CSCI E-103

*Data Engineering for Analytics to Solve Business Challenges*

# Towards Reproducible Machine Learning

***Lecture 08***

Anindita Mahapatra & Eric Gieseke

Harvard Extension, Fall 2025

# Agenda

- ML Review for the Data Persona
- Role of Data Engineering in ML
- What is a Model Lifecycle
- Challenges in ML development
- ML Pipelines - Formalizing the ML development process for Model Reproducibility
- Feature Store
- MLOps - Managing ML lifecycle using SDLC discipline
- Intro to MLFlow

- Lab
    - MLFlow for model lifecycle management & MLOps
    - https://docs.databricks.com/aws/en/notebooks/source/mlflow/mlflow-classic-ml-e2e-mlflow-3.html
        - Replace main.default with cscie103_catalog.default

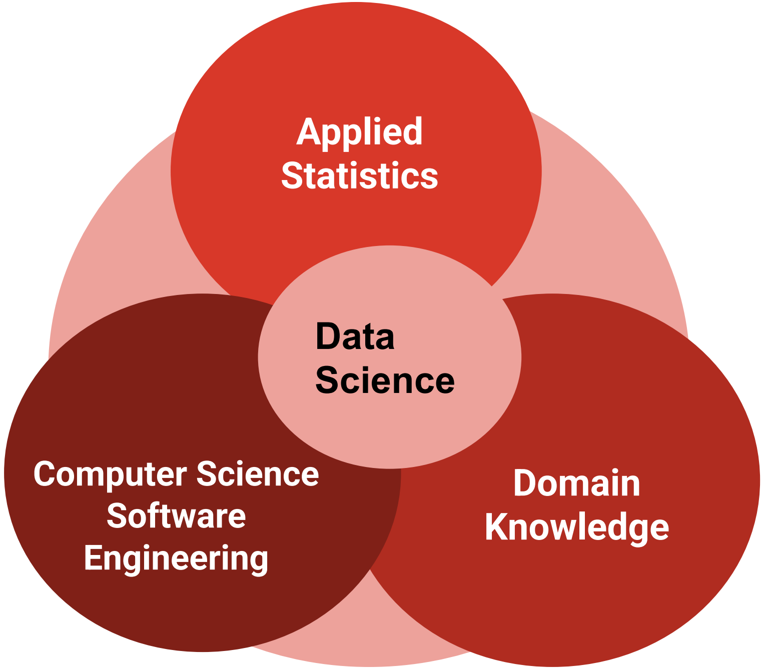
# Review Previous Material

| | |
|---|---|
| Scale Out refers to | Adding more nodes |
| Autoscale uses | Scale out |
| Ability to use the service anytime is | Availability |
| Planning to switch to a different **region** on an outage | Disaster Recovery |
| Ability to grow/shrink nodes refers to | Elasticity |
| Example of a Bridge Pattern in Spark | Connector Pattern |
| What are the advantages of multi-hop pattern | Cater to different SLAs, more robust, modular |
| What is BI Vs BA | What Happened Vs Why/What Next |
| 2 KPIs of BI tool performance | Latency & Concurrency |
| Architecture paradigm that supports AI+BI | Lakehouse |

# Data Science

## Process & Skills

Question → Hypothesize → Experiment → Interpret → Share Results




Using Data to Solve

Measurable Real World Problems

"The scientific method is a **dynamic** process that allows for the **continual advancement** of human understanding of the natural world.

It is characterized by its commitment to empirical evidence, objectivity, and the use of critical thinking to refine and expand our knowledge."

# Algorithms & Machine Learning

## How it relates to Data Science

**Rule Based**

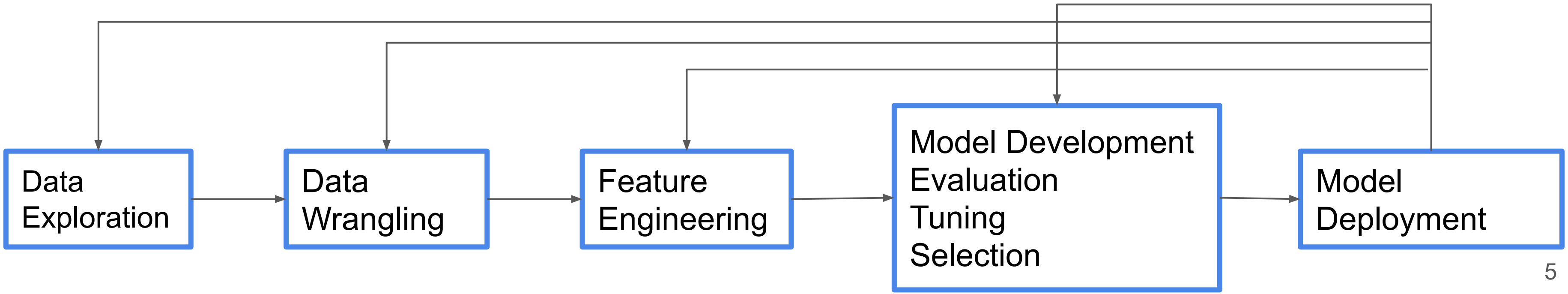
Set of facts + Set of Rules
Deterministic - if - then - else
Beyond a certain set, it is unmanageable

