

Overview of Lakehouse AI

DATABRICKS CONCEPTS



Kevin Barlow
Data Practitioner

Lakehouse AI

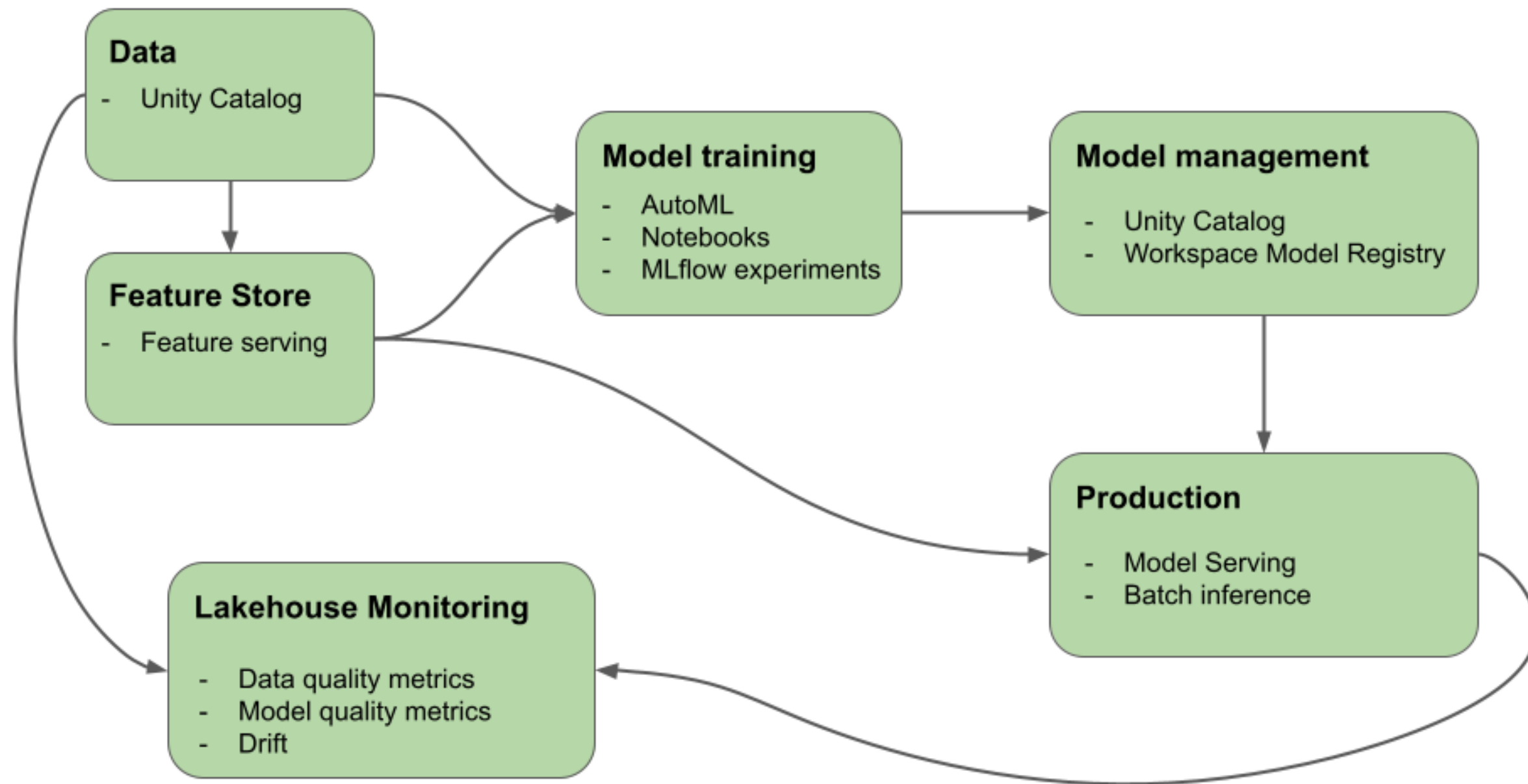


Why the Lakehouse for AI / ML?

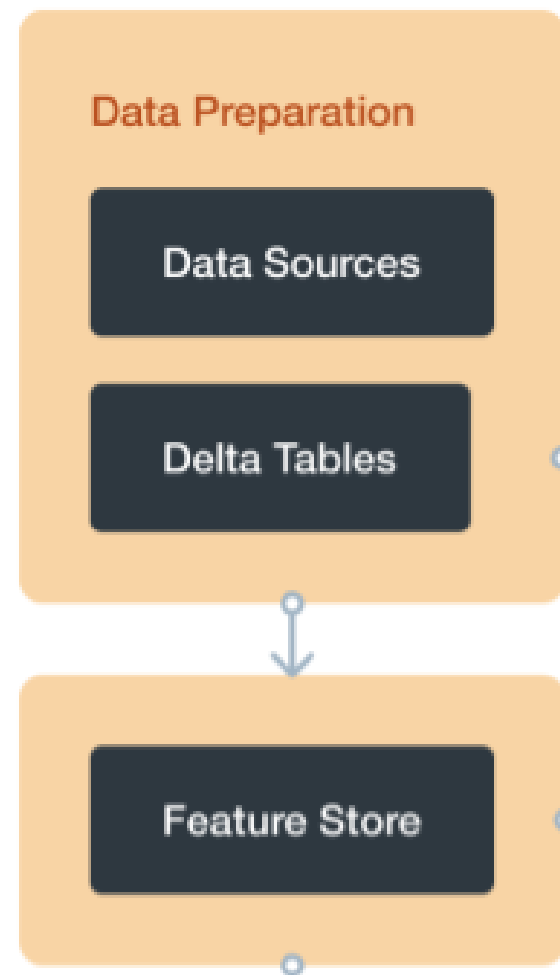
1. Reliable data and files in the Delta lake
2. Highly scalable compute
3. Open standards, libraries, frameworks
4. Unification with other data teams

¹ <https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html>

MLOps Lifecycle



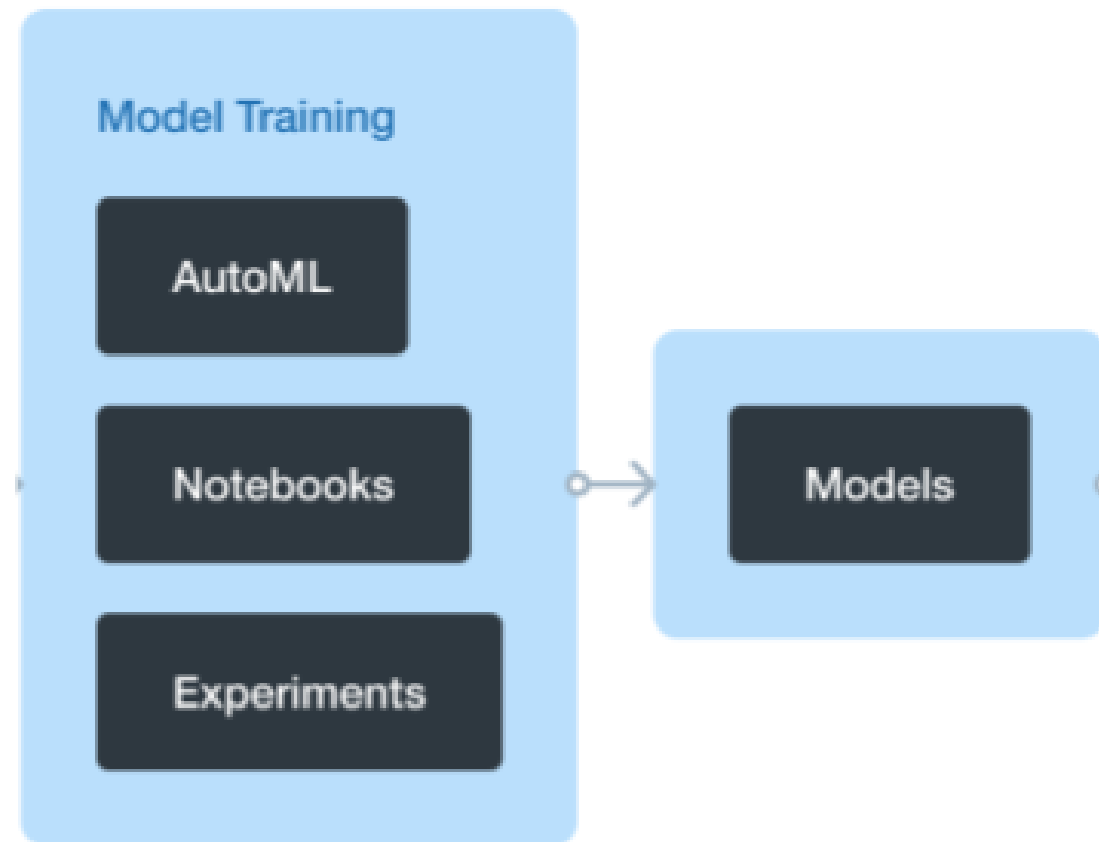
MLOps in the Lakehouse



DataOps

- Integrating data across different sources (*AutoLoader*)
- Transforming data into a usable, clean format (*Delta Live Tables*)
- Creating useful features for models (*Feature Store*)

