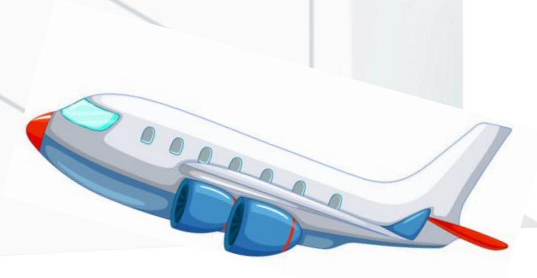
www.advancinganalytics.co.uk



# FIRST CLASS MLOPS WITH DATABRICKS


databricks

FIRST CLASS

# WHO ARE WE?

- Tori Tompkins
  - Senior Data Science Consultant
  - Cohost of Data Podcast, Totally Skewed


- Alex Billington
  - Senior AI Consultant
  - ML Engineer for 5.5 years

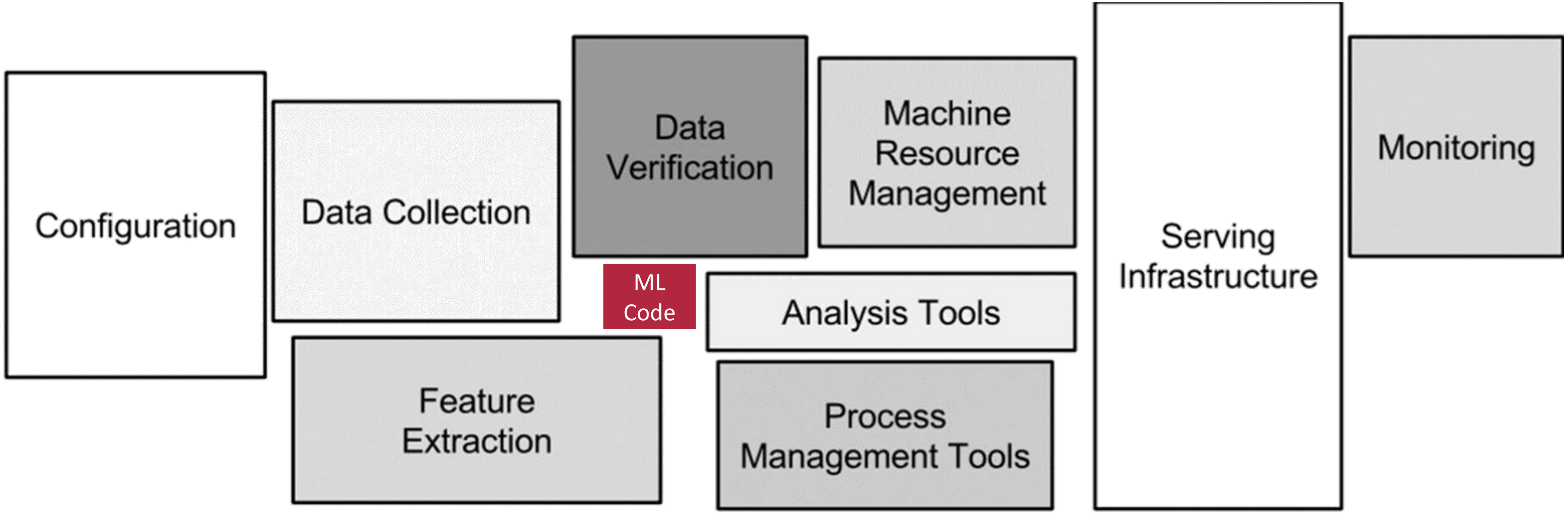
# AGENDA

| Time | Topic |
|---|---|
| 09:00 - 09:10 | Introduction to **MLOps** |
| 09:10 – 09:30 | Model & Experiment Tracking with **Mlflow** ft. **SHAP** |
| 09:30 – 09:50 | **Feature Stores** |
| 09:50 – 10:00 | Break! |
| 10:00 – 10:10 | Deployment Strategies |
| 10:10 – 10:30 | Model Deployment with **Databricks Real-Time Endpoints** |
| 10:30 – 10:50 | Model Monitoring with **Unity Catalog** |

# INTRO TO MLOPS AND THE ML LIFECYCLE


WHEN YOU START WORKING ON THE ML INFRASTRUCTURE INSTEAD OF THE MODEL
1 hour here is 7 years on earth
imgflip.com

# A TYPICAL MACHINE LEARNING PROJECT



# WHAT IS MLOPS?

- MLOps is a set of **processes** and **automation**
  - For managing **code, data,** and **models**
  - To improve performance, stability and long-term efficiency of ML systems