**AI Based**

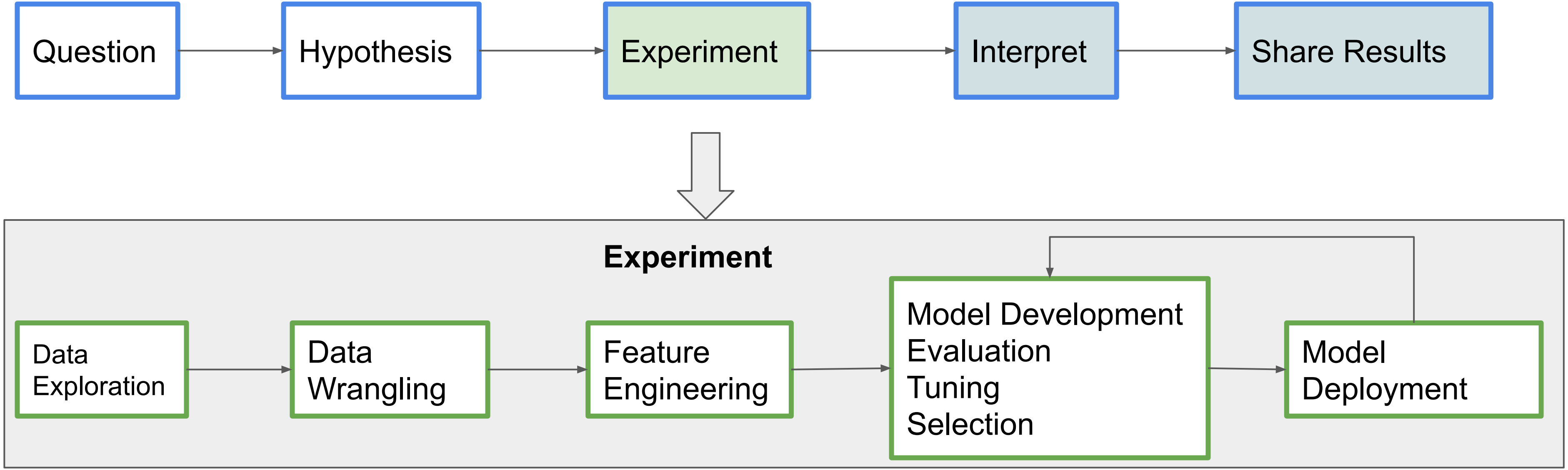
May have a set of facts
Probabilistic
Require more data points and are scalable

**AI/ML**: Perform tasks without being explicitly told via rules aka if/then/else i.e. learn relationships (algorithms)

- Columns -> Features -> Predict
- More data -> more wisdom -> better
- More data -> algorithm needs to be computationally efficient



# ML as a part of Data Science



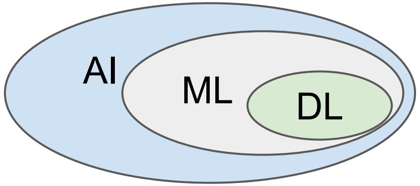

ML is a part of Data Science

Primarily the Experiment and Analysis Phase

Extends into Interpretability and Result Delivery

# AI Vs ML Vs DL

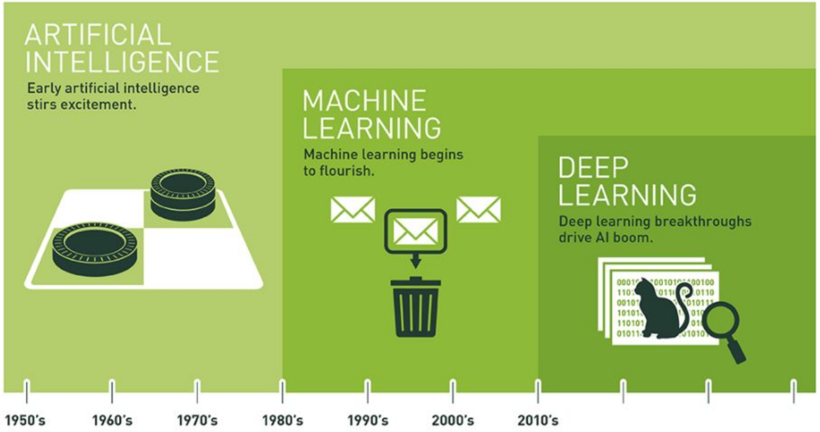
## Role of Spark




**AI** - machines perform tasks normally requiring human intelligence

**ML** - the process of <u>learning from data</u> without being explicitly programmed aka rule based

**DL** - Deep Learning: complex neural networks detect patterns in large unstructured data sets




APACHE Spark + AI

"Hidden Technical Debt in Machine Learning Systems ", Google

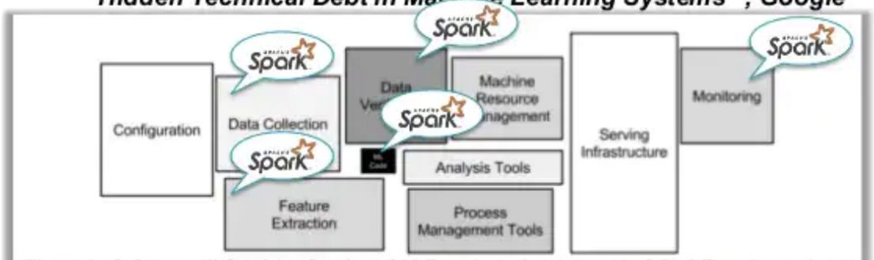



Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

FLEXIBLE | FAST | BIG DATA

### So what has changed lately?

- Lots of Data & Affordable Compute

# ML Frameworks & Libraries

- Classical/Statistical Algorithms
    - Suitable for most DS problems around structured/semi data
    - Mature and work with low data volume
- Neural Network/Deep Learning
    - Mostly unstructured data
    - Typically, performance increases significantly with training data volume
- Generative AI
    - LLM Large Language Models (text)
    - GAN Generative Adversarial Networks (images)

AutoML
Fast & Easy but less customizable

Notebook based
Powerful, but more complex

# Classification examples of ML types

| | Supervised Learning | Unsupervised Learning |
|---|---|---|
| Discrete | classification or categorization | clustering |
| Continuous | regression | dimensionality reduction |

How much will it rain tomorrow?

Will it rain tomorrow?