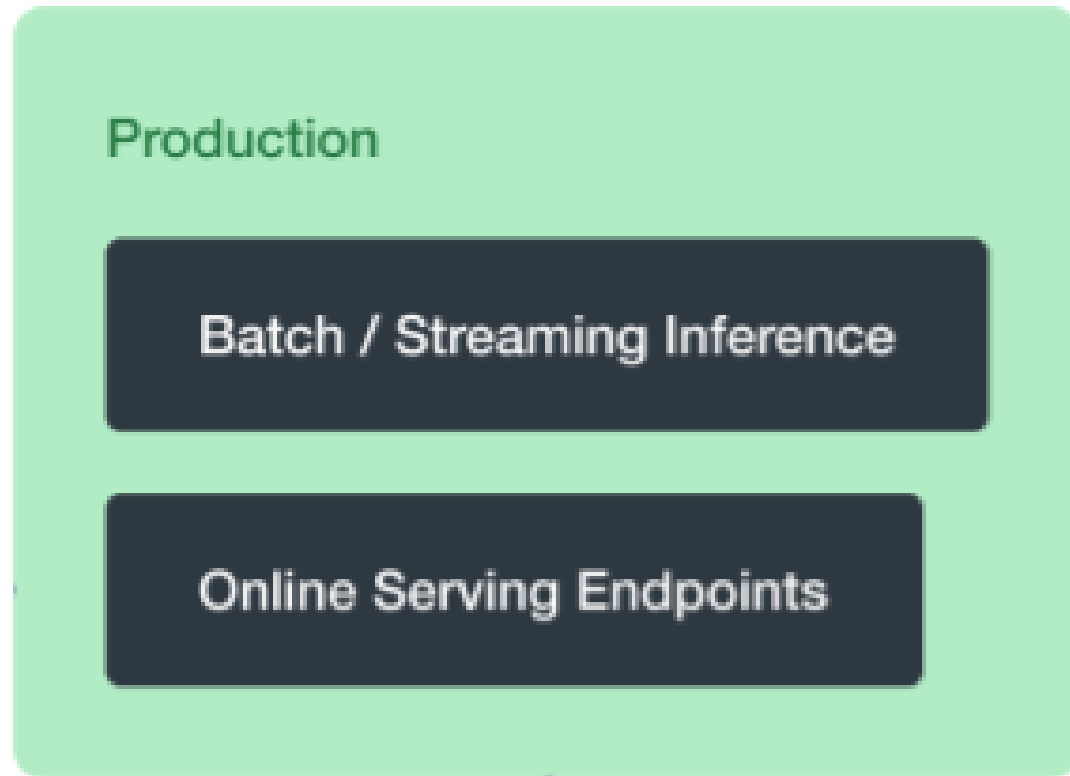
MLOps in the Lakehouse



ModelOps

- Develop and train different models (*Notebooks*)
- Machine learning templates and automation (*AutoML*)
- Track parameters, metrics, and trials (*MLFlow*)
- Centralize and consume models (*Model Registry*)

MLOps in the Lakehouse



DevOps

- Govern access to different models (*Unity Catalog*)
- Continuous Integration and Continuous Deployment (CI/CD) for model versions (*Model Registry*)
- Deploy models for consumption (*Serving Endpoints*)

Let's review!

DATABRICKS CONCEPTS

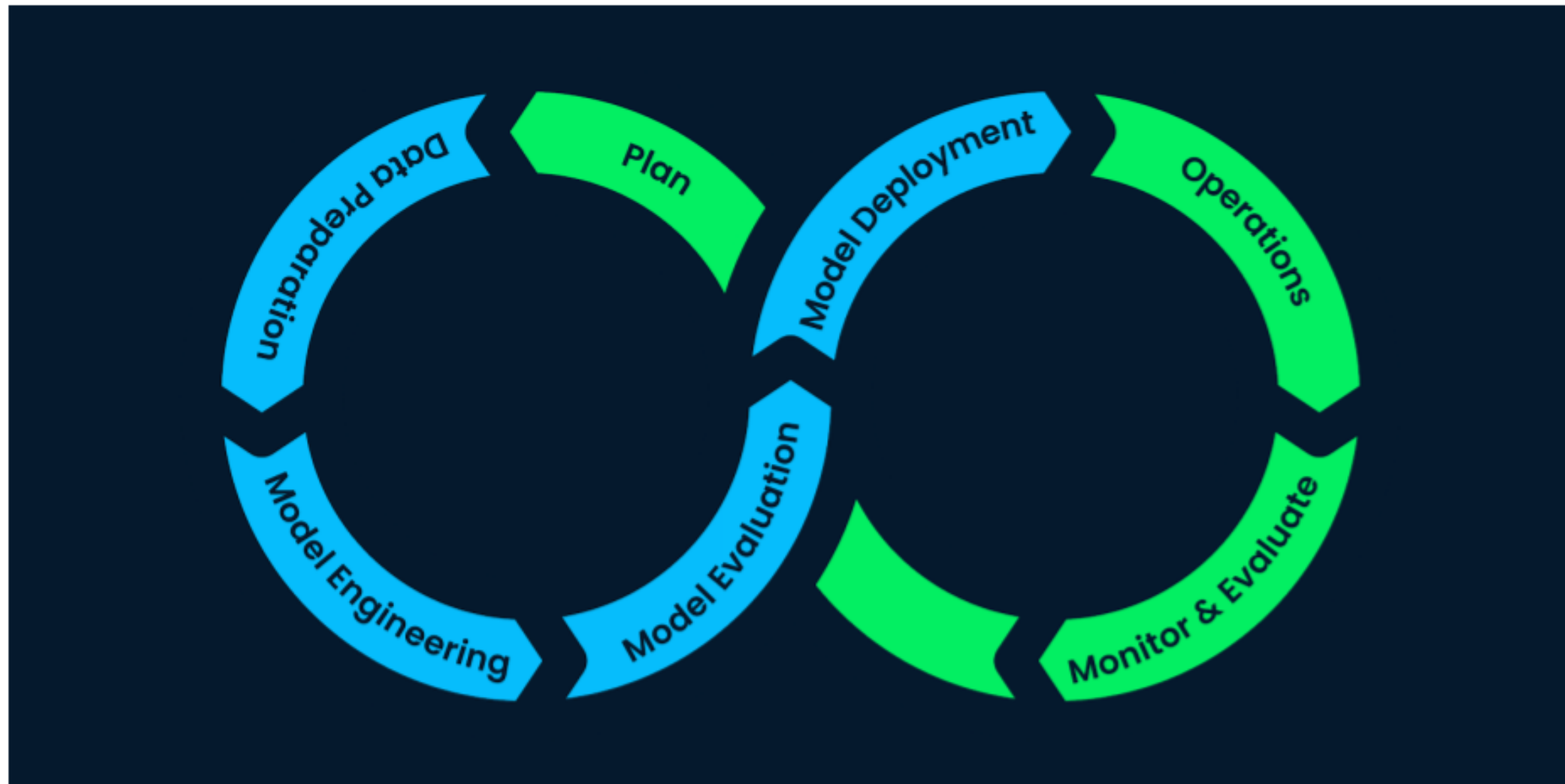
Using Databricks for machine learning

DATABRICKS CONCEPTS



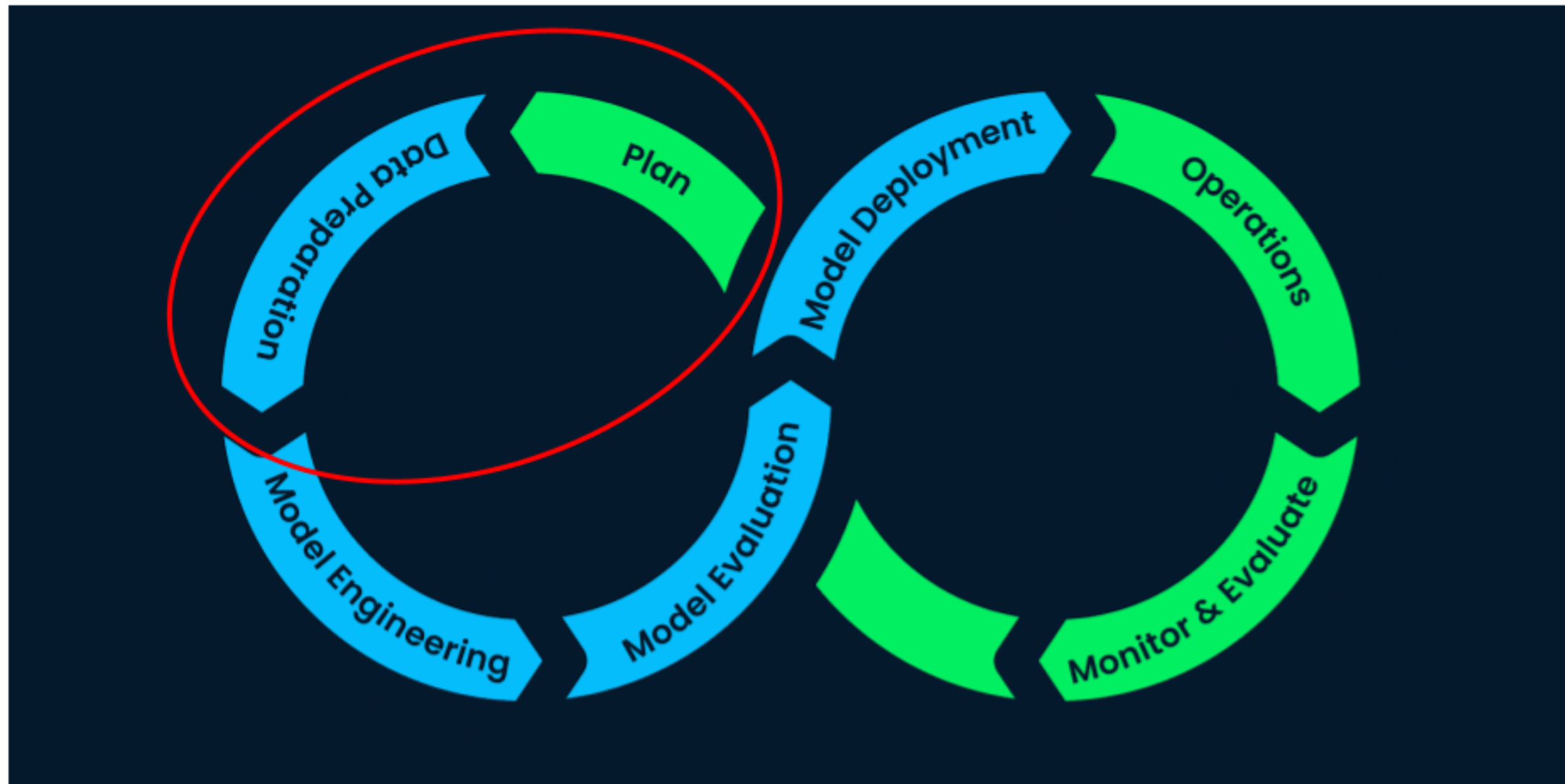
Kevin Barlow
Data Practitioner

Machine Learning Lifecycle



¹ <https://www.datacamp.com/blog/machine-learning-lifecycle-explained>

Planning and preparation



Planning for machine learning

What do I have?