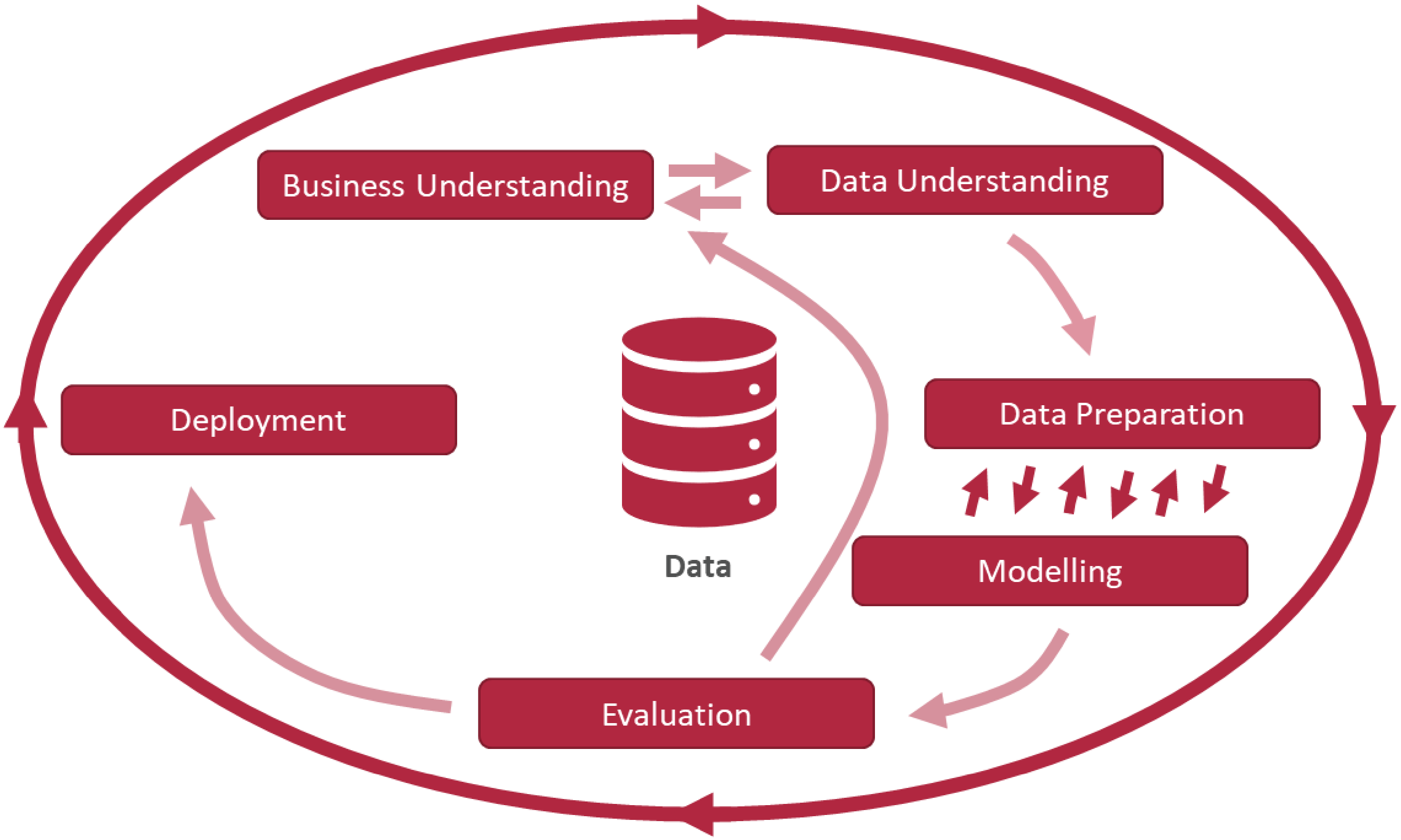
MLOps = DevOps + DataOps + ModelOps

# WHY IS MLOPS IMPORTANT?

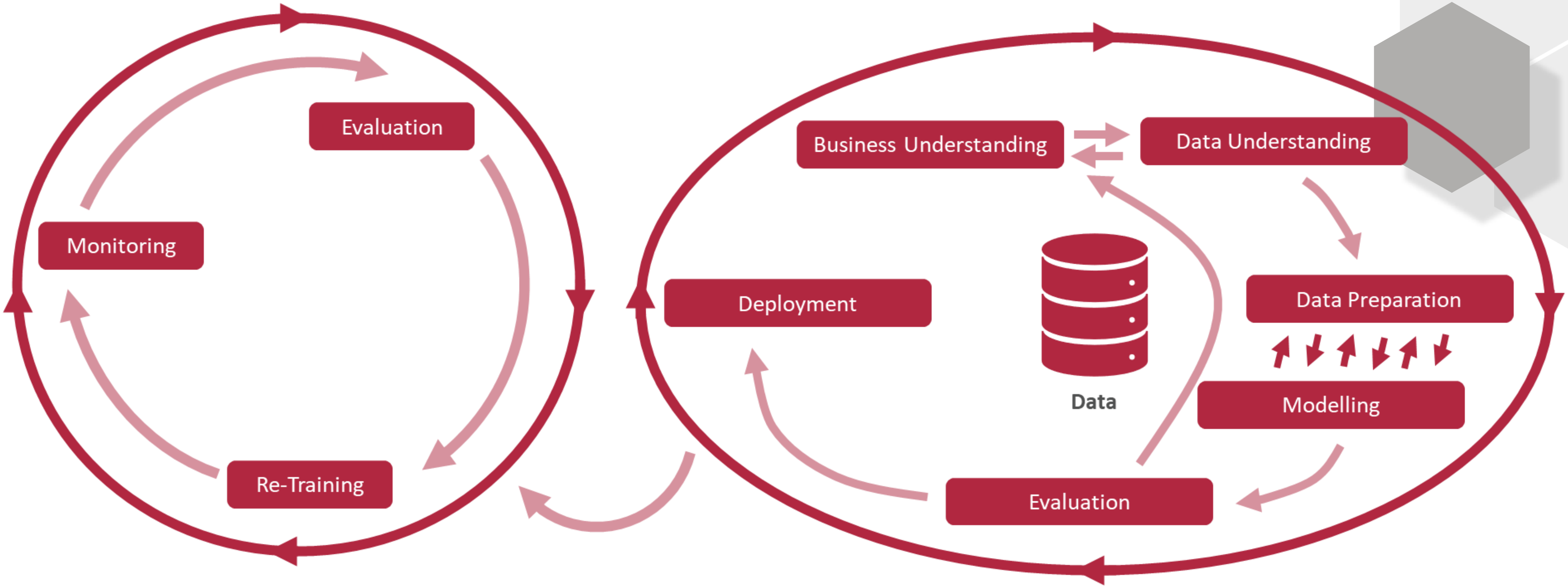
- Scaling
- Trust
- Better Integration
- Compliance
