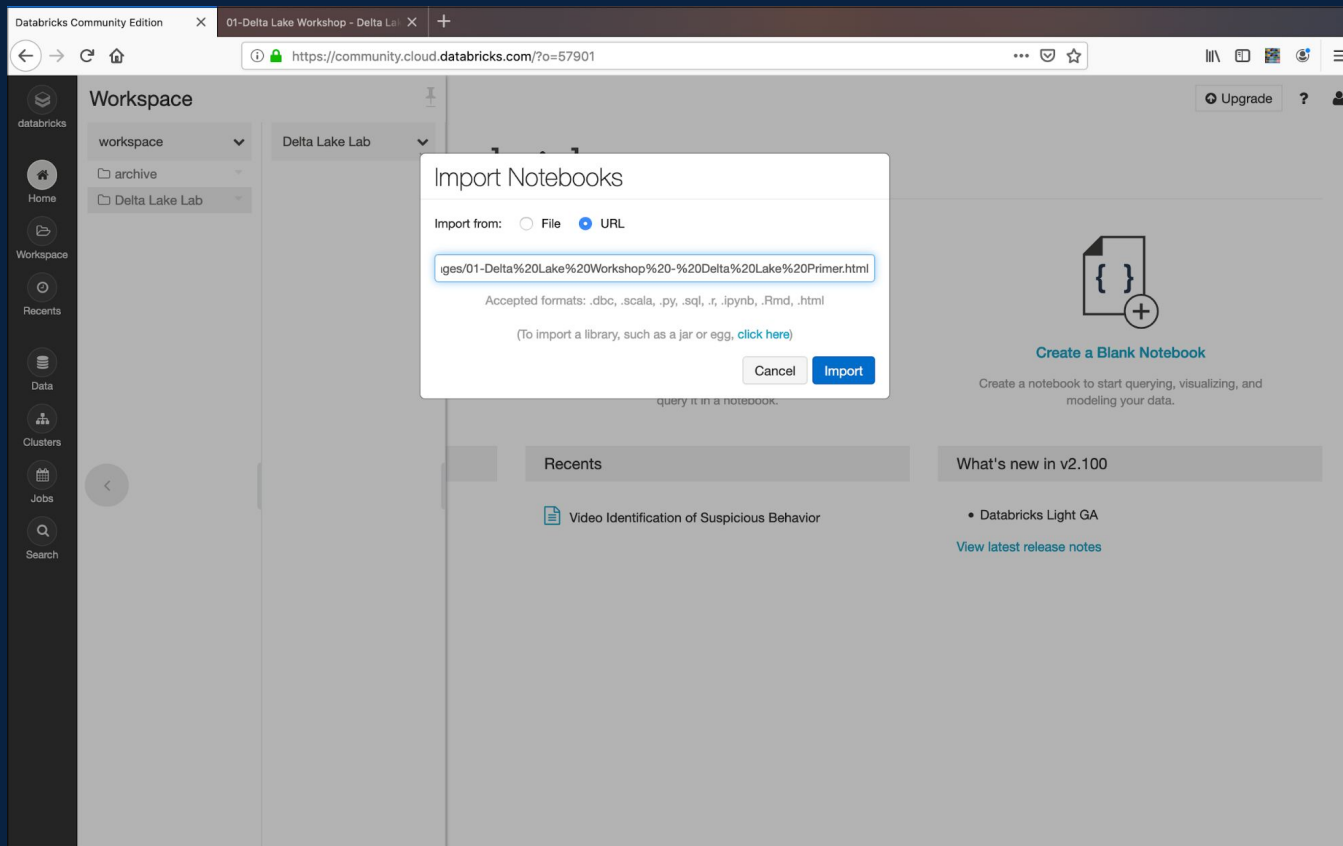
The screenshot displays the Databricks Community Edition web interface. The browser address bar shows the URL <https://community.cloud.databricks.com/?o=57901>. The main workspace area is titled "Workspace" and shows a folder structure with "archive" and "Delta Lake Lab". A context menu is open over a cluster, listing actions: Create, Clone, Rename, Move, Delete, **Import** (highlighted), Export, and Permissions. The "Import & Explore Data" section includes a dashed box with the text "Drop files or click to browse" and a cloud upload icon. Below this, it says "Import & Explore Data" and "Quickly import data, preview its schema, create a table, and query it in a notebook." The "Create a Blank Notebook" section features a notebook icon with a plus sign and the text "Create a Blank Notebook" and "Create a notebook to start querying, visualizing, and modeling your data." The "Recents" section lists "Video Identification of Suspicious Behavior". The "What's new in v2.100" section lists "Databricks Light GA" and a link to "View latest release notes". The left sidebar contains navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The bottom of the screenshot shows the URL <https://community.cloud.databricks.com/?o=57901#>.

# Attach a Notebook to your Cluster



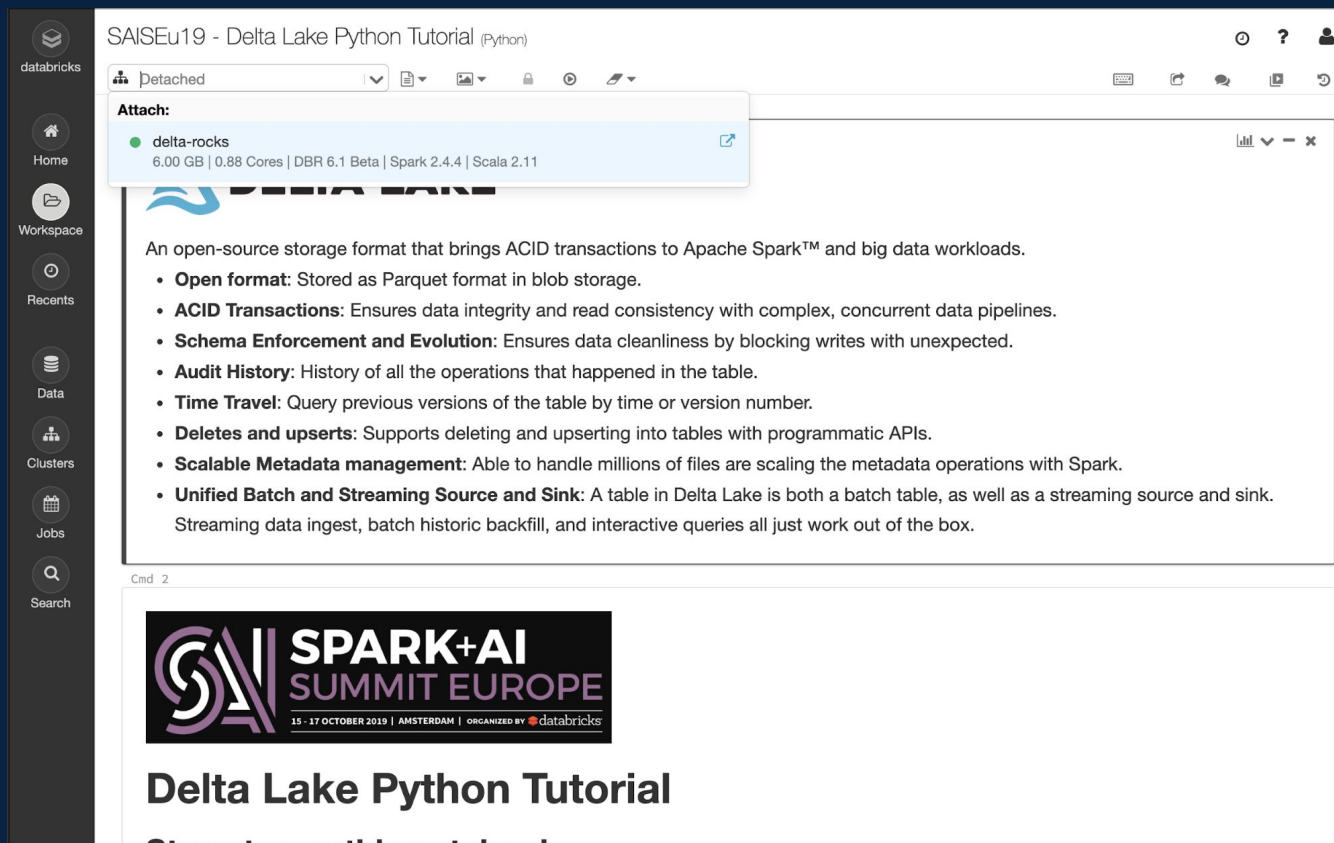
The screenshot displays the Databricks Community Edition web interface. The browser address bar shows the URL <https://community.cloud.databricks.com/?o=57901>. The main workspace area is titled "Workspace" and shows a folder structure for "Delta Lake Lab". A modal dialog box titled "Import Notebooks" is centered on the screen. It features two radio buttons for "Import from": "File" (unselected) and "URL" (selected). Below the radio buttons is a red-outlined text input field. Underneath the input field, it lists "Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html" and includes a link "(To import a library, such as a jar or egg, [click here](#))". At the bottom of the dialog are "Cancel" and "Import" buttons. In the background, the workspace sidebar on the left includes navigation options like Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area on the right features a "Create a Blank Notebook" button with a plus icon, and a "What's new in v2.100" section listing "Databricks Light GA" with a link to "View latest release notes".