1. Data availability
2. Business requirements
3. Data scientists/data analysts



What do I want?

1. Use cases
2. Legal and security compliance
3. Business outcomes



ML Runtime

- Extension of Databricks compute
- Optimized for machine learning applications
- Contains most common libraries and frameworks
 - `scikit-learn` , `SparkML` , `TensorFlow`
 - `MLFlow`
- Works with cluster library management

Databricks runtime version ?

| | | |
|---|--------------|------------------------------|
| Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1) ▼ | | |
| Standard > | 14.0 ML Beta | includes GPU, Scala 2.12 |
| ML > | 14.0 ML Beta | includes Scala 2.12 |
| Uncategorized > | 13.3 LTS ML | GPU, Scala 2.12, Spark 3.4.1 |
| | 13.3 LTS ML | Scala 2.12, Spark 3.4.1 |
| | 13.2 ML | GPU, Scala 2.12, Spark 3.4.0 |
| | 13.2 ML | Scala 2.12, Spark 3.4.0 |
| | 13.1 ML | GPU, Scala 2.12, Spark 3.4.0 |
| | 13.1 ML | Scala 2.12, Spark 3.4.0 |
| | 13.0 ML | GPU, Scala 2.12, Spark 3.4.0 |
| | 13.0 ML | Scala 2.12, Spark 3.4.0 |
| | 12.2 LTS ML | GPU, Scala 2.12, Spark 3.3.2 |
| | 12.2 LTS ML | Scala 2.12, Spark 3.3.2 |

Exploratory Data Analysis

```
import pandas as pd  
pd.describe(df)
```

```
# Spark DF  
df.summary()
```

```
dbutils.data.summarize()
```

```
import bamboolib as bam  
df
```

The screenshot shows the Databricks Exploratory Analysis interface. On the left, a sidebar contains icons for various actions: Get latest data, Transform with bamboolib, Visualize, Save to Delta Lake, and View table. The main area displays a Python script in a code editor:

```
1 import bamboolib as bam  
2  
3 # This opens a UI from which you can import your data  
4 df
```

Below the code editor, a button labeled "Show bamboolib UI" is visible. Underneath the button is a table of data:

| | entity | iso_code | date | indicator | value |
|--------|---------------|----------|------------|--|-----------|
| 0 | Algeria | DZA | 2020-07-17 | Daily ICU occupancy | 62.000 |
| 1 | Algeria | DZA | 2020-07-17 | Daily ICU occupancy per million | 1.403 |
| 2 | Algeria | DZA | 2020-07-18 | Daily ICU occupancy | 67.000 |
| 3 | Algeria | DZA | 2020-07-18 | Daily ICU occupancy per million | 1.517 |
| 4 | Algeria | DZA | 2020-07-20 | Daily ICU occupancy | 64.000 |
| ... | ... | ... | ... | ... | ... |
| 160361 | United States | USA | 2022-09-26 | Daily ICU occupancy per million | 8.588 |
| 160362 | United States | USA | 2022-09-26 | Daily hospital occupancy | 23659.000 |
| 160363 | United States | USA | 2022-09-26 | Daily hospital occupancy per million | 70.205 |
| 160364 | United States | USA | 2022-09-26 | Weekly new hospital admissions | 27246.000 |
| 160365 | United States | USA | 2022-09-26 | Weekly new hospital admissions per million | 80.849 |

Below the table, it states "160366 rows x 5 columns". At the bottom, a message indicates the command took 0.46 seconds and was executed by rafi.kurlanskik@databricks.com at 9/29/2022, 11:58:56 AM on storm.

Below the table, there is a "Visualize" section with the following text:

Now that we have a tidy dataset we can visualize the trends over time. Let's use bamboolib to create a plot that we will ultimately include in our weekly report.

In the following cell, enter the name of your pandas dataframe that you created in the previous step and run the cell to launch bamboolib again.

To Do:

1. Using the Plot Creator in bamboolib, **create a Line plot**.

Feature tables and feature stores

Raw Data

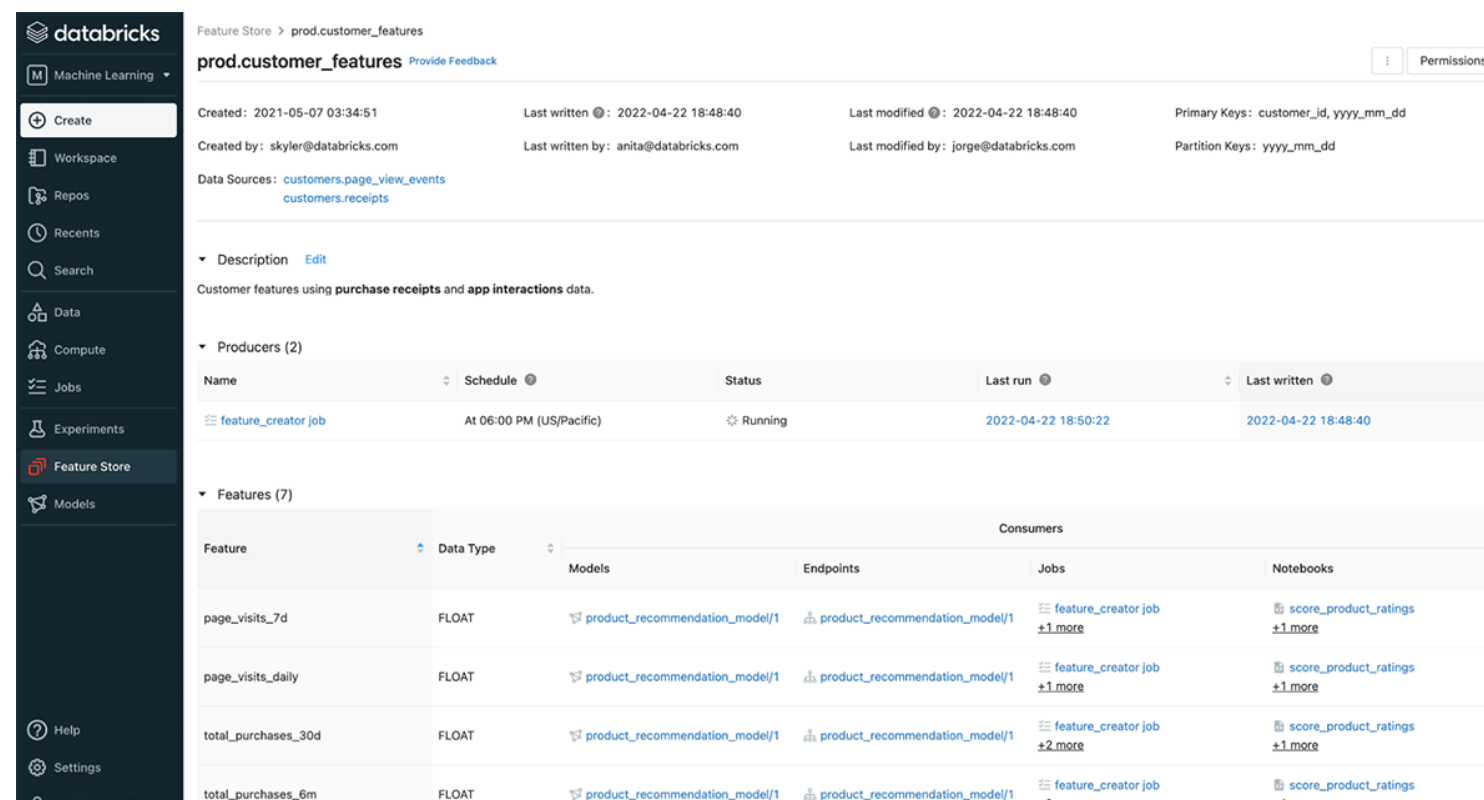
| count | category | price | shelf_loc | rating |
|-------|----------|-------|-----------|--------|
| 4 | horror | 12.50 | end | 3 |
| 6 | romance | 13.99 | top | 4.5 |
| 12 | sci-fi | 16.50 | bottom | 5 |
| 31 | romance | 9.99 | bottom | 3.5 |
| 23 | fantasy | 24.99 | top | 4 |
| 18 | horror | 19.99 | end | 2.5 |
| 19 | cooking | 17.50 | end | 4.5 |
| 7 | fantasy | 12.99 | top | 3 |
| 37 | sci-fi | 14.99 | bottom | 5 |

Feature table

| count | category | price | shelf_loc | rating |
|-------|----------|-------|-----------|--------|
| 4 | 1 | 12.50 | 1 | 3 |
| 6 | 2 | 13.99 | 2 | 4.5 |
| 12 | 3 | 16.50 | 3 | 5 |
| 31 | 2 | 9.99 | 3 | 3.5 |
| 23 | 4 | 24.99 | 2 | 4 |
| 18 | 1 | 19.99 | 1 | 2.5 |
| 19 | 5 | 17.50 | 1 | 4.5 |
| 7 | 4 | 12.99 | 2 | 3 |
| 37 | 3 | 14.99 | 3 | 5 |

Databricks Feature Store

- Centralized storage for featurized datasets
- Easily discover and re-use features for machine learning models
- Upstream and downstream lineage



The screenshot displays the Databricks Feature Store interface for a table named 'prod.customer_features'. The interface includes a sidebar with navigation options like 'Machine Learning', 'Workspace', 'Repos', 'Recents', 'Search', 'Data', 'Compute', 'Jobs', 'Experiments', 'Feature Store', and 'Models'. The main content area shows the table's metadata, including creation and modification dates, data sources, and a description. Below this, there are sections for 'Producers' and 'Features'. The 'Features' section lists seven features with their data types and consumers.

| Feature | Data Type | Consumers |
|---------------------|-----------|--|
| page_visits_7d | FLOAT | product_recommendation_model/1, feature_creator job, score_product_ratings |
| page_visits_daily | FLOAT | product_recommendation_model/1, feature_creator job, score_product_ratings |
| total_purchases_30d | FLOAT | product_recommendation_model/1, feature_creator job, score_product_ratings |
| total_purchases_6m | FLOAT | product_recommendation_model/1, feature_creator job, score_product_ratings |

```
from databricks import feature_store

fs = feature_store.FeatureStoreClient()

fs.create_table(
    name=table_name,
    primary_keys=["wine_id"],
    df=features_df,
    schema=features_df.schema,
    description="wine features"
)
```

Let's practice!