- Reduce Risk and Bias

**The MLOps market could grow to around $2 billion by 2025, up from about $185 million in 2020.**

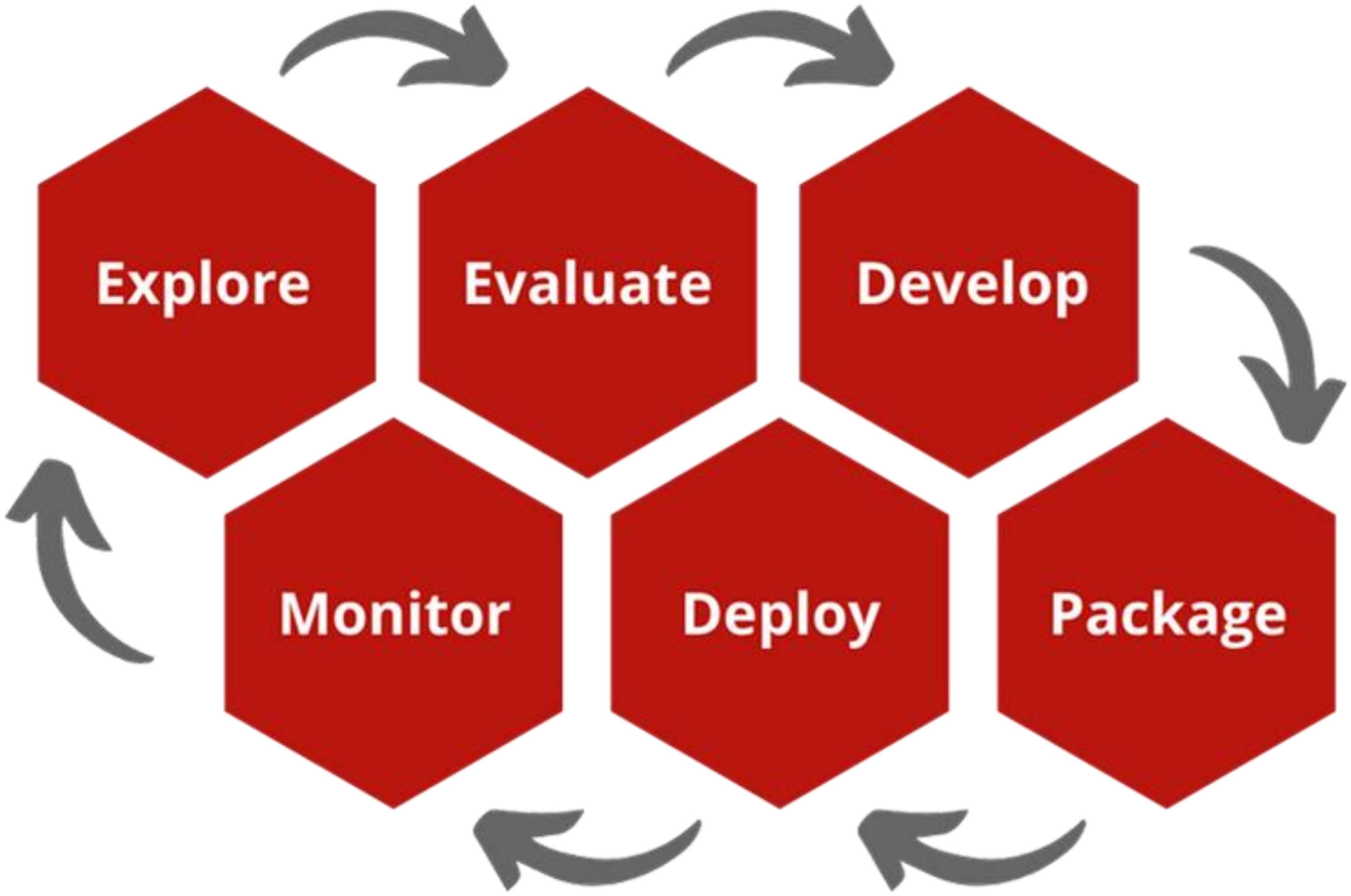
# WHY IS MLOPS DIFFERENT FROM DEVOPS?