# Attach a Notebook to your Cluster



The screenshot displays the Databricks Community Edition web interface. A modal dialog titled "Import Notebooks" is centered on the screen. The dialog has two radio buttons for "Import from": "File" (unselected) and "URL" (selected). A text input field contains the URL: `iges/01-Delta%20Lake%20Workshop%20-%20Delta%20Lake%20Primer.html`. Below the input field, it lists "Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html" and includes a link "(To import a library, such as a jar or egg, [click here](#))". At the bottom of the dialog are "Cancel" and "Import" buttons. The background interface shows a sidebar with navigation options like Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main workspace area displays a "Delta Lake Lab" cluster and a "Create a Blank Notebook" button with a plus sign icon. A "Recents" section lists "Video Identification of Suspicious Behavior".

# Attach a Notebook to your Cluster



The screenshot shows the Databricks workspace interface. On the left is a sidebar with navigation icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main area displays a notebook titled "SAISEu19 - Delta Lake Python Tutorial (Python)". At the top, a toolbar shows the notebook is "Detached". An "Attach:" dialog box is open, showing a cluster named "delta-rocks" with specifications: 6.00 GB | 0.88 Cores | DBR 6.1 Beta | Spark 2.4.4 | Scala 2.11. The notebook content includes a paragraph about Delta Lake and a bulleted list of features.

SAISEu19 - Delta Lake Python Tutorial (Python)

Detached

**Attach:**

- delta-rocks  
6.00 GB | 0.88 Cores | DBR 6.1 Beta | Spark 2.4.4 | Scala 2.11

An open-source storage format that brings ACID transactions to Apache Spark™ and big data workloads.

- **Open format:** Stored as Parquet format in blob storage.
- **ACID Transactions:** Ensures data integrity and read consistency with complex, concurrent data pipelines.
- **Schema Enforcement and Evolution:** Ensures data cleanliness by blocking writes with unexpected.
- **Audit History:** History of all the operations that happened in the table.
- **Time Travel:** Query previous versions of the table by time or version number.
- **Deletes and upserts:** Supports deleting and upserting into tables with programmatic APIs.
- **Scalable Metadata management:** Able to handle millions of files are scaling the metadata operations with Spark.
- **Unified Batch and Streaming Source and Sink:** A table in Delta Lake is both a batch table, as well as a streaming source and sink. Streaming data ingest, batch historic backfill, and interactive queries all just work out of the box.

Cmd 2

**SPARK+AI SUMMIT EUROPE**  
15 - 17 OCTOBER 2019 | AMSTERDAM | ORGANIZED BY databricks

## Delta Lake Python Tutorial

# Broad Adoption of ML

*Huge disruptive innovations are affecting most enterprises on the planet*



Healthcare and Genomics



Fraud Prevention



Digital Personalization



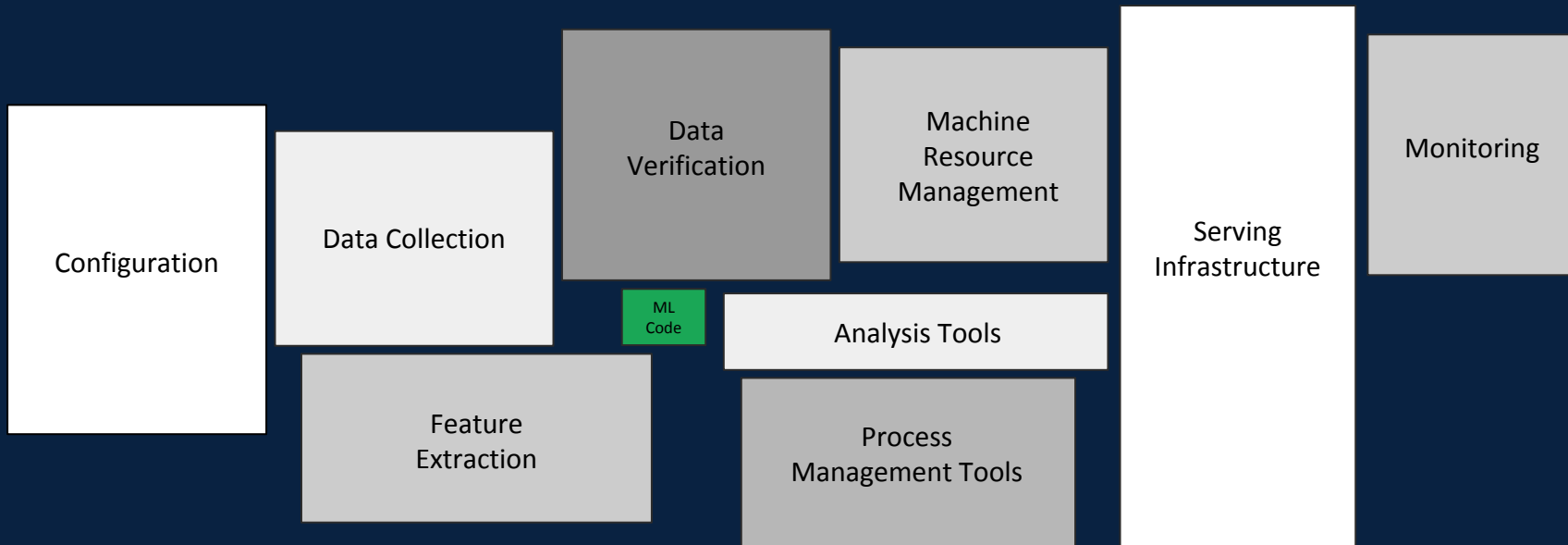
Internet of Things

*and many many more customers in different industries and segments*



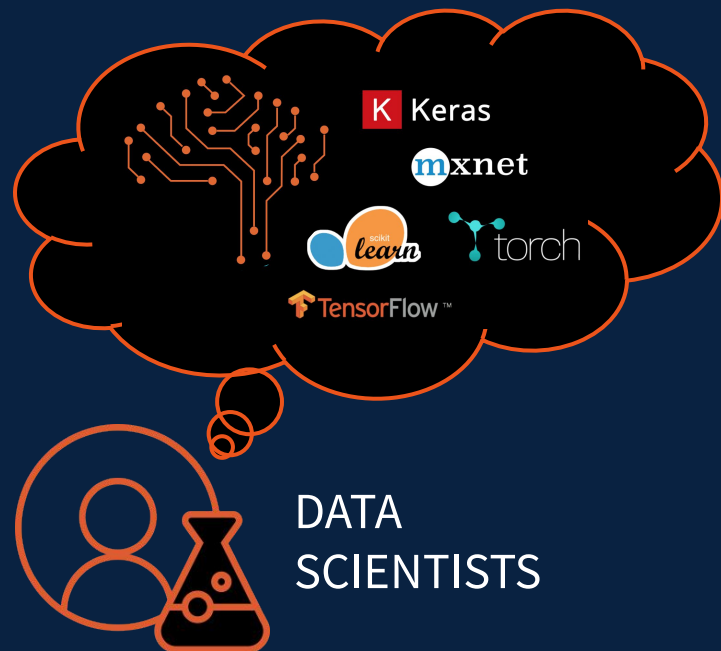
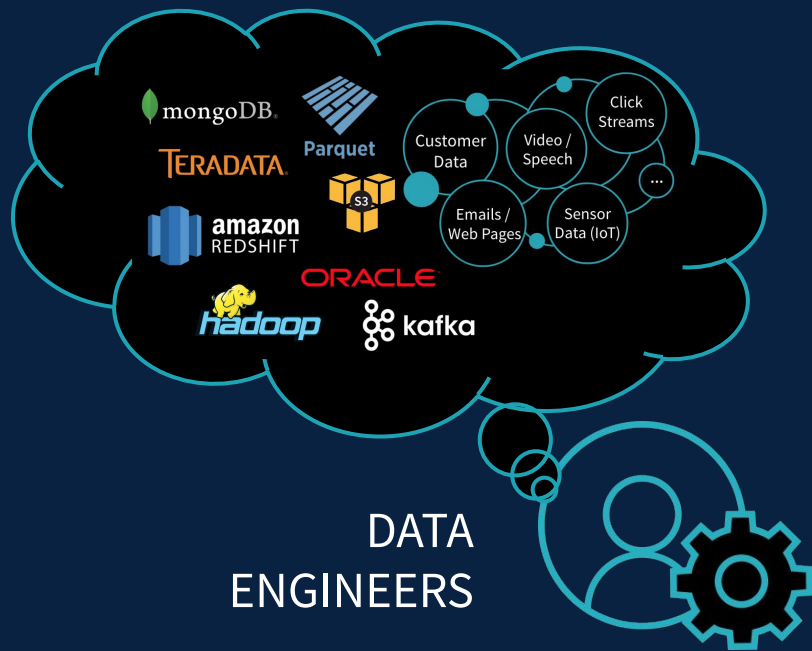
# Hardest Part of ML isn't ML, it's Data