DATABRICKS CONCEPTS

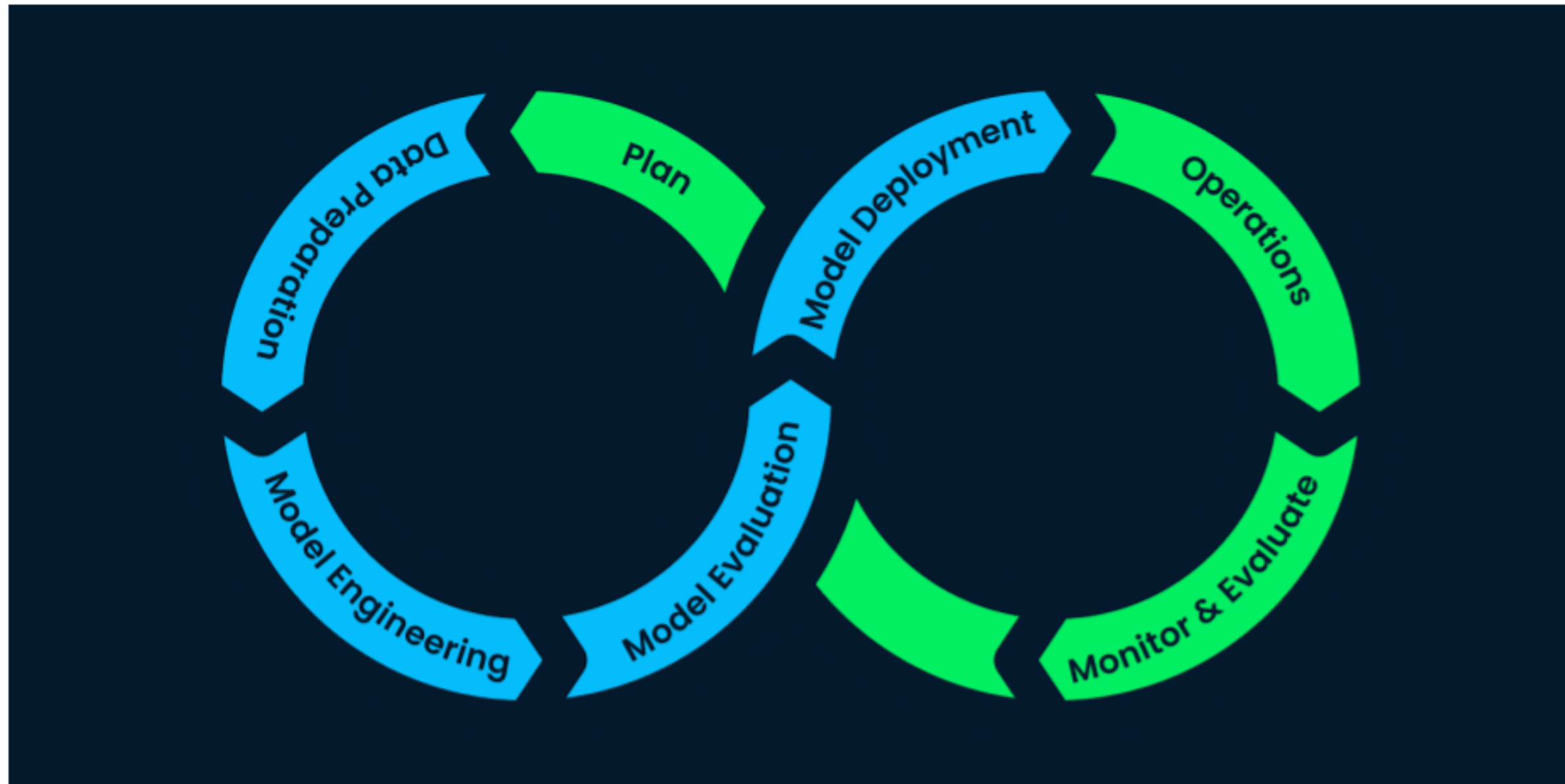
Model training with MLFlow in Databricks

DATABRICKS CONCEPTS



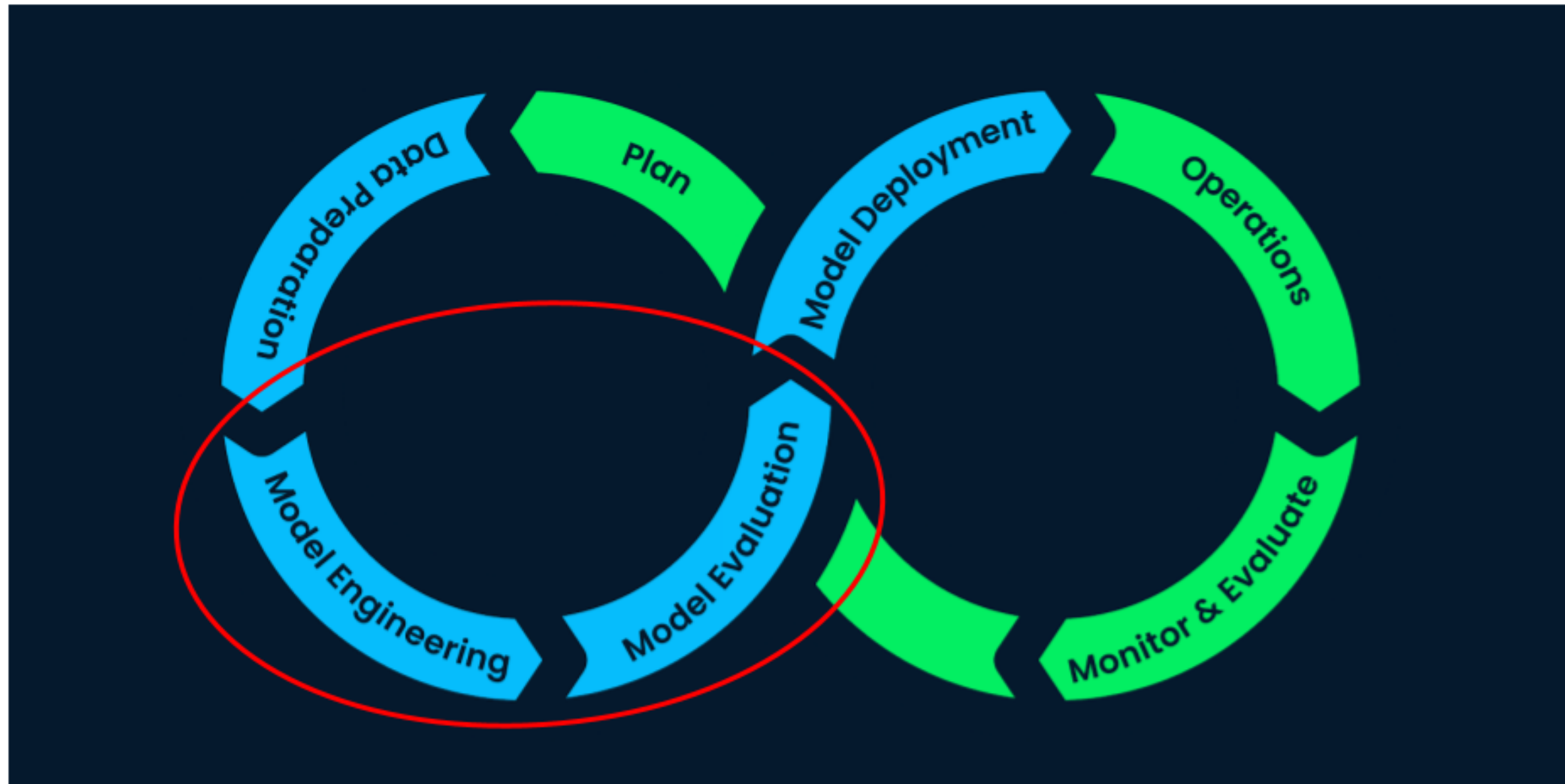
Kevin Barlow
Data Practitioner

Machine Learning Lifecycle



¹ <https://www.datacamp.com/blog/machine-learning-lifecycle-explained>

Model training and development



Single-node vs. Multi-node

Single-node machine learning

- Great for experimenting and starting
- Easier initial setup
- Hard to implement in production



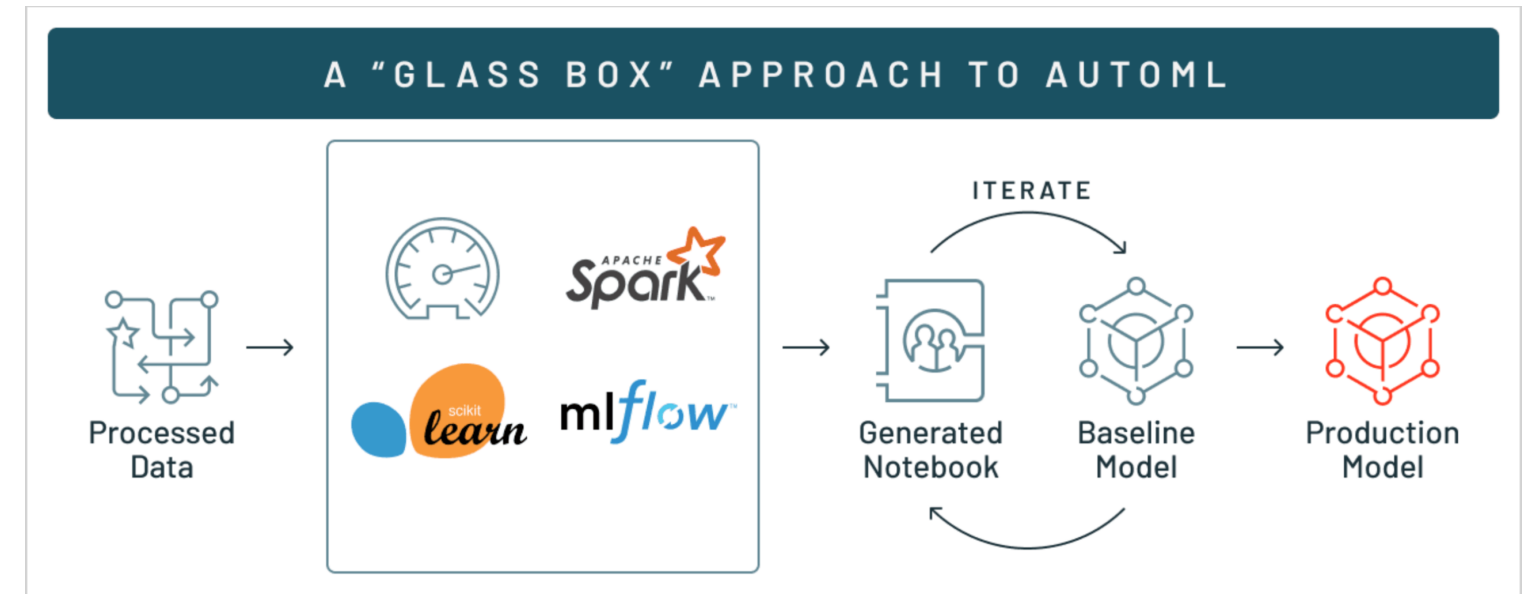
Multi-node machine learning

- Great for production workloads
- Easier maintenance long-term
- Highly scalable