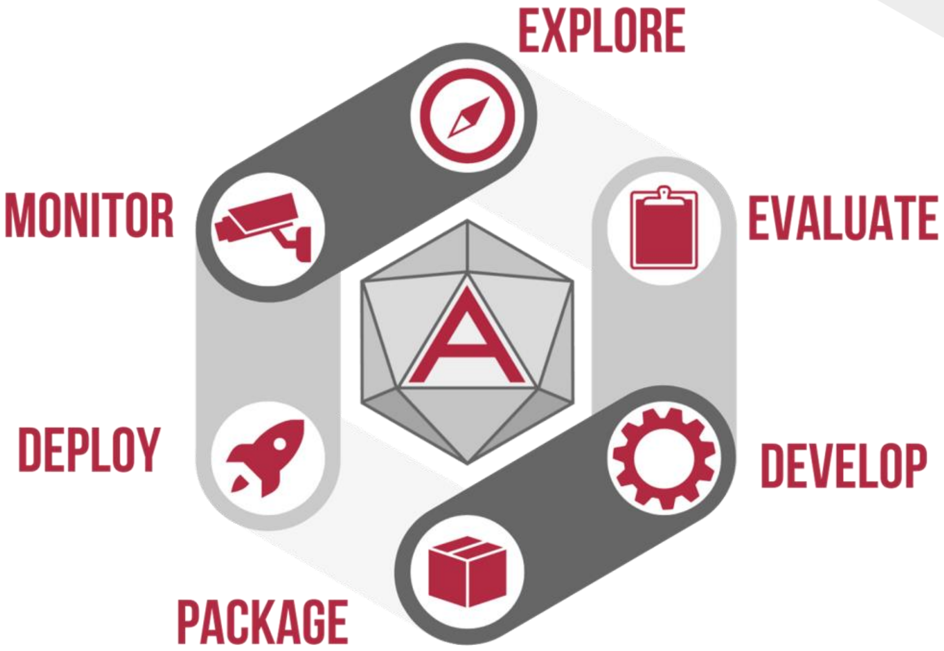


# WHY IS MLOPS DIFFERENT FROM DEVOPS?



# MLOPS LIFECYCLE



# EXPLORE

**Explore**

- Experimental Phase for Data Scientists (DS) to try models
- MLOps should supply DS the environment they need to achieve this
- This can include leveraging feature stores, model experiment tracking and collaborative workspace

# EVALUATE

## Evaluate

- Testing phase for models to be evaluated for business value, accuracy and ethics
- MLOps should provide a platform and process to achieve this
- This can include leveraging libraries to analyse results, interpretability, explainability and fairness