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



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

# Data & ML Tech and People are in Silos



# ML Lifecycle is Manual, Inconsistent and Disconnected

## Prep Data

- Low level integrations for Data and ML
- Difficult to track data used for a model



## Build Model

- Ad hoc approach to track experiments
- Very hard to reproduce experiments

GitHub

CONDA



## Deploy Model

- Multiple tightly coupled deployment options
- Different monitoring approach for each framework



kubernetes



docker



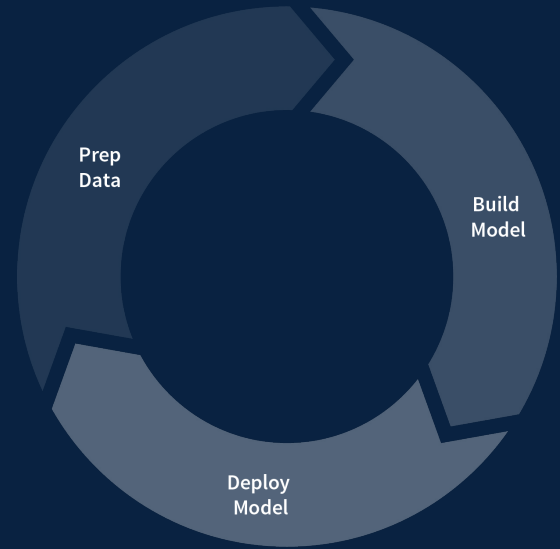
Amazon SageMaker



Azure Machine Learning



# The need for standardization



# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

```
R2: 0.2291130640003659
```

# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

```
R2: 0.2291130640003659
```



Did anything change in the feature engineering?

# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

```
R2: 0.2291130640003659
```



How did the hyperparameters change?

# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

```
R2: 0.2291130640003659
```



What data was this model trained on?



# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

```
R2: 0.2291130640003659
```



How did the offline metrics change?

# Day in the life of a data scientist (tracking edition)

```
Elasticnet model (alpha=0.01, l1_ratio=1.0):
```

```
RMSE: ??
```

```
MAE: 51.051828604086325
```

```
R2: 0.3951809598912357
```

```
Elasticnet model (alpha=?, l1_ratio=0.75):
```

```
RMSE: 65.28994906390733
```

```
MAE: 53.759148284349266
```

```
R2: ??
```

```
Elasticnet model (alpha=0.01, l1_ratio=?):
```

```
RMSE: 71.40362571026475
```

```
MAE: ??
```

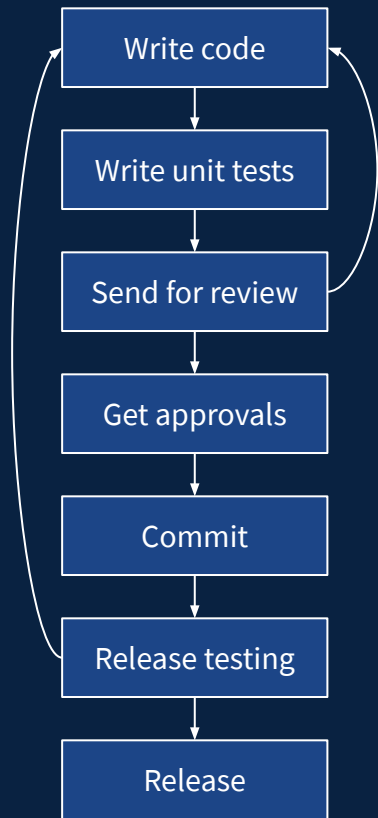
```
R2: 0.2291130640003659
```



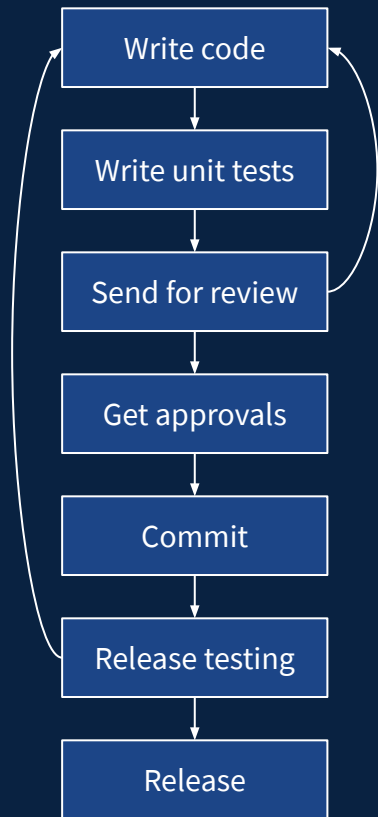
What else am I missing?