AutoML

- "Glass box" approach to automated machine learning
- Leverages open-source libraries
- Creates models based on data and targeted prediction
- Provides notebook with generated code for further



¹ <https://www.databricks.com/product/automl>

MLFlow

- Open-source framework
- End-to-end machine learning lifecycle management
- Track, evaluate, manage, and deploy
- Pre-installed on ML Runtime!



```
import mlflow

with mlflow.start_run() as run:
    # machine learning training

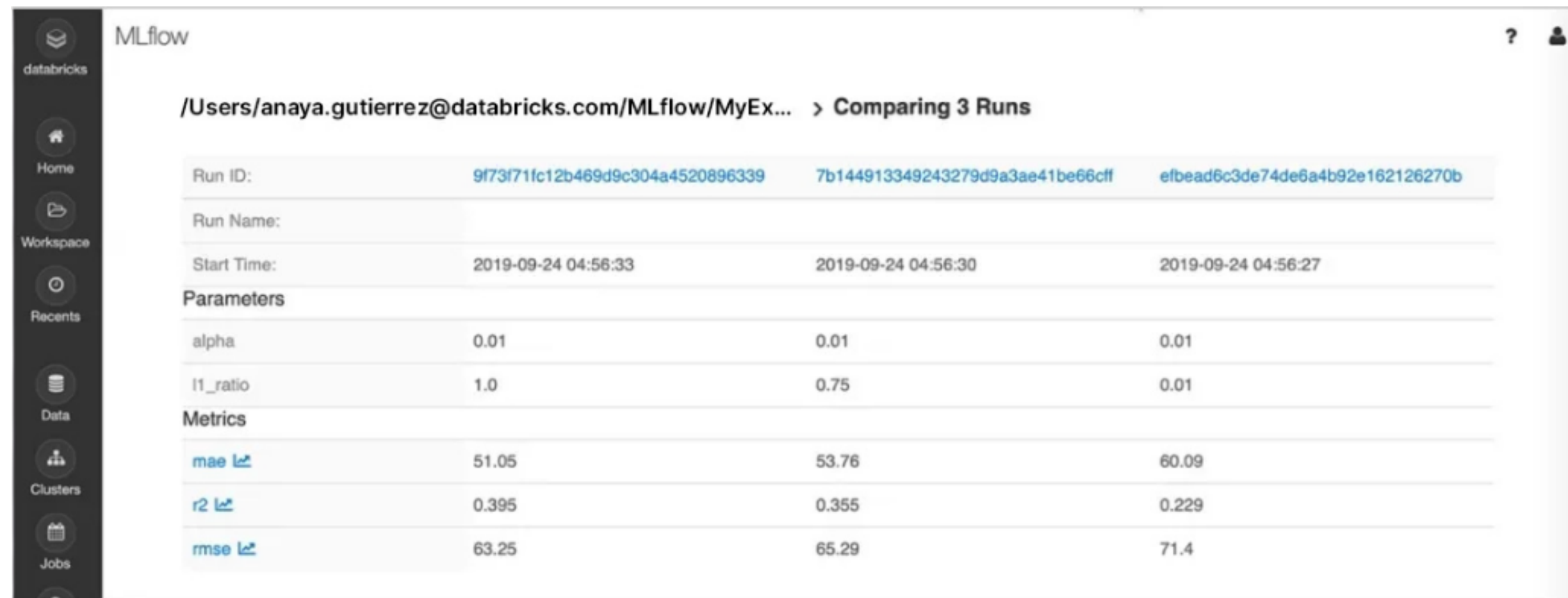
mlflow.autolog()

mlflow.log_metric('accuracy', acc)

mlflow.log_param('k', kNum)
```




MLFlow Experiments

- Collect information across multiple runs in a single location
- Sort and compare model runs
- Find and promote the best model



MLflow

/Users/anaya.gutierrez@databricks.com/MLflow/MyEx... > Comparing 3 Runs

| | | | |
|--|----------------------------------|----------------------------------|----------------------------------|
| Run ID: | 9f73f71fc12b469d9c304a4520896339 | 7b144913349243279d9a3ae41be66cff | efbead6c3de74de6a4b92e162126270b |
| Run Name: | | | |
| Start Time: | 2019-09-24 04:56:33 | 2019-09-24 04:56:30 | 2019-09-24 04:56:27 |
| Parameters | | | |
| alpha | 0.01 | 0.01 | 0.01 |
| l1_ratio | 1.0 | 0.75 | 0.01 |
| Metrics | | | |
| mae  | 51.05 | 53.76 | 60.09 |
| r2  | 0.395 | 0.355 | 0.229 |
| rmse  | 63.25 | 65.29 | 71.4 |