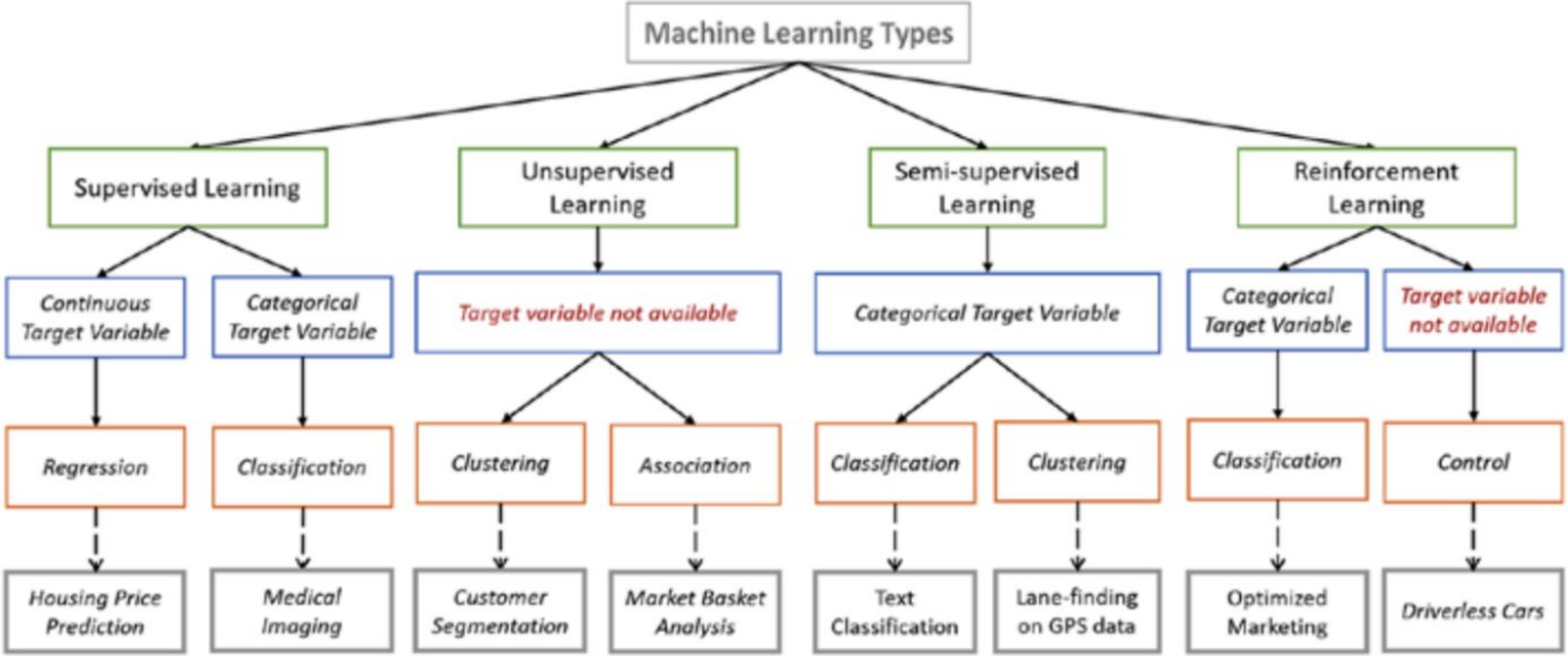
Which days of the week can be grouped based on weather patterns?

What are the key factors that influence consumer purchasing?

# Art of Prediction

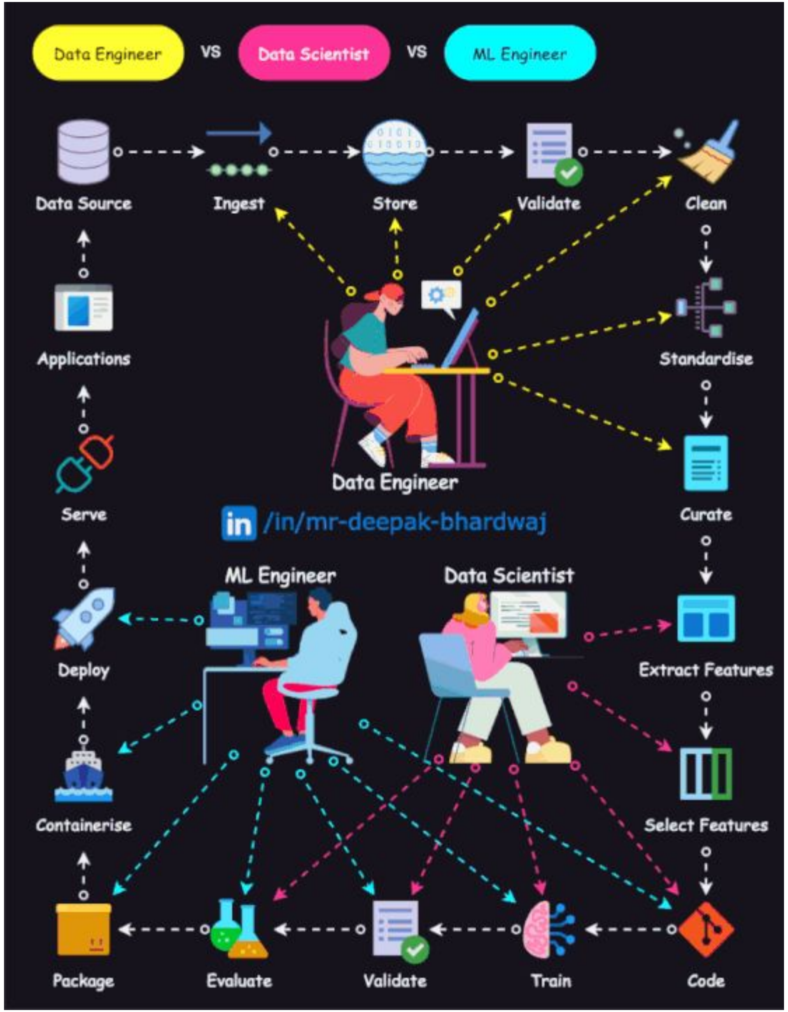

The art of looking into the future using the data from the past with/without labels