# The difference between releasing Software and deploying ML Models

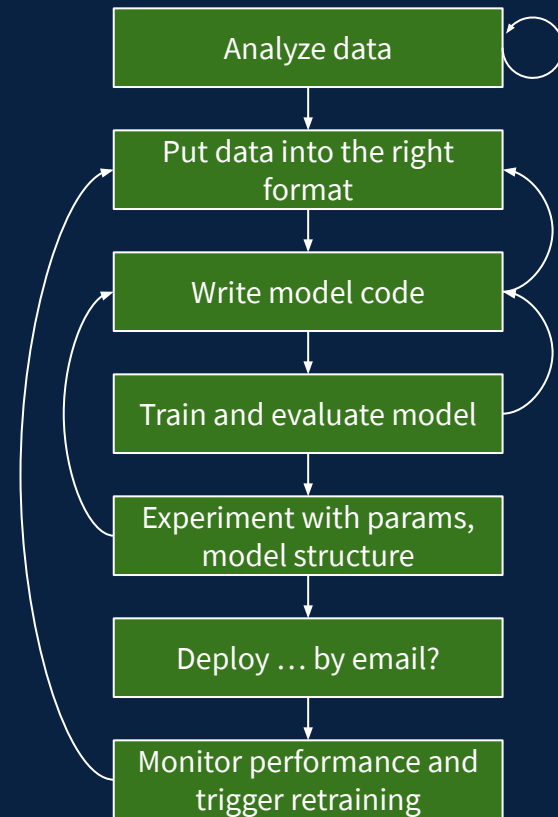
# Software



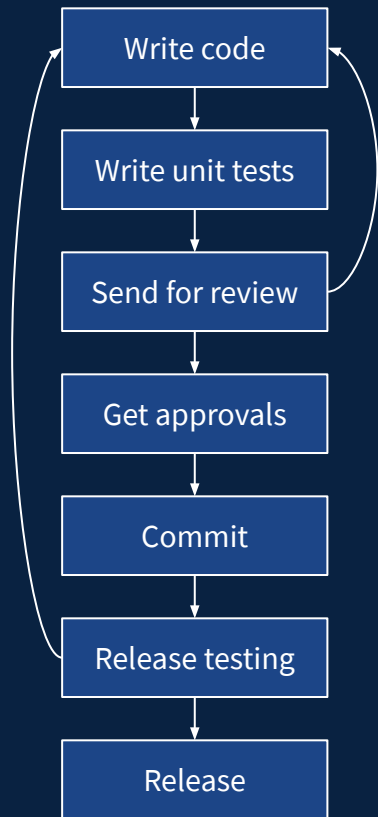
# Software



# ML Models



# Software

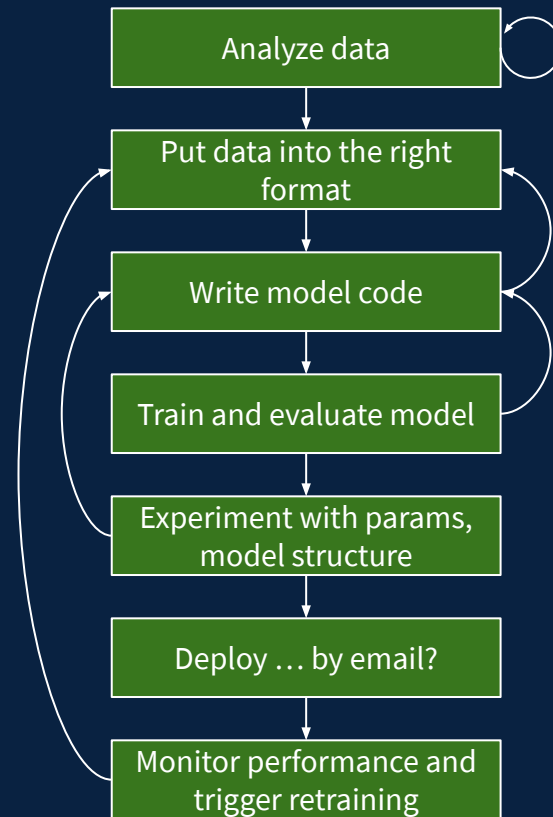


## Goal

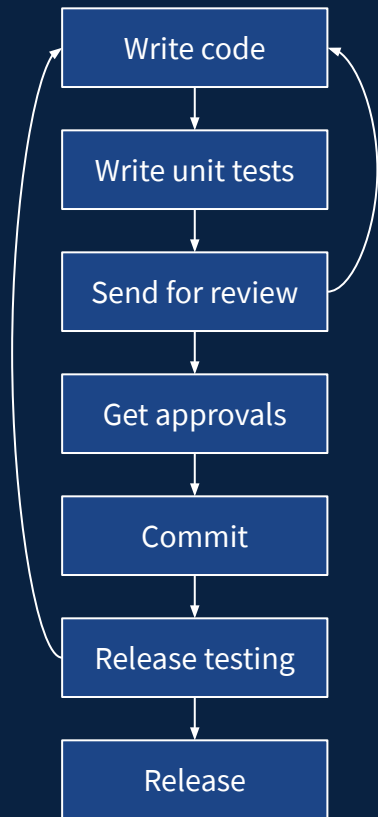
Meet a functional specification

Optimize a metric, e.g. CTR

# ML Models



# Software



## Goal

Meet a functional specification

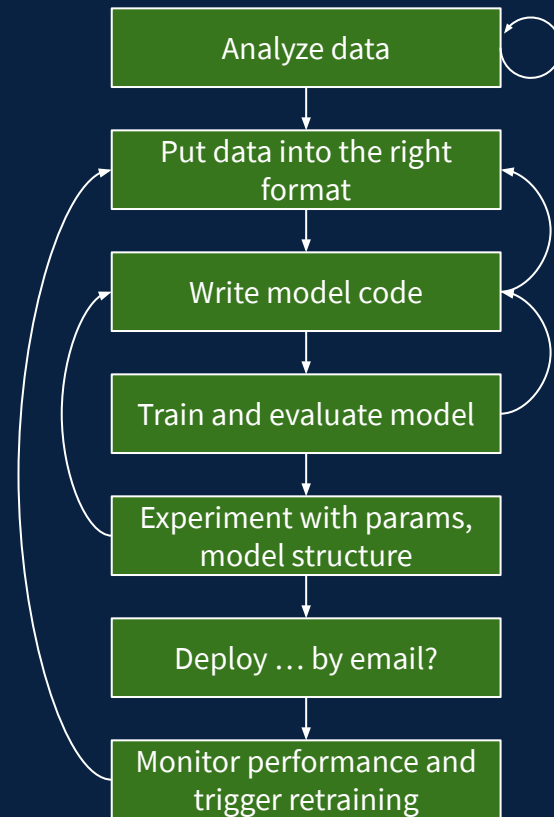
Optimize a metric, e.g. CTR

## Quality

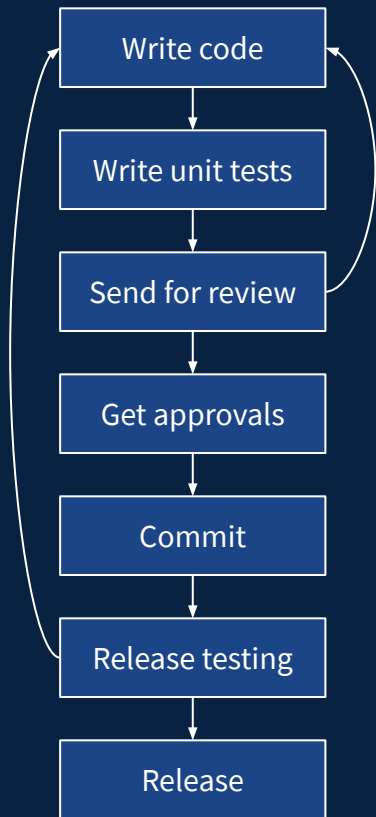
Depends on code

Depends on data, code, model, params, ...

# ML Models



# Software



## Goal

Meet a functional specification

Optimize a metric, e.g. CTR

## Quality

Depends on code

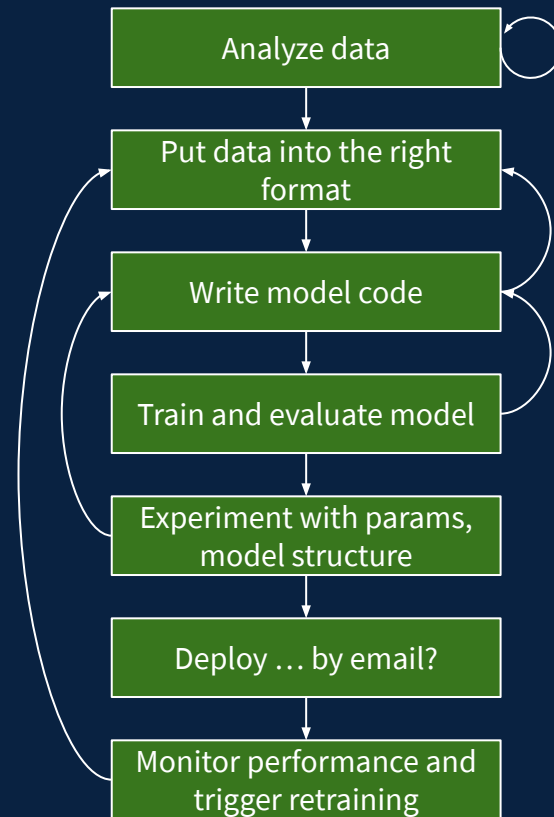
Depends on data, code, model, params, ...

## Tools

Typically one software stack

Combination of many libraries, tools, ...

# ML Models





# Software



## Goal

Meet a functional specification

Optimize a metric, e.g. CTR

## Quality

Depends on code

Depends on data, code, model, params, ...

## Tools

Typically one software stack

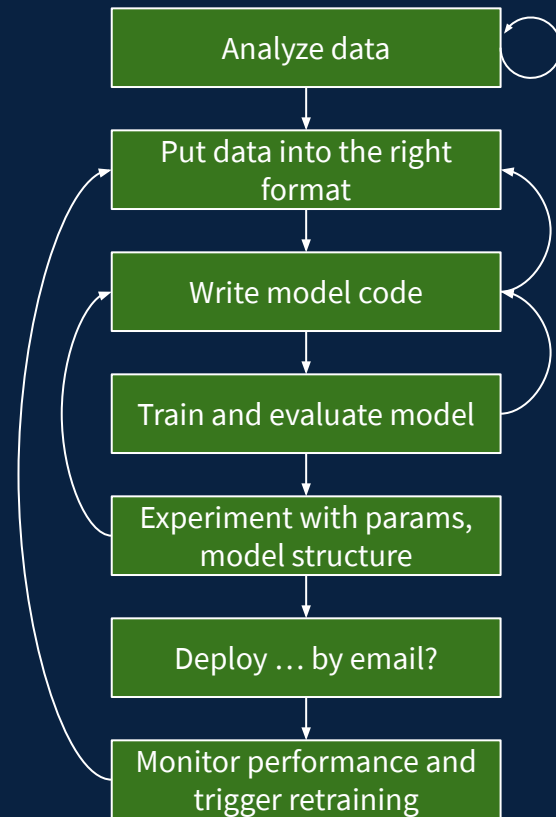
Combination of many libraries, tools, ...

## Outcome

Works deterministically

Keeps changing with data, etc.