Let's practice!
DATABRICKS CONCEPTS

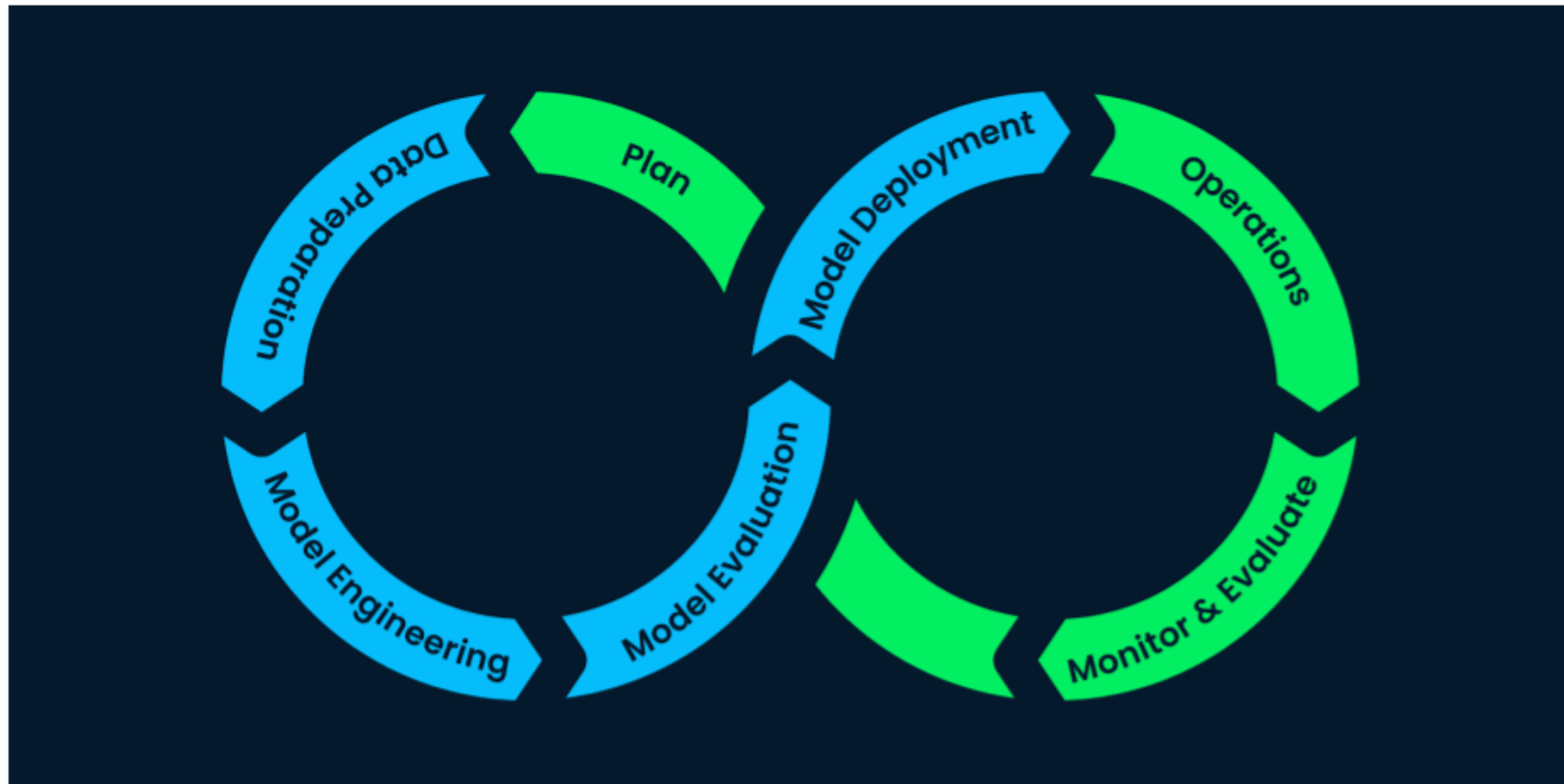
Deploying a model in Databricks

DATABRICKS CONCEPTS



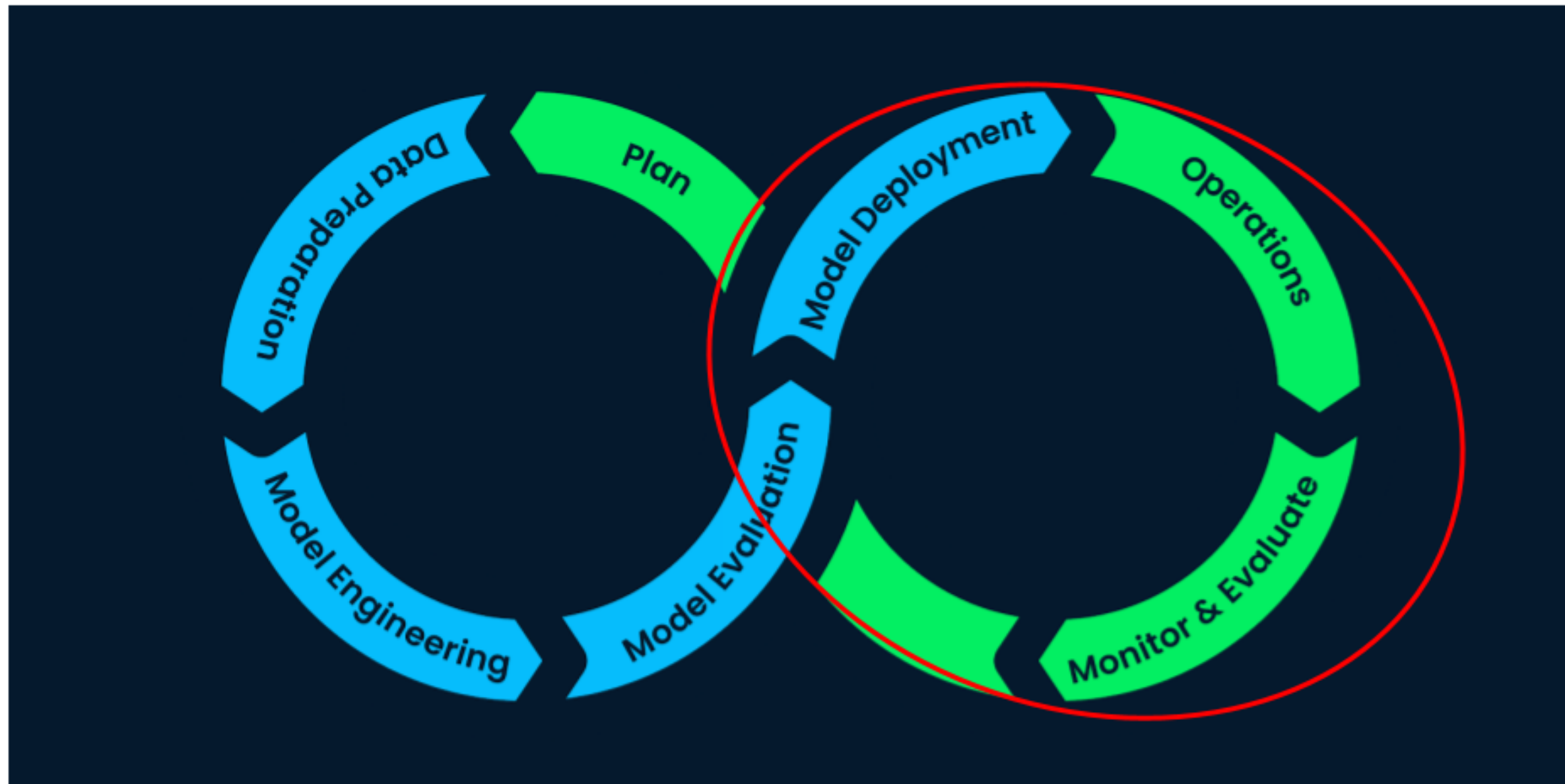
Kevin Barlow
Data Practitioner

Machine Learning Lifecycle



¹ <https://www.datacamp.com/blog/machine-learning-lifecycle-explained>

Model Deployment and Operations



Concerns with deploying models

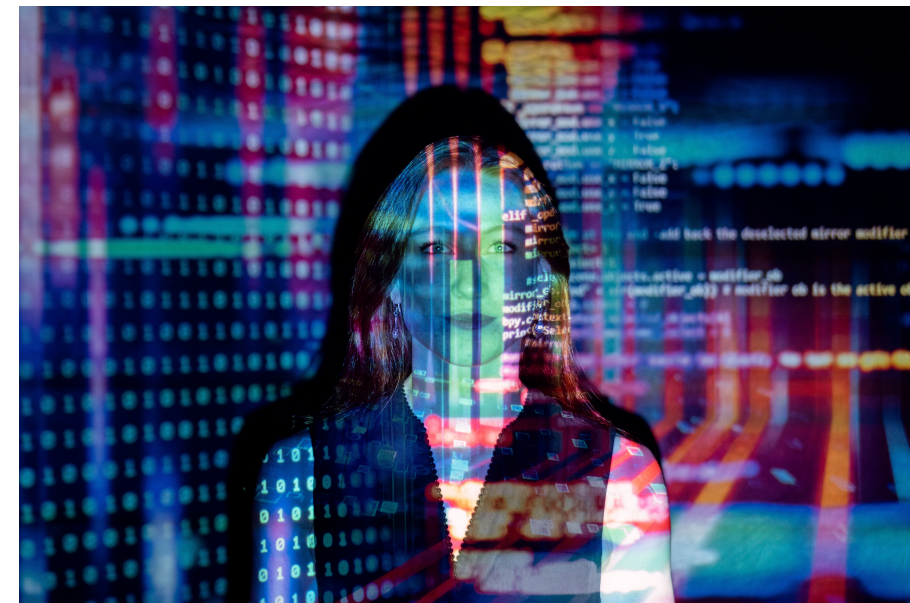
Availability

- How will my end users or application use the model?
- Where do I need to put my model to access it?
- Will the model be easy to understand or use?

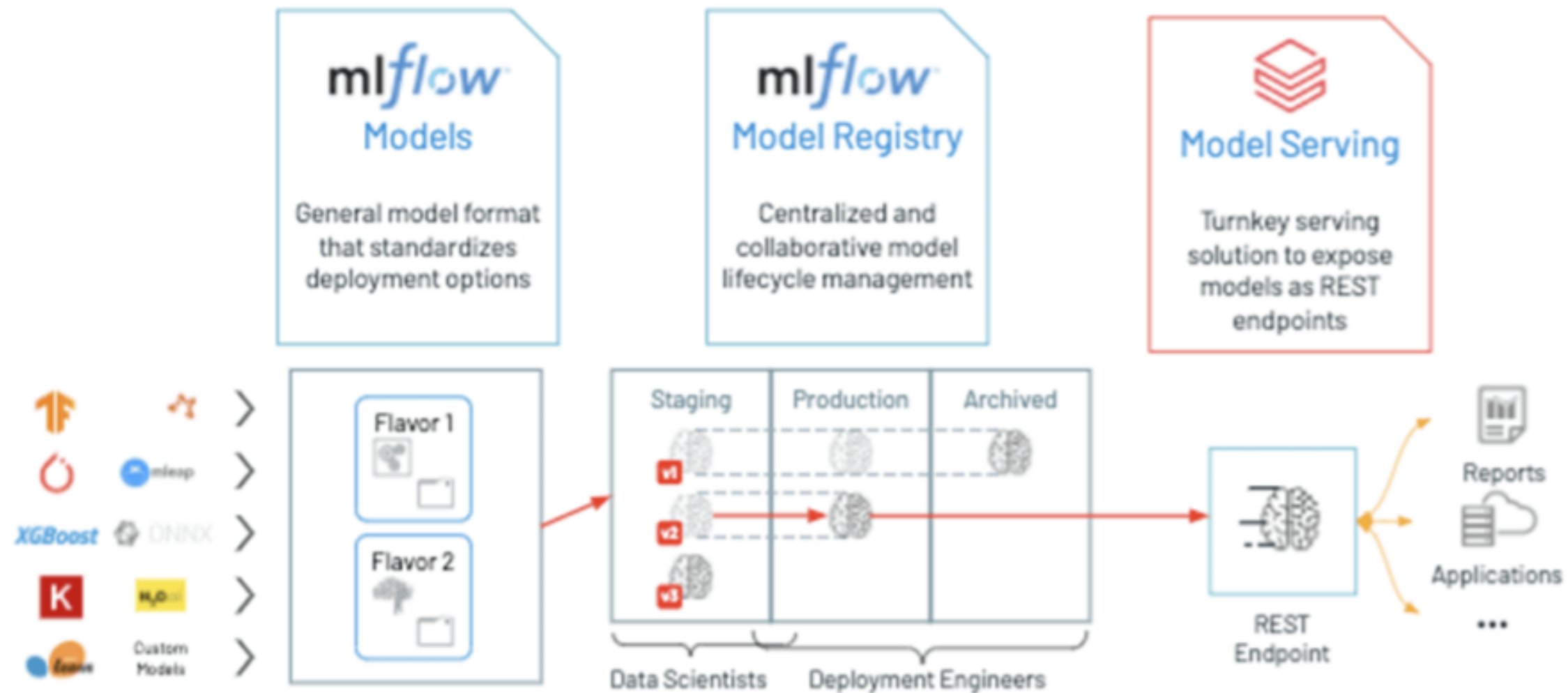


Evaluation

- Are my users *actually* using my model?
- Is my model still performing well?
- Do I need to retrain my model?
- Do I need a new model that is better?