Lots of data & simple algorithms are more effective than the converse



# Role of Data Engineering in ML

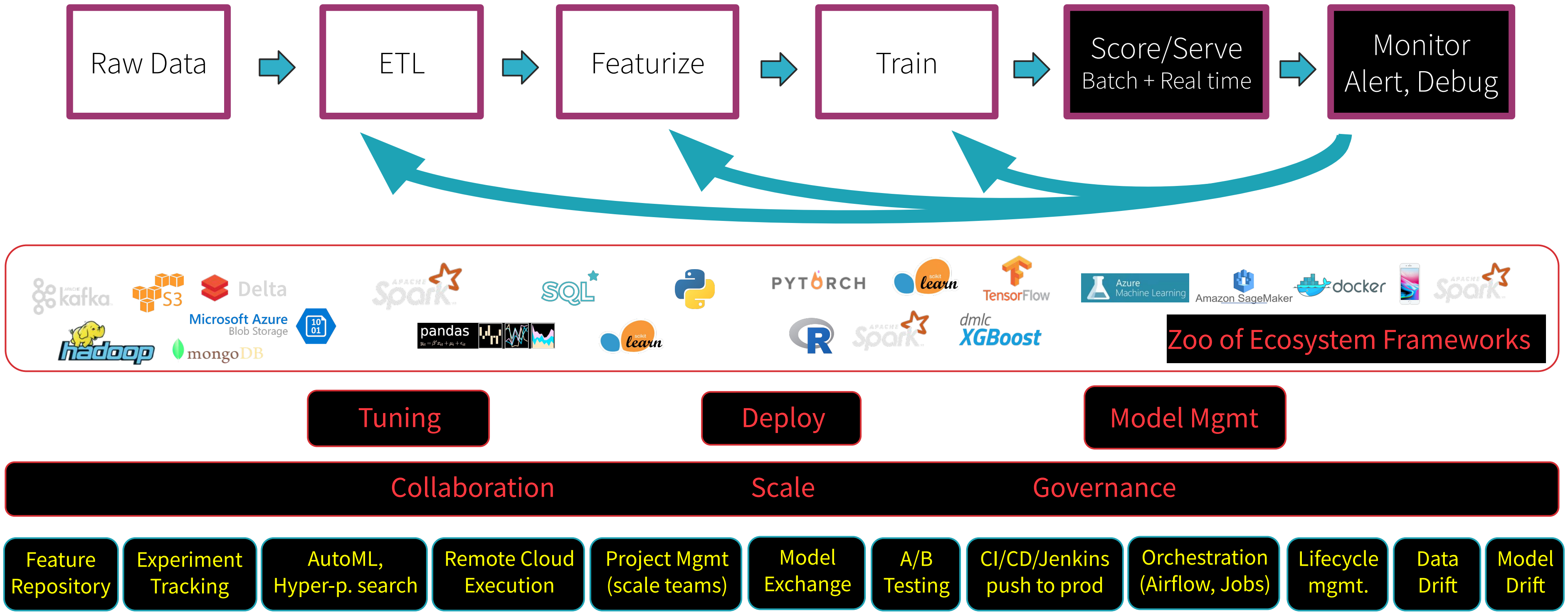
- Role Interplay
  - Data Engineer:  Infrastructure Architects
  - Data Scientist:  Insight Alchemists
  - ML Engineer:  Automation Crafters
- Data Integration
  - Data engineering is responsible for merging disparate sources of data into a unified view that can be used by an AI model.
- High Quality Data
  - It ensures that high-quality data is available, accessible, and usable for AI models.
  - Feature Engineering
- Scalability
  - Data engineering ensures that the entire data infrastructure can scale to handle increasing amounts of data
- Monitoring
  - Data Drift detection signals model retraining