# ML Models

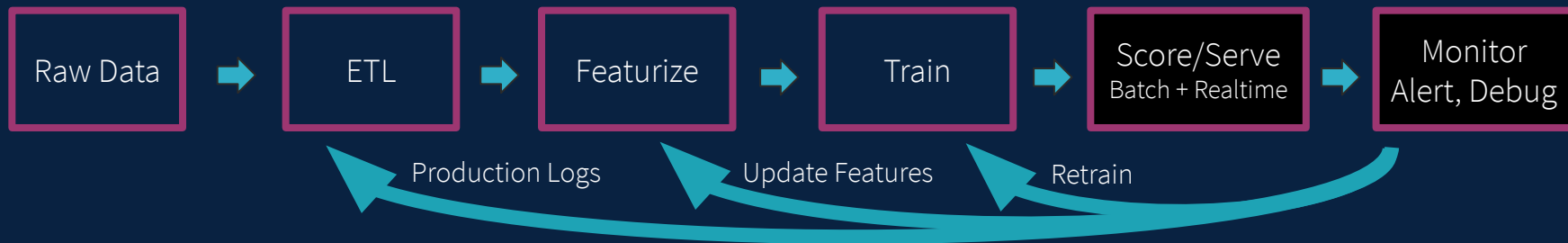


In summary, deploying ML Models is hard!

# ML Lifecycle and Challenges

**mlflow**

An open source platform for the machine learning lifecycle



Zoo of Ecosystem Frameworks

Tuning

Deploy

Model Mgmt

Collaboration

Scale

Governance

Feature Repository

Experiment Tracking

AutoML, Hyper-p. search

Remote Cloud Execution

Project Mgmt (scale teams)

Model Exchange

A/B Testing

CI/CD/Jenkins push to prod

Orchestration (Airflow, Jobs)

Lifecycle mgmt.

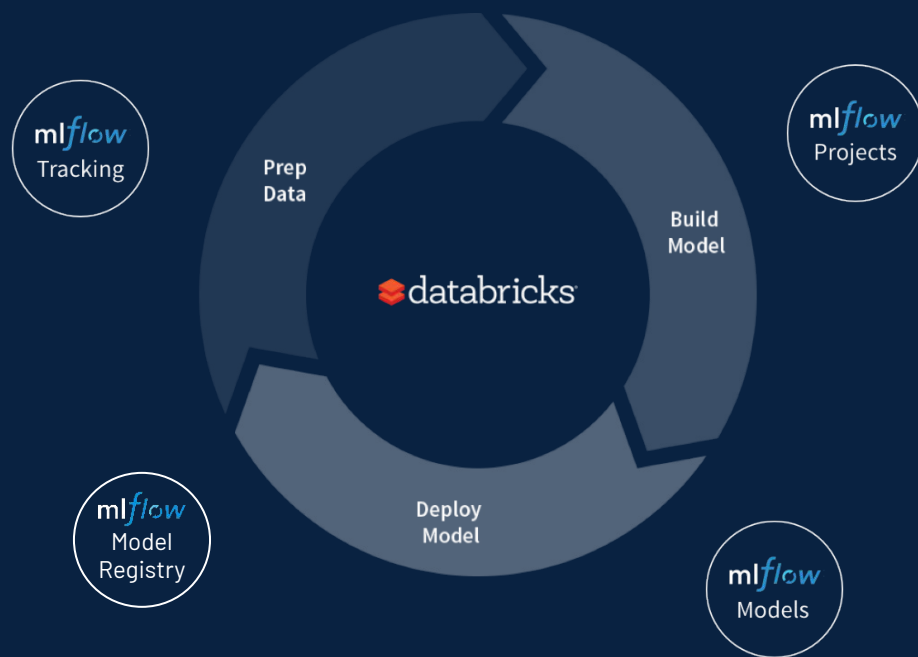
Data Drift

Model Drift

**mlflow**

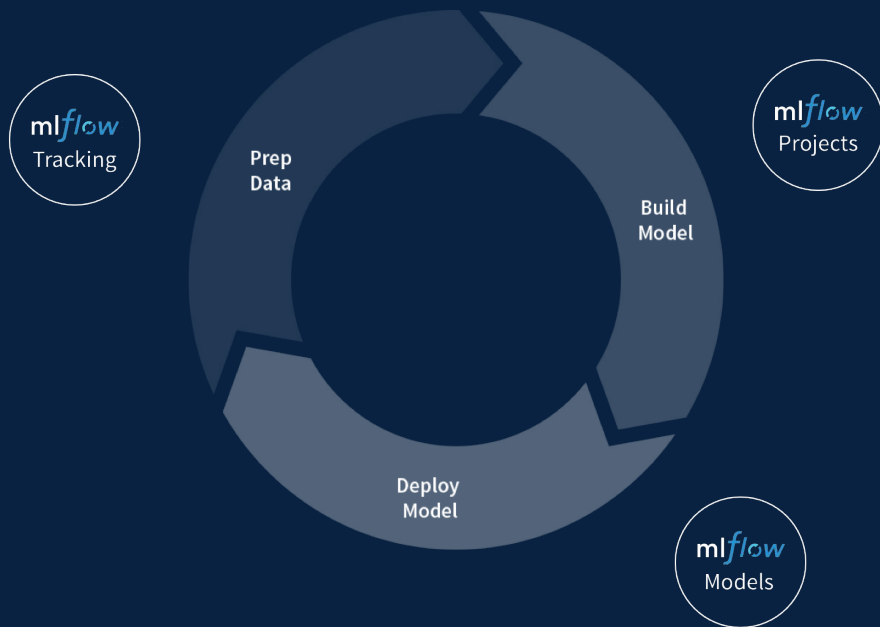
# Introducing MLflow

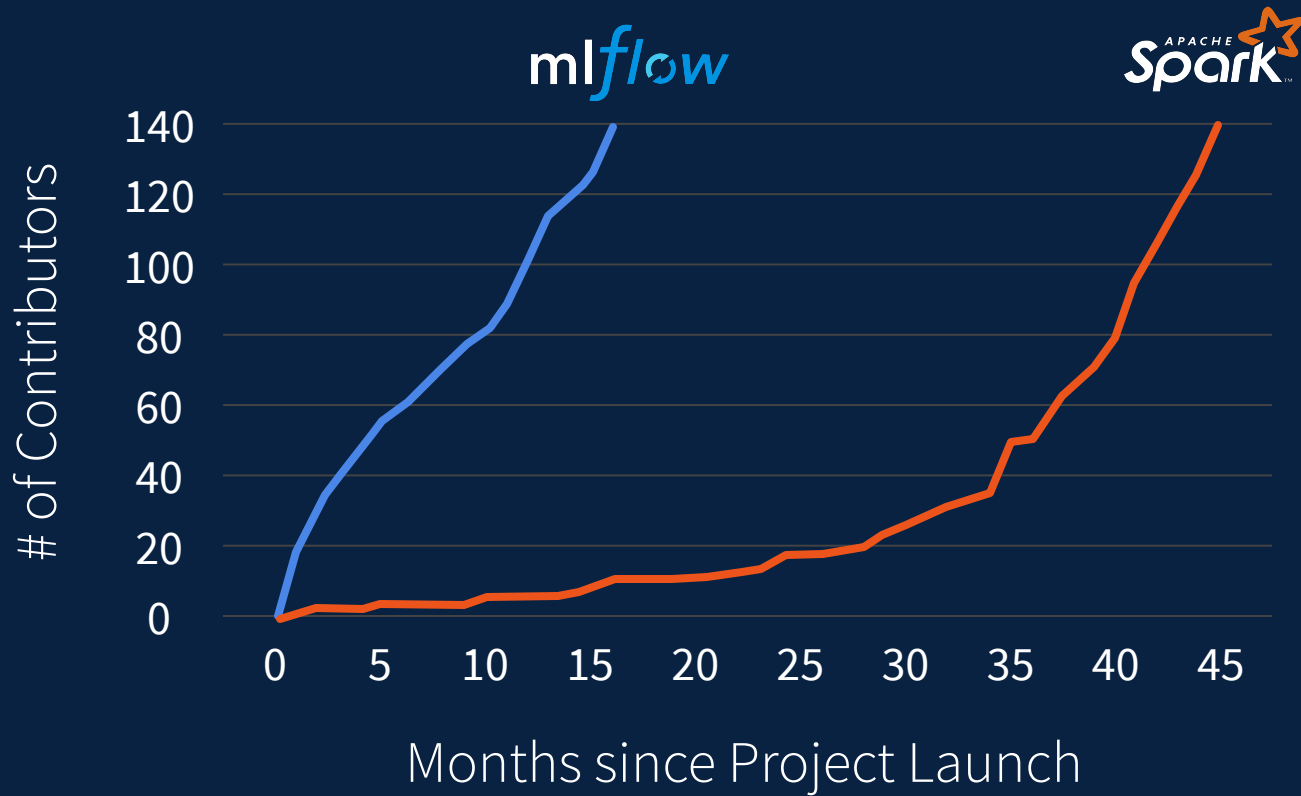
Unveiled in June 2018, MLflow is the only open source framework designed to manage the complete Machine Learning Lifecycle.