Model Deployment Process



Model Flavors

- MLFlow Models can store a model from any machine learning framework
- Models are stored alongside different configurations and artifacts
- Models can be "translated" into another kind of model based on needs. For example:
 - scikit-learn
 - pyfunc
 - spark
 - tensorflow



Model Registry

Home

Workspace

Recents

Data

Clusters

Jobs

Models

Search

Registered Models

search model name

| Name | Latest Version | Staging | Production | Last Modified |
|------------------------------|----------------|-----------|------------|---------------------|
| Item_Recommender | Version 5 | Version 5 | Version 4 | 2019-10-11 15:30:02 |
| Airline_Delay_Scikit | Version 3 | — | Version 1 | 2019-10-11 12:41:43 |
| Airline_Delay_SparkML | Version 5 | Version 5 | Version 3 | 2019-10-11 12:45:15 |
| Transaction_Fraud_Classifier | Version 1 | — | — | 2019-10-11 15:18:05 |
| Icon_GAN | Version 1 | — | — | 2019-10-12 08:20:12 |
| Power_Forecasting_Model | Version 1 | — | Version 1 | 2019-10-07 15:38:27 |
| Product_Image_Classifier | Version 6 | — | Version 5 | 2019-10-12 00:38:56 |
| Comment_Summarizer | Version 3 | Version 2 | Version 3 | 2019-10-12 00:39:40 |
| Movie_Recommender | Version 5 | Version 5 | Version 3 | 2019-10-10 14:07:07 |
| Translation_Alpha | — | — | — | 2019-10-11 16:45:01 |

< 1 2 3 >

Model Registry

databricks

Home

Workspace

Recents

Data

Clusters

Jobs

Models

Search

?

search model name

| Name | Latest Version | Staging | Production | Last Modified |
|------------------------------|----------------|-----------|------------|---------------------|
| Item_Recommender | Version 5 | Version 5 | Version 4 | 2019-10-11 15:30:02 |
| Airline_Delay_Scikit | Version 3 | — | Version 1 | 2019-10-11 12:41:43 |
| Airline_Delay_SparkML | Version 5 | Version 5 | Version 3 | 2019-10-11 12:45:15 |
| Transaction_Fraud_Classifier | Version 1 | — | — | 2019-10-11 15:18:05 |
| Icon_GAN | Version 1 | — | — | 2019-10-12 08:20:12 |
| Power_Forecasting_Model | Version 1 | — | Version 1 | 2019-10-07 15:38:27 |
| Product_Image_Classifier | Version 6 | — | Version 5 | 2019-10-12 00:38:56 |
| Comment_Summarizer | Version 3 | Version 2 | Version 3 | 2019-10-12 00:39:40 |
| Movie_Recommender | Version 5 | Version 5 | Version 3 | 2019-10-10 14:07:07 |
| Translation_Alpha | — | — | — | 2019-10-11 16:45:01 |

<

1

2

3

>

Model Registry

databricks

Home

Workspace

Recents

Data

Clusters

Jobs

Models

Search

?

search model name

| Name | Latest Version | Staging | Production | Last Modified |
|------------------------------|----------------|-----------|------------|---------------------|
| Item_Recommender | Version 5 | Version 5 | Version 4 | 2019-10-11 15:30:02 |
| Airline_Delay_Scikit | Version 3 | — | Version 1 | 2019-10-11 12:41:43 |
| Airline_Delay_SparkML | Version 5 | Version 5 | Version 3 | 2019-10-11 12:45:15 |
| Transaction_Fraud_Classifier | Version 1 | — | — | 2019-10-11 15:18:05 |
| Icon_GAN | Version 1 | — | — | 2019-10-12 08:20:12 |
| Power_Forecasting_Model | Version 1 | — | Version 1 | 2019-10-07 15:38:27 |
| Product_Image_Classifier | Version 6 | — | Version 5 | 2019-10-12 00:38:56 |
| Comment_Summarizer | Version 3 | Version 2 | Version 3 | 2019-10-12 00:39:40 |
| Movie_Recommender | Version 5 | Version 5 | Version 3 | 2019-10-10 14:07:07 |
| Translation_Alpha | — | — | — | 2019-10-11 16:45:01 |

<

1

2

3

>

Model Registry

Home

Workspace

Recents

Data

Clusters

Jobs

Models

Search

Registered Models

search model name

| Name | Latest Version | Staging | Production | Last Modified |
|------------------------------|----------------|-----------|------------|---------------------|
| Item_Recommender | Version 5 | Version 5 | Version 4 | 2019-10-11 15:30:02 |
| Airline_Delay_Scikit | Version 3 | — | Version 1 | 2019-10-11 12:41:43 |
| Airline_Delay_SparkML | Version 5 | Version 5 | Version 3 | 2019-10-11 12:45:15 |
| Transaction_Fraud_Classifier | Version 1 | — | — | 2019-10-11 15:18:05 |
| Icon_GAN | Version 1 | — | — | 2019-10-12 08:20:12 |
| Power_Forecasting_Model | Version 1 | — | Version 1 | 2019-10-07 15:38:27 |
| Product_Image_Classifier | Version 6 | — | Version 5 | 2019-10-12 00:38:56 |
| Comment_Summarizer | Version 3 | Version 2 | Version 3 | 2019-10-12 00:39:40 |
| Movie_Recommender | Version 5 | Version 5 | Version 3 | 2019-10-10 14:07:07 |
| Translation_Alpha | — | — | — | 2019-10-11 16:45:01 |

< 1 2 3 >

Model Serving

The screenshot displays the Databricks Model Serving interface for an endpoint named 'churn_prediction_test'. The interface includes a sidebar with navigation icons, a breadcrumb 'Endpoints >', and a 'Permissions' button. The endpoint state is 'Ready' with a green checkmark, and the URL is 'https://e2-dogfood.staging.cloud.databricks.com/model-endpoint/charlie/1/invocations'. Below this, the 'Served models' section contains a table with one entry: 'churn prediction' (Version 1, churn-prediction-1, Ready, Large 16-64 (8 provisioned)). An 'Edit configuration' button is next to the table. At the bottom, the 'Call endpoint' tab is active, showing 'Request' and 'Response' input fields, a 'Send Request' button, and a 'Show Example' button. Other tabs include 'Metrics', 'Logs', and 'Events'.

Endpoints > churn_prediction_test Permissions

Endpoint state: Ready
URL: <https://e2-dogfood.staging.cloud.databricks.com/model-endpoint/charlie/1/invocations>

Served models Edit configuration

| Model | Version | Name | State | Compute |
|------------------|-----------|--------------------|--------------------|-----------------------------|
| churn prediction | Version 1 | churn-prediction-1 | Ready | Large 16-64 (8 provisioned) |

Call endpoint Metrics Logs Events

Browser Curl Python

Request ⓘ Response ⓘ

Send Request Show Example

Model Serving

The screenshot shows the Databricks Model Serving interface for an endpoint named `churn_prediction_test`. The endpoint state is **Ready**, and its URL is `https://e2-dogfood.staging.cloud.databricks.com/model-endpoint/charlie/1/invocations`. Below this, a table titled "Served models" lists the models served by the endpoint. The table has columns: Model, Version, Name, State, and Compute. The first row shows a model named `churn prediction` with version `Version 1`, name `churn-prediction-1`, state `Ready`, and compute configuration `Large 16-64 (8 provisioned)`. The `Compute` column header and the `Large 16-64 (8 provisioned)` value are highlighted with a red box. Below the table, there are tabs for `Call endpoint`, `Metrics`, `Logs`, and `Events`. The `Call endpoint` tab is active, showing a `Request` and `Response` section. The `Request` section has a `Send Request` button, and the `Response` section has a `Show Example` button.

| Model | Version | Name | State | Compute |
|------------------|-----------|--------------------|-------|-----------------------------|
| churn prediction | Version 1 | churn-prediction-1 | Ready | Large 16-64 (8 provisioned) |

Model Serving

The screenshot displays the Databricks Model Serving interface. At the top, the breadcrumb 'Endpoints >' is followed by the endpoint name 'churn_prediction_test'. To the right of the name are three vertical dots and a 'Permissions' button. Below this, the 'Endpoint state' is 'Ready' in green. The 'URL' is 'https://e2-dogfood.staging.cloud.databricks.com/model-endpoint/charlie/1/invocations'. A 'Served models' section contains a table with one model. The table has columns: Model, Version, Name, State, and Compute. The model 'churn prediction' (Version 1) is in 'Ready' state and is running on 'Large 16-64 (8 provisioned)' compute. Below the table, there are tabs for 'Call endpoint', 'Metrics', 'Logs', and 'Events'. The 'Call endpoint' tab is active, showing 'Browser', 'Curl', and 'Python' options. There are empty text areas for 'Request' and 'Response', with 'Send Request' and 'Show Example' buttons at the bottom.

Endpoints > churn_prediction_test

Endpoint state: **Ready**

URL: https://e2-dogfood.staging.cloud.databricks.com/model-endpoint/charlie/1/invocations

Served models

| Model | Version | Name | State | Compute |
|------------------|-----------|--------------------|-------|-----------------------------|
| churn prediction | Version 1 | churn-prediction-1 | Ready | Large 16-64 (8 provisioned) |