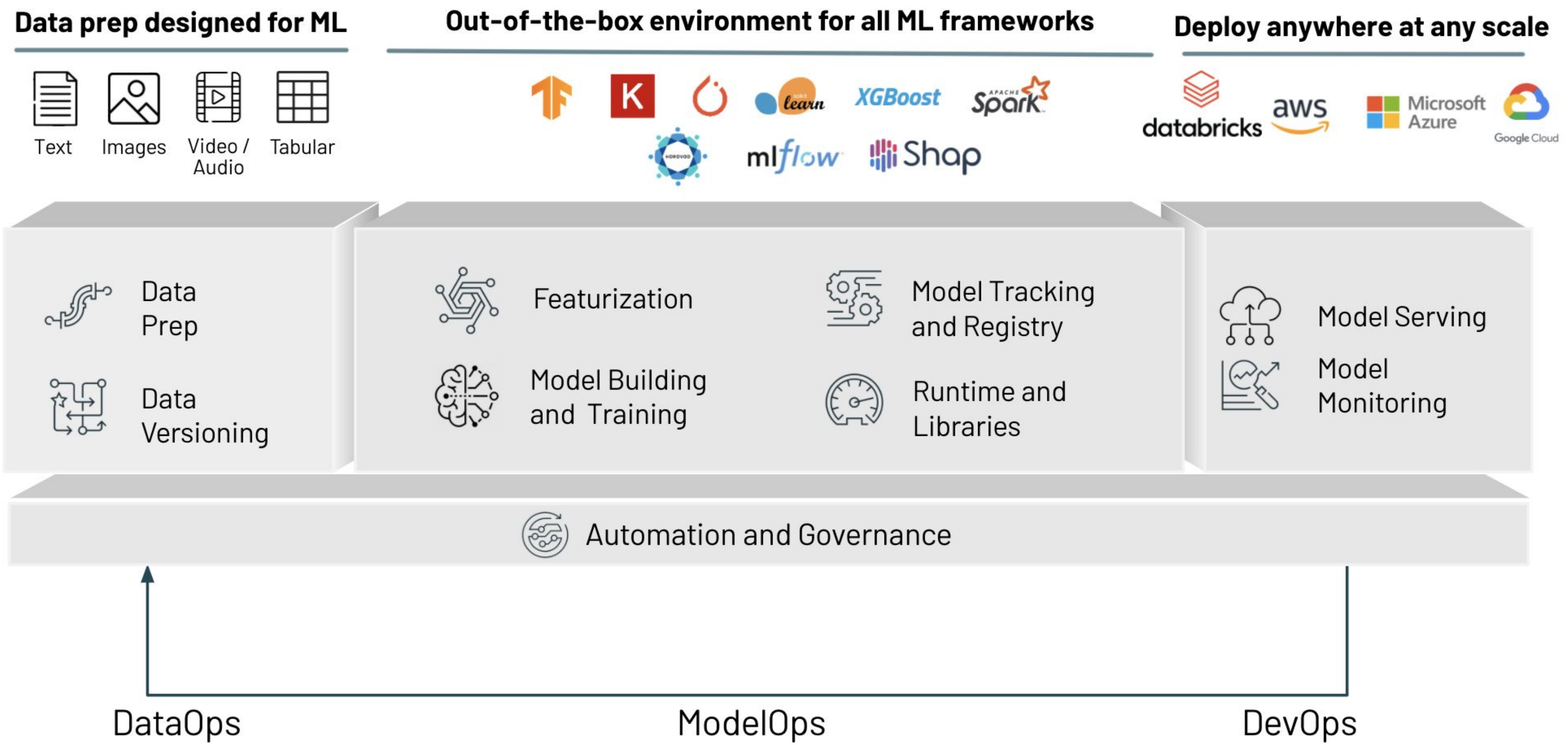
# Aspirations of a ML Platform

- Support & aid development  on diverse ML tools & frameworks
  - Libraries
  - Languages
  - Collaboration
- Support various deployment strategies
  - Batch, Streaming, REST EndPoint, Edge
- Single node & distributed algorithms (CPU/GPU/Classical/Deep Learning)
- Support higher efficiencies
  - Faster training, tuning, tracking
- Feature Store for feature reuse
- AutoML for Citizen data scientists
- Monitor drift
- Aid in operationalizing ML
- Reproducible ML

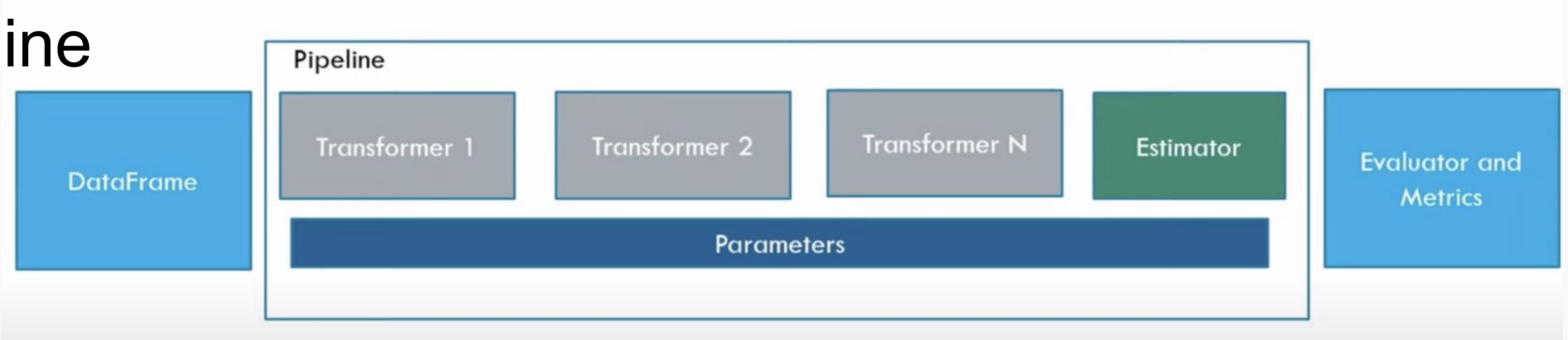
# ML Lifecycle and Challenges



# From Data Ingestion to Model Deployment



# ML Pipeline




**Stages**

- **Transformers** (preprocessing)
  - String Indexer : string to index
  - One Hot Encoding : index to a binary vector
  - Standard Scaler : Scales the data to unit standard deviation.
  - Imputer: completes missing values in a dataset, either using the mean or the median
  - VectorAssembler: combines a given list of columns into a single vector column
- **Estimators** (learn)
  - Fit: Input is a dataframe, output is a model

## Confusion Matrix

| | Actually Positive (1) | Actually Negative (0) |
|---|---|---|
| Predicted Positive (1) | True Positives (TPs) | False Positives (FPs) |
| Predicted Negative (0) | False Negatives (FNs) | True Negatives (TNs) |

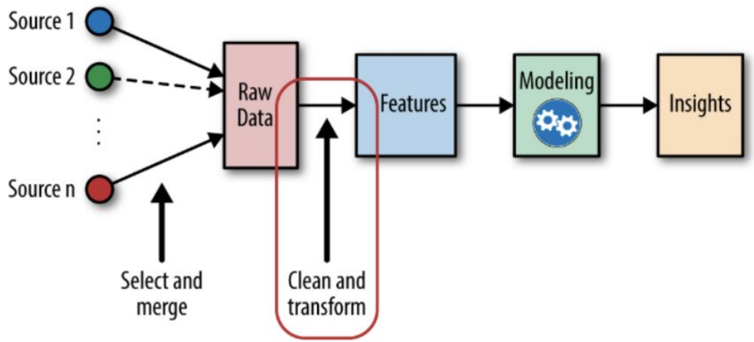
**Metrics:** Evaluation metrics explain the performance of a model

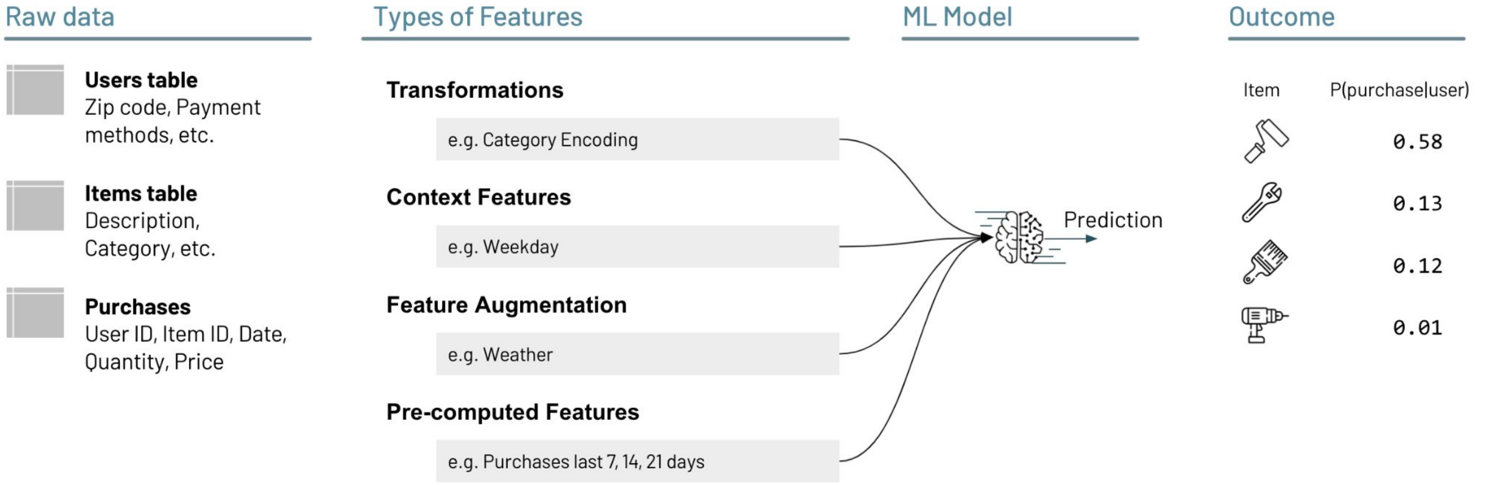
**Evaluators:** functions to calculate performance metrics based on actuals & predicted values