# Introducing MLflow

Unveiled in June 2018, MLflow is the only open source framework designed to manage the complete Machine Learning Lifecycle.





# mlflow Components

## mlflow Tracking

Record and query experiments: code, data, config, results

## mlflow Projects

Packaging format for reproducible runs on any platform

## mlflow Models

General format that standardizes deployment paths

## mlflow Model Registry

Centralized and collaborative model lifecycle management



# mlflow Components

## mlflow Tracking

Record and query experiments: code, data, config, results

## mlflow Projects

Packaging format for reproducible runs on any platform

## mlflow Models

General format that standardizes deployment paths

new

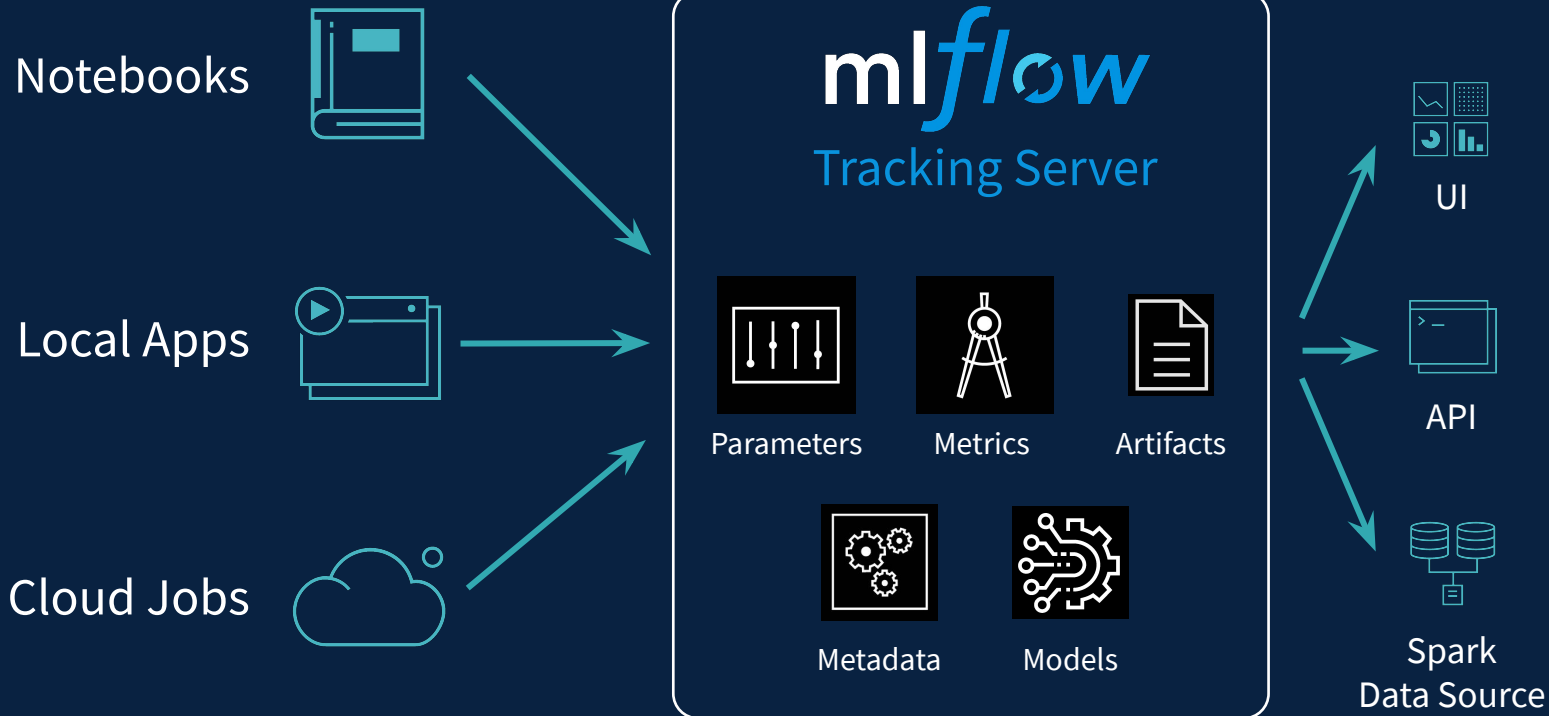
## mlflow Model Registry

Centralized and collaborative model lifecycle management





# mlflow Tracking



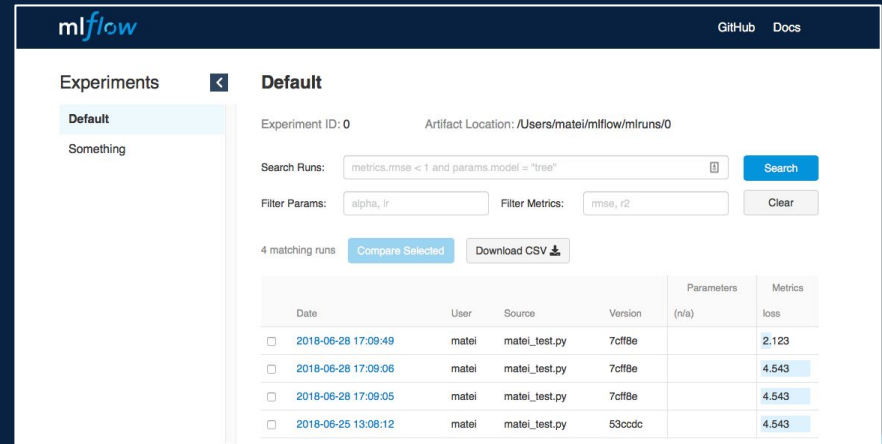
# Key Concepts in Tracking

**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 displays the mlflow web interface for an experiment. The top navigation bar includes the mlflow logo, GitHub, and Docs links. The main content area is titled "Experiments" and shows a sidebar with "Default" and "Something" options. The "Default" experiment is selected, showing "Experiment ID: 0" and "Artifact Location: /Users/matei/mlflow/mlruns/0". Search filters are applied: "Search Runs: metrics.rmse < 1 and params.model = 'tree'", "Filter Params: alpha, lr", and "Filter Metrics: rmse, r2". There are 4 matching runs, with buttons for "Compare Selected" and "Download CSV". A table lists the runs with columns for Date, User, Source, Version, Parameters (n/a), and Metrics (loss).

Date	User	Source	Version	Parameters (n/a)	Metrics (loss)
2018-06-28 17:09:49	matei	matei_test.py	7cff8e		2.123
2018-06-28 17:09:06	matei	matei_test.py	7cff8e		4.543
2018-06-28 17:09:05	matei	matei_test.py	7cff8e		4.543
2018-06-25 13:08:12	matei	matei_test.py	53ccdc		4.543

```
# Scikit Learn Linear Regression via ElasticNet
lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)
lr.fit(train_x, train_y)

# Predict
predicted_qualities = lr.predict(test_x)

# Evaluate Metrics
(rmse, mae, r2) = eval_metrics(test_y, predicted_qualities)
```

```
with mlflow.start_run() as run:

    # Scikit Learn Linear Regression via ElasticNet
    lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)
    lr.fit(train_x, train_y)

    # Predict
    predicted_qualities = lr.predict(test_x)

    # Evaluate Metrics
    (rmse, mae, r2) = eval_metrics(test_y, predicted_qualities)

    # Log
    mlflow.log_param("alpha", alpha)
    ...
```