Call endpoint Metrics Logs Events

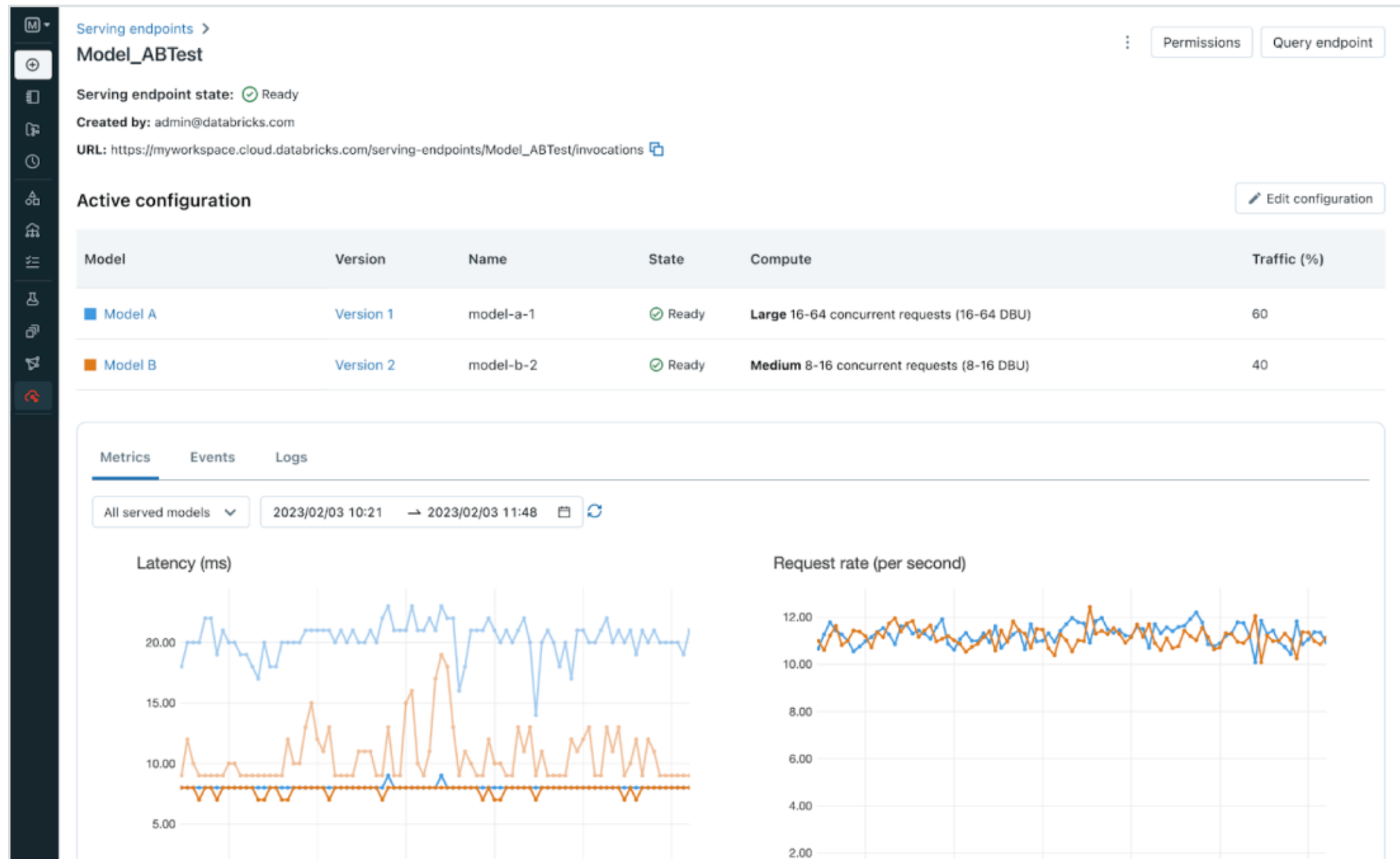
Browser Curl Python

Request

Response

Send Request Show Example

Model Serving



¹ <https://www.databricks.com/product/model-serving>

Let's practice!
DATABRICKS CONCEPTS

Example end-to-end machine learning pipeline

DATABRICKS CONCEPTS



Kevin Barlow
Data Practitioner

Let's practice!

DATABRICKS CONCEPTS

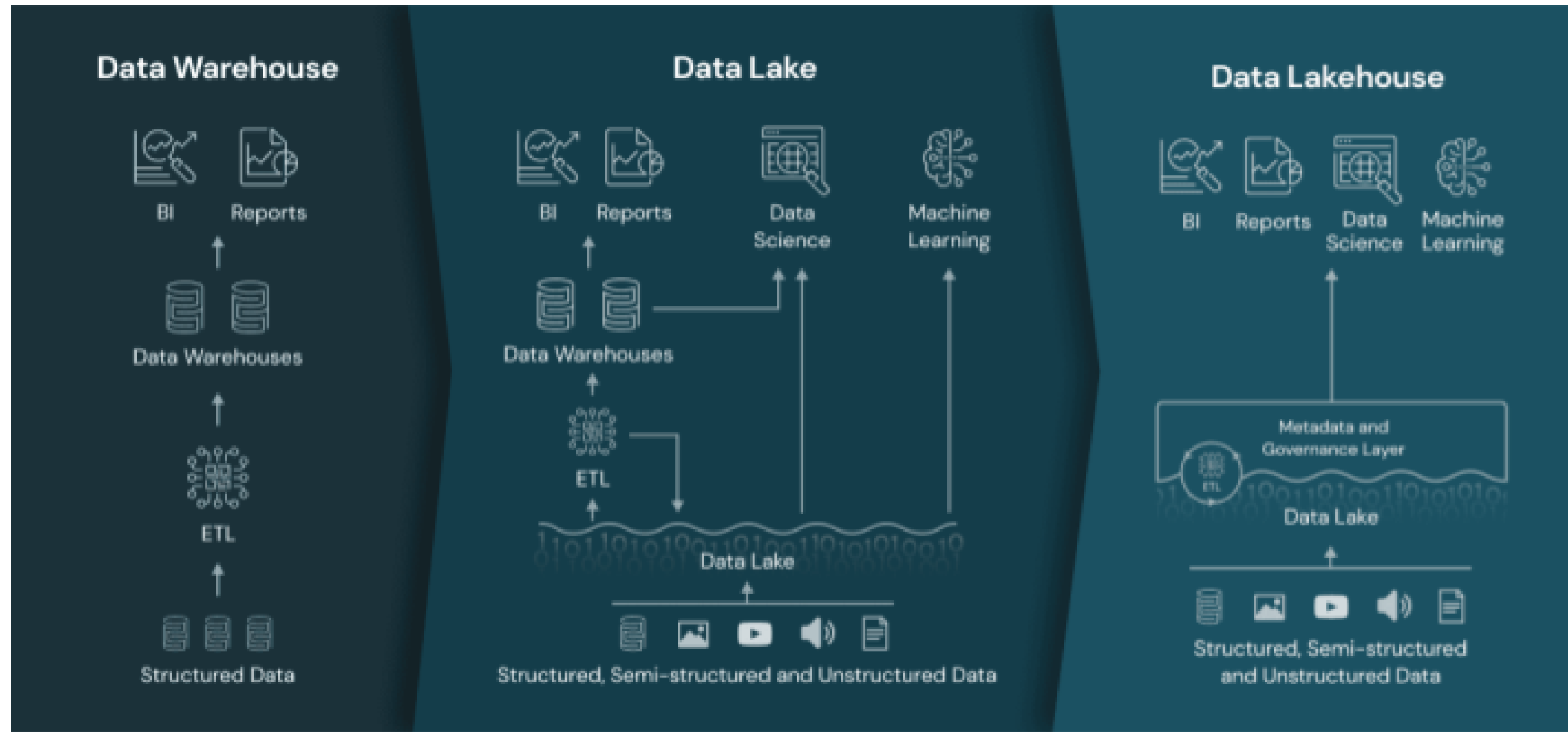
Wrap Up

DATABRICKS CONCEPTS



Kevin Barlow
Data Practitioner

Why the Lakehouse?



¹ <https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html>

Databricks for data engineering

Apache Spark

Delta

Delta Live Tables

Auto Loader

Structured Streaming

Workflows



Databricks for data warehousing

SparkSQL

ANSI SQL

SQL Warehouses

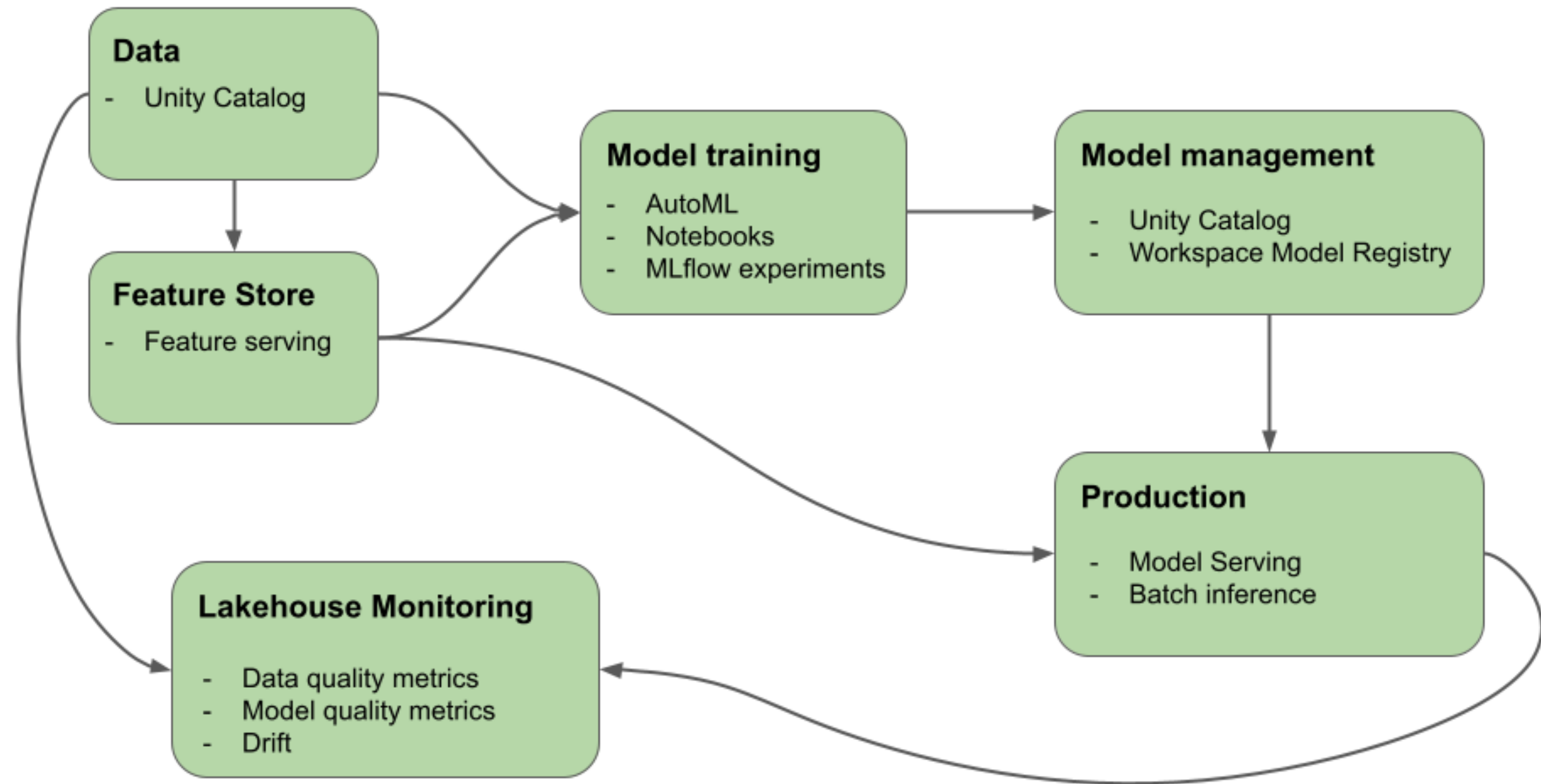
Queries

Visualizations

Dashboards



Databricks for machine learning



Congratulations!

DATABRICKS CONCEPTS