# Feature Engineering

## Feature Store

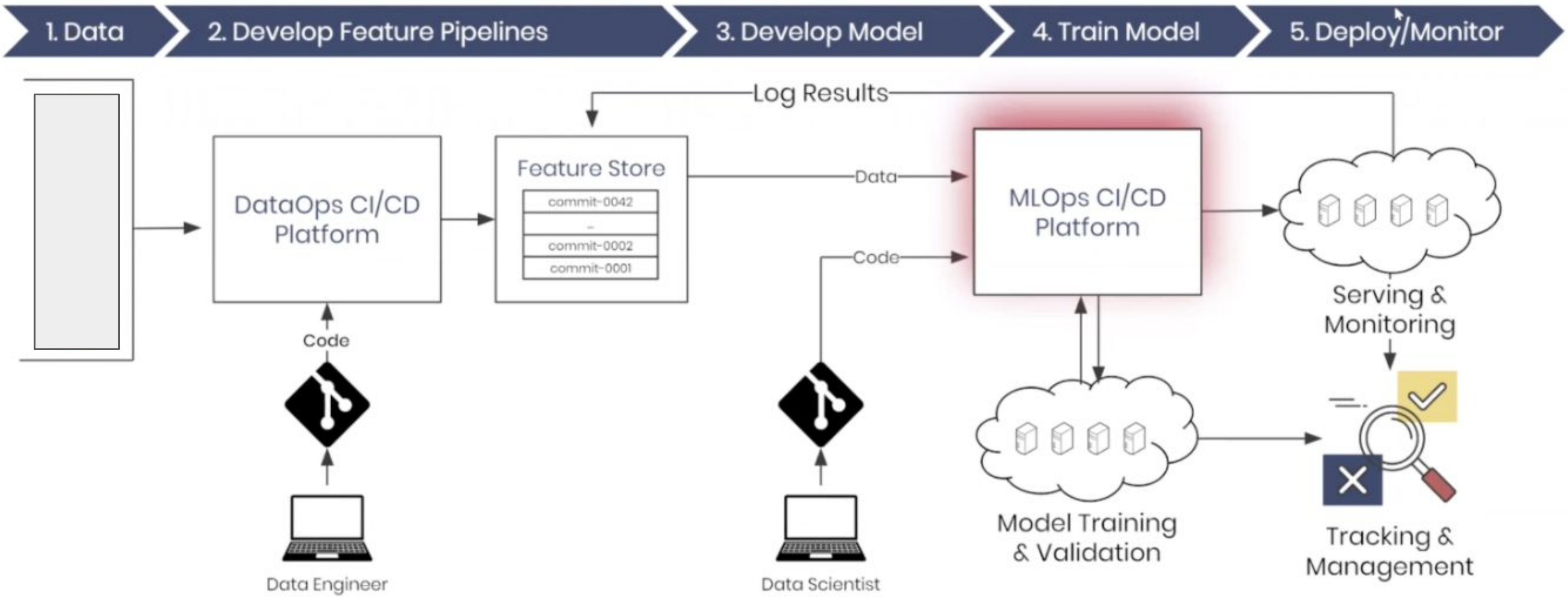
Process of using domain knowledge of the data to create features that make machine learning algorithms work.




*Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.*
— Andrew Ng, Machine Learning and AI via Brain simulations

# MLOps



1. Data
2. Develop Feature Pipelines
3. Develop Model
4. Train Model
5. Deploy/Monitor
Log Results
DataOps CI/CD Platform
Feature Store
commit-0042
commit-0002
commit-0001
Data
Code
MLOps CI/CD Platform
Serving & Monitoring
Code
Data Engineer
Data Scientist
Model Training & Validation
Tracking & Management

# MLflow

Open source project

Conventions, specs, tools

CLI, libraries, REST server, UI

Community backed

Framework and tool agnostic, on-prem & in-cloud

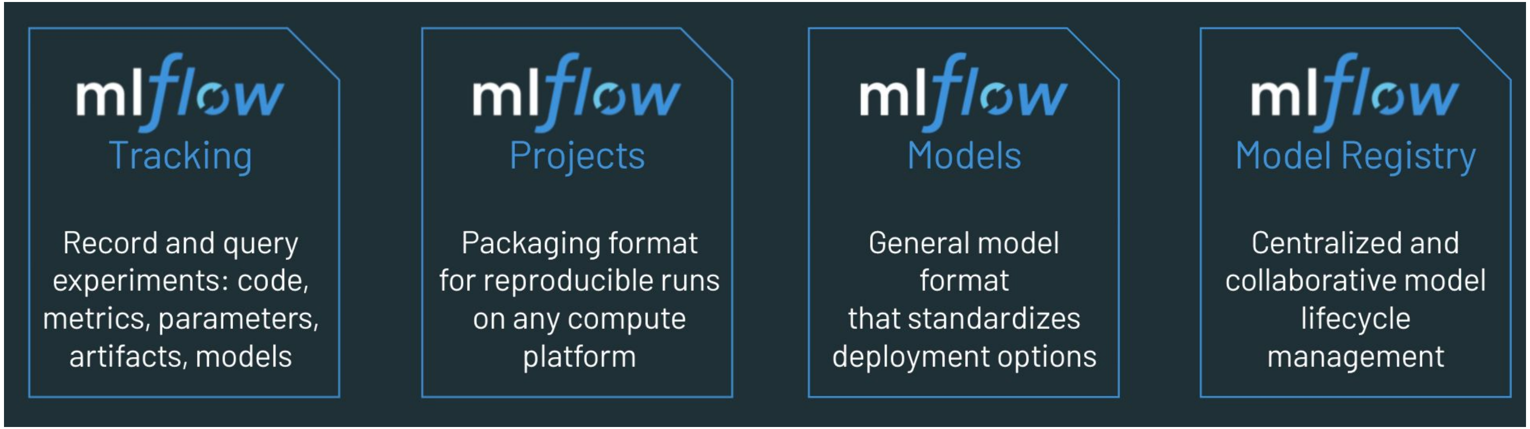
1) **API first:** built around REST APIs that allows submitting runs, models, etc. from any library & language