# DEVELOP

## Develop

- Development phase to build robustness into the model inference pipeline
- MLOps should ensure models continuing to the next phase are production-ready
- This can include auditing, unit-testing, data and model versioning

# PACKAGE

**Package**

- Packaging phase to ensure the production environment resembles development
- MLOps should ensure resources are accurate and accessible to prevent unexpected behaviour
- This can include versioning of libraries and frameworks or packaging code into a wheel

# DEPLOY

## Deploy

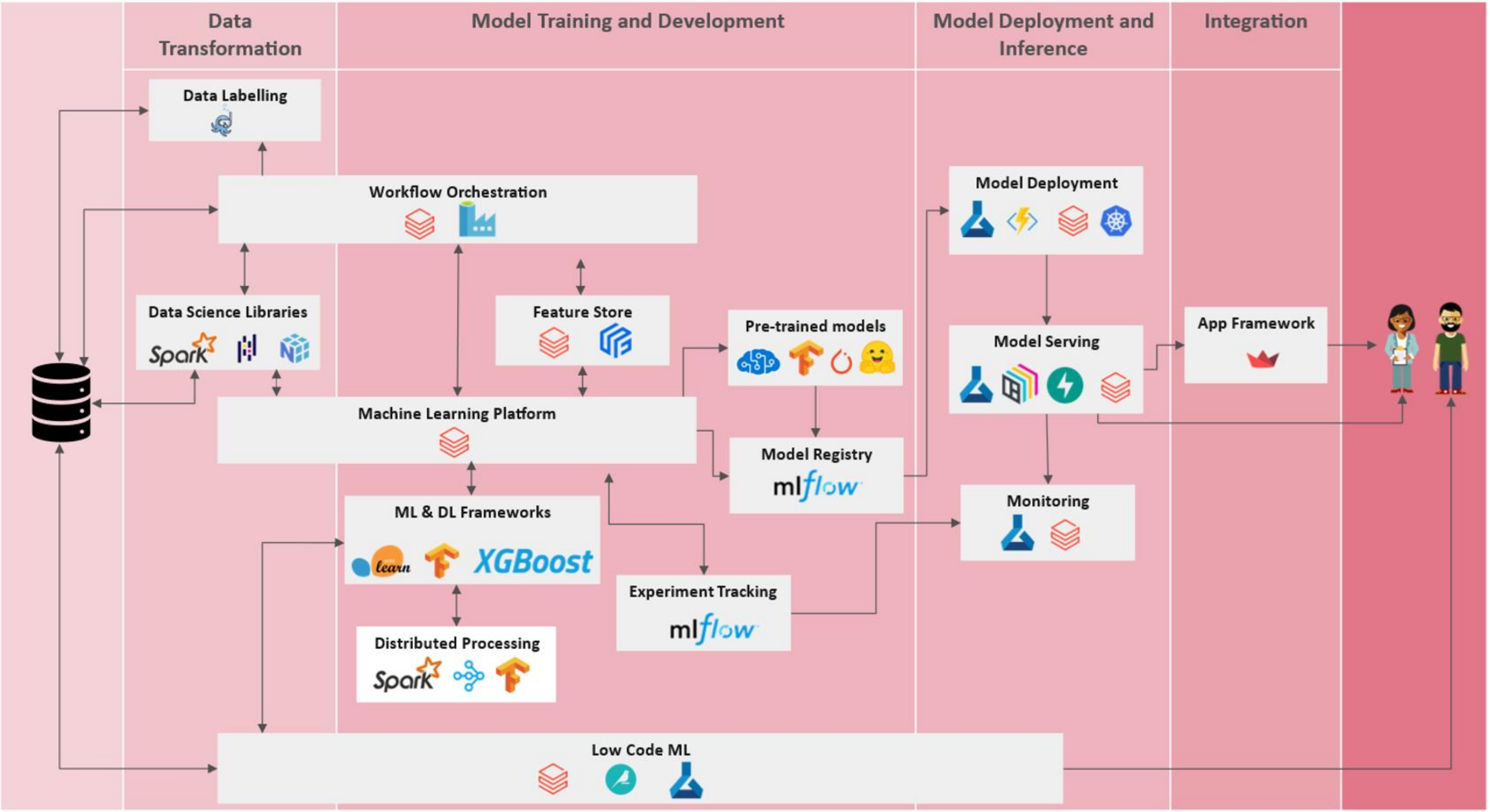
- Deployment phase to serve the model to user base
- MLOps should deploy the model according to business requirements
- This can include considering model complexity, size, importance, need for autoscaling or A/B testing etc

# MONITOR

## Monitor