# GitHub Demo

<https://github.com/dennyglee/mlflow-diabetes-example>

# Comparing Runs Contour Plot

Scatter Plot

**Contour Plot**

Parallel Coordinates Plot

X-axis:

l1\_ratio

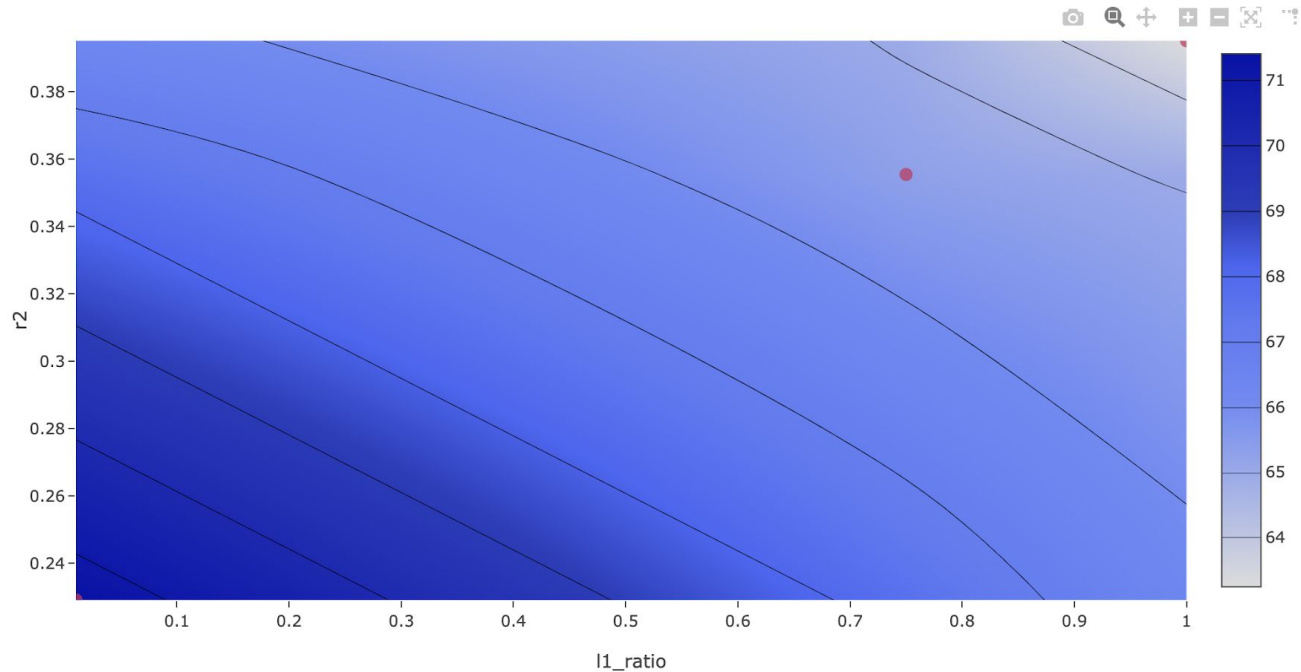
Y-axis:

r2

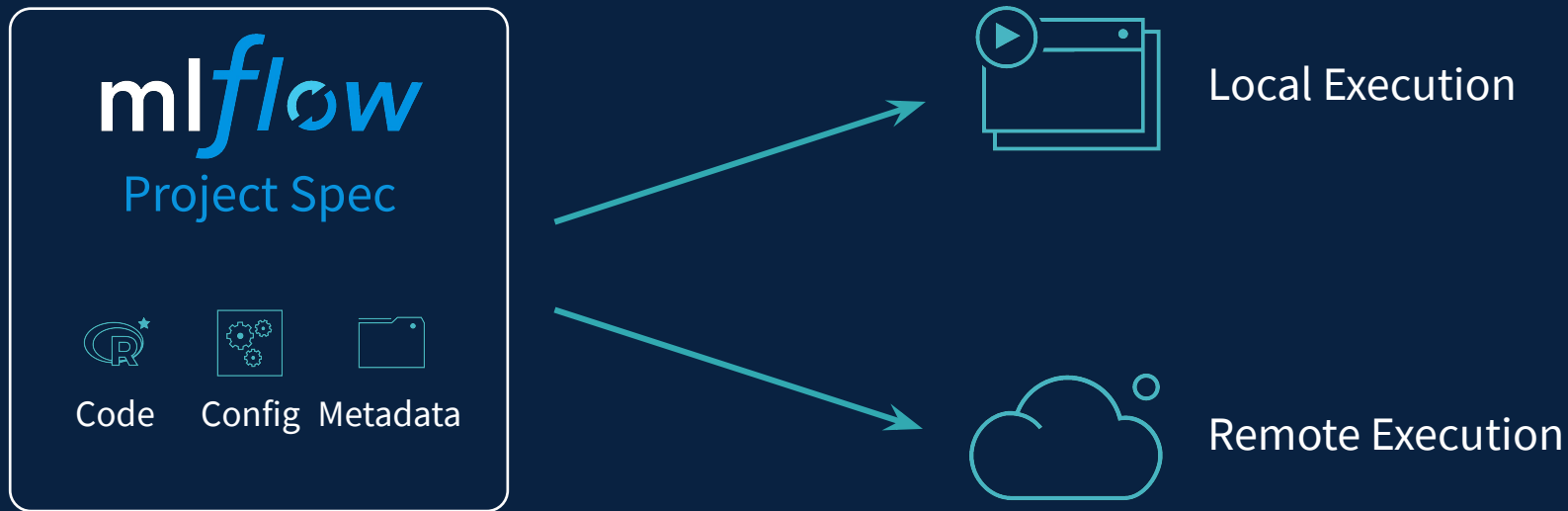
Z-axis:

rmse

Reverse color:  Off



# mlflow Projects



# Example MLflow Project

my\_project/  
├── MLproject

```
conda_env: conda.yaml
```

```
entry_points:
```

```
  main:
```

```
    parameters:
```

```
      training_data: path
```

```
      lambda: {type: float, default: 0.1}
```

```
    command: python main.py {training_data} {lambda}
```

├── conda.yaml

├── main.py

├── model.py

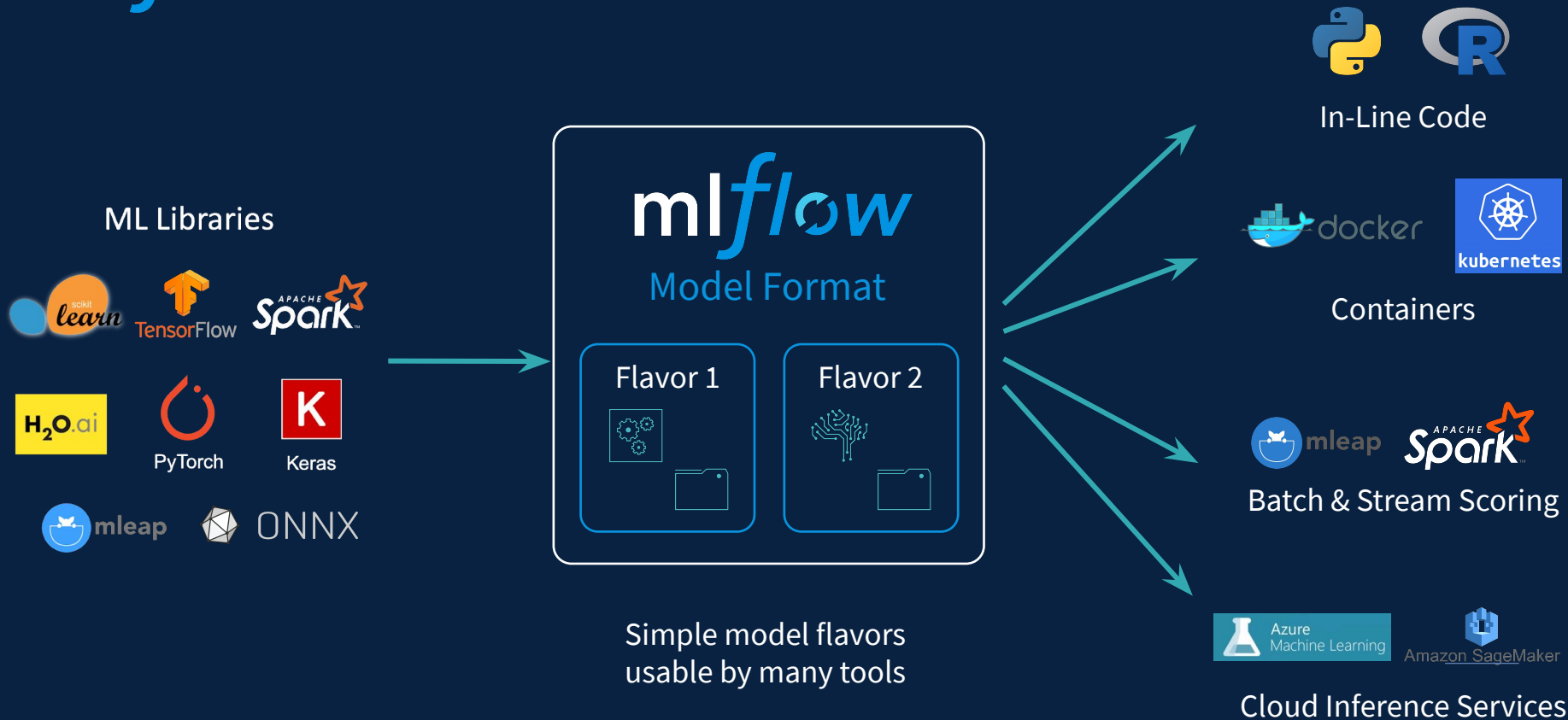
...

```
$ mlflow run git://<my_project>
```

```
mlflow.run("git://<my_project>", ...)
```