2) **Modular** : independent components (Tracking/Projects/Models) E.g., use MLflow's project format but not its deployment tools. Easy to integrate into existing ML platforms & workflows

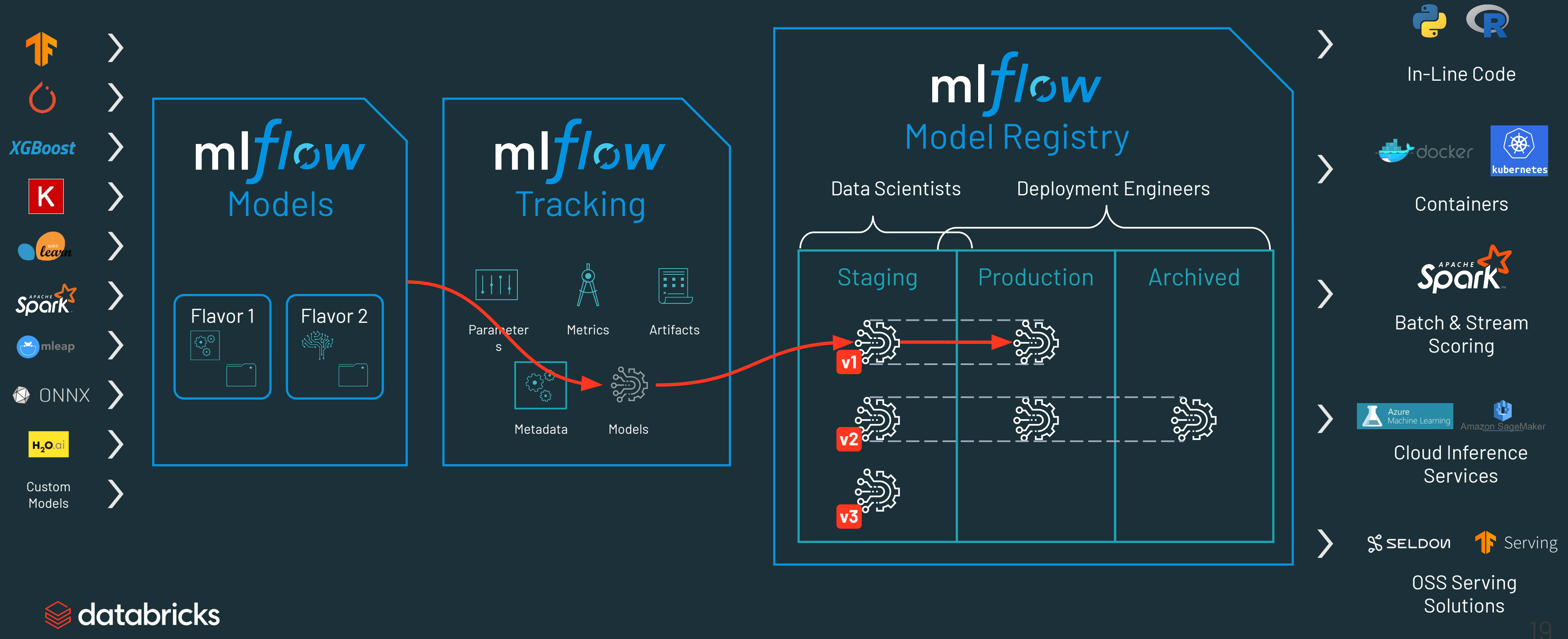
3) **Easy** : minimal, ubiquitous dependencies. Runs the same way anywhere (local, cloud platforms) Easy for a single dev to use locally, or very large teams

4) **Ecosystem first** : choose open and flexible core abstractions and interface points (e.g. entry point to an MLProject is a shell script so no bias to language runtime)Make it easy to use ecosystem frameworks, rather than competing with those frameworks.

5) **Open Source & Open interface**: MLflow is designed to work with any ML library, algorithm, deployment tool or language.



# mlflow Model Lifecycle

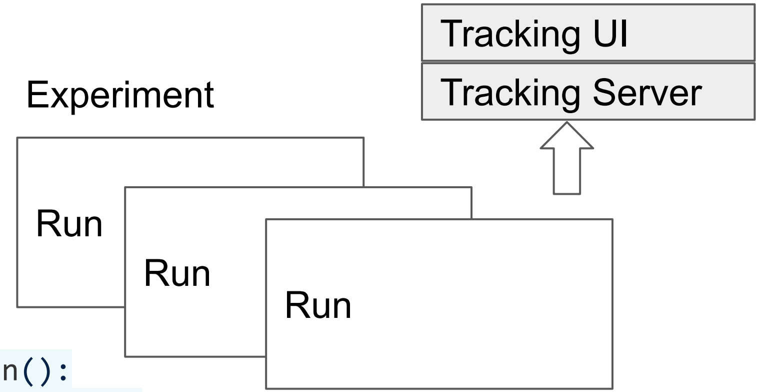



databricks

# MLFlow

## Tracking

- mlflow.start_run()
- mlflow.log_param()
- mlflow.log_metric()
- mlflow.set_tag()
- mlflow.log_artifact()
- mlflow.autolog()



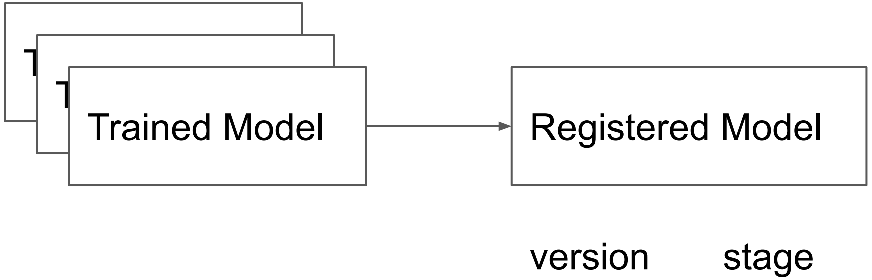

```python
with mlflow.start_run():
    mlflow.log_param("x", 1)
    mlflow.log_metric("y", 2)
    ...
```

## Registry

- `create_registered_model()`
- `create_model_version()`
- `Transition_model_version_stage()`
- `list_registered_models()`

```python
from mlflow.tracking import MlflowClient

client = MlflowClient()
client.create_registered_model("random-forest-model")
```



# Model & Data Drift

What to monitor?

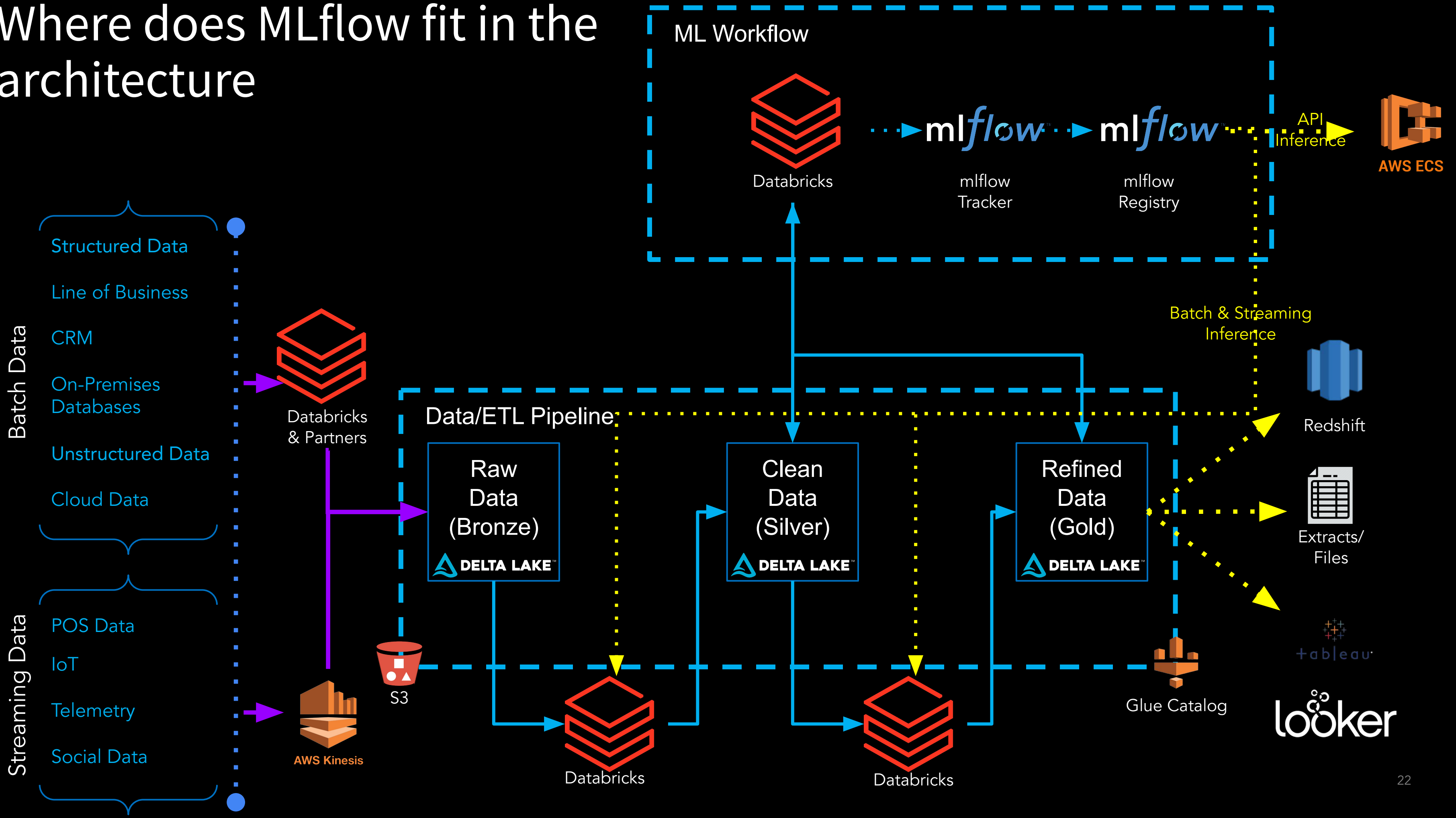

- Basic summary statistics of features & target
- Distribution of features & target
- Model Performance Metrics
- Business Metrics

| Drift Type Identified | Action |
|---|---|
| **Feature Drift** | • Investigate feature generation process<br>• Retrain using new data |
| **Label Drift** | • Investigate label generation process<br>• Retrain using new data |
| **Prediction Drift** | • Investigate model training process<br>• Assess business impact of change in predictions |
| **Concept Drift** | • Investigate additional feature engineering<br>• Consider alternative approach/solution<br>• Retrain/tune using new data |

Talks:
Testing ML Models in Production

Webinars:
Productionalizing ML

# Where does MLflow fit in the architecture



# ToDos

- Read chapters 7,8, 9
- Lab
  - XGBoost: Link
  - Replace main.default with cscie103_catalog.default or something that you have access to
- Lab
  - Scikit: Link
  - Update `CATALOG_NAME = "cscie103_catalog"`
  - **Use local Hyperopt trials instead of SparkTrials.**