# mlflow Models



# 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
```

} Usable by tools that understand TensorFlow model format

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

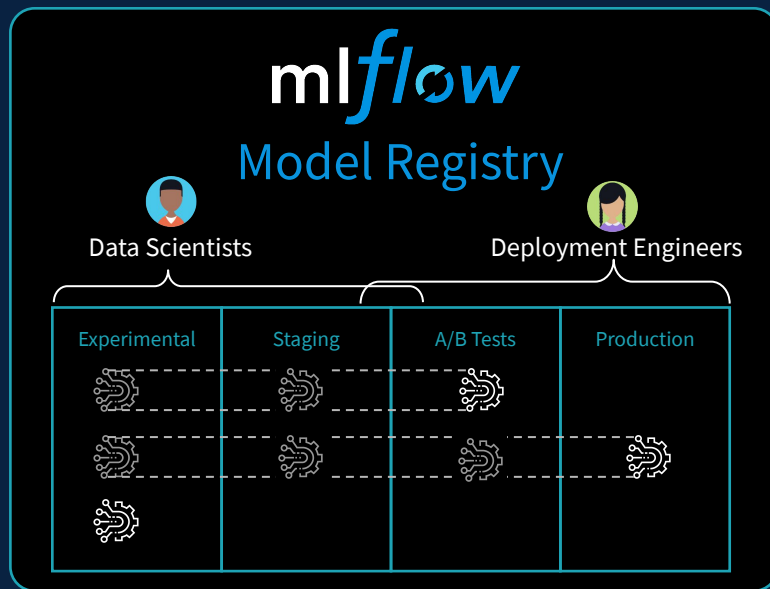
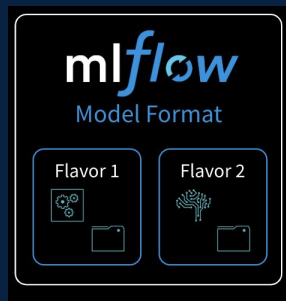
estimator/

saved\_model.pb

variables/

...

# mlflow Model Registry



Reviewers + CI/CD Tools



Downstream Users



Automated Jobs



REST Serving

# mflow Model Registry: Benefits

The screenshot displays the 'Registered Models' page in the mflow interface. The left sidebar contains navigation options: Home, Workspace, Projects, Recents, Data, Clusters, and Models (highlighted with callout 1). The main content area shows a table of models with columns for Name, Latest Version, Staging, Production, and Last Modified. Callout 2 highlights a row for 'Airline\_Delay\_SparkML' showing its latest version (11) and previous versions (6 and 5) in Staging and Production environments. Callout 3 highlights the search bar at the top right and the pagination controls at the bottom right.

Name	Latest Version	Staging	Production	Last Modified
AaronModel	Version 108	–	–	2019-10-18 09:04:20
Airline_Delay_Scikit	Version 3	–	Version 1	2019-10-11 12:41:43
Airline_Delay_SparkML	Version 11	Version 6	Version 5	2019-10-22 10:30:21
Andre_BasicModels_02_Sklearn_Train_Predict	Version 2	Version 1	–	2019-10-19 14:38:11
BertsLarge	Version 1	–	–	2019-10-11 15:18:05
Brooke Keras Model	Version 1	–	–	2019-10-12 08:20:12
holland-forecast-model	Version 1	–	Version 1	2019-10-07 15:38:27
joytesting	–	–	–	2019-10-15 18:23:17
ManiErrorModel	–	–	–	2019-10-14 16:53:10
MatelModel	Version 5	Version 5	Version 3	9-10-10 14:07:07

## One Collaborative Hub

- 1 Central Model Repository
- 2 Overview of versions in Staging/Production/etc.
- 3 Search/filter/pagination

# mlflow Model Registry: Benefits

Registered Models > Airline\_Delay\_SparkML

Created Time: 2019-10-10 15:20:29 Last Modified: 2019-10-22 17:08:29

Description

Predicts airline delays (in minutes) using the best Spark RF model from the AutoML Toolkit.

Versions All Active(2)

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

Stage: Staging

Request transition to → None

Request transition to → Production

Request transition to → Archived

Transition to → None

Transition to → Production

Transition to → Archived

## Management of the entire ML Lifecycle (MLOps)

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

# mlflow Model Registry: Benefits

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.

▸ Pending Requests

▾ Activities

- 1 ✓ 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

## Visibility

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

# mlflow Model Registry: Benefits

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

1.a

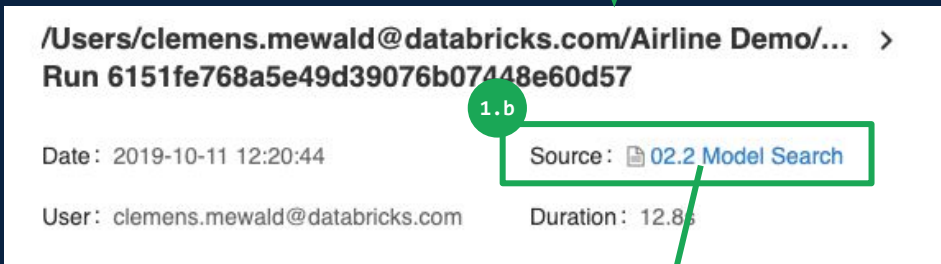


/Users/clemens.mewald@databricks.com/Airline Demo/... >  
Run 6151fe768a5e49d39076b07448e60d57

Date: 2019-10-11 12:20:44 Source: 02.2 Model Search

User: clemens.mewald@databricks.com Duration: 12.8s

1.b



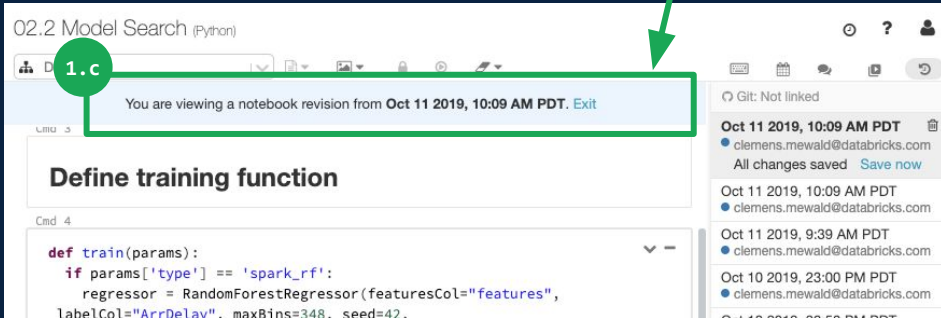
02.2 Model Search (Python)

You are viewing a notebook revision from Oct 11 2019, 10:09 AM PDT. Exit

### Define training function

```
Cmd 4
def train(params):
    if params['type'] == 'spark_rf':
        regressor = RandomForestRegressor(featuresCol="features",
            labelCol="ArrDelay", maxBins=348, seed=42,
```

1.c



## Governance and Auditability

- Full provenance from Model marked production in the Registry to ...

- - 1.a Run that produced the model
  - 1.b Notebook that produced the run
  - 1.c Exact revision history of the notebook that produced the run

# Notebook Demo

[https://github.com/dennyglee/tech-talks/blob/master/samples/MLflow%20Diabetes%20Example%20\(with%20MLflow%20Registry\).ipynb](https://github.com/dennyglee/tech-talks/blob/master/samples/MLflow%20Diabetes%20Example%20(with%20MLflow%20Registry).ipynb)



# *mlflow*: An Open Source ML Platform

Towards more principled  
Data Science and ML

[mlflow.org](https://mlflow.org)



[github.com/mlflow](https://github.com/mlflow)



[twitter.com/MLflow](https://twitter.com/MLflow)

*mlflow*

# Hands-on Workshop

[bit.ly/mlflow-boss-2020](https://bit.ly/mlflow-boss-2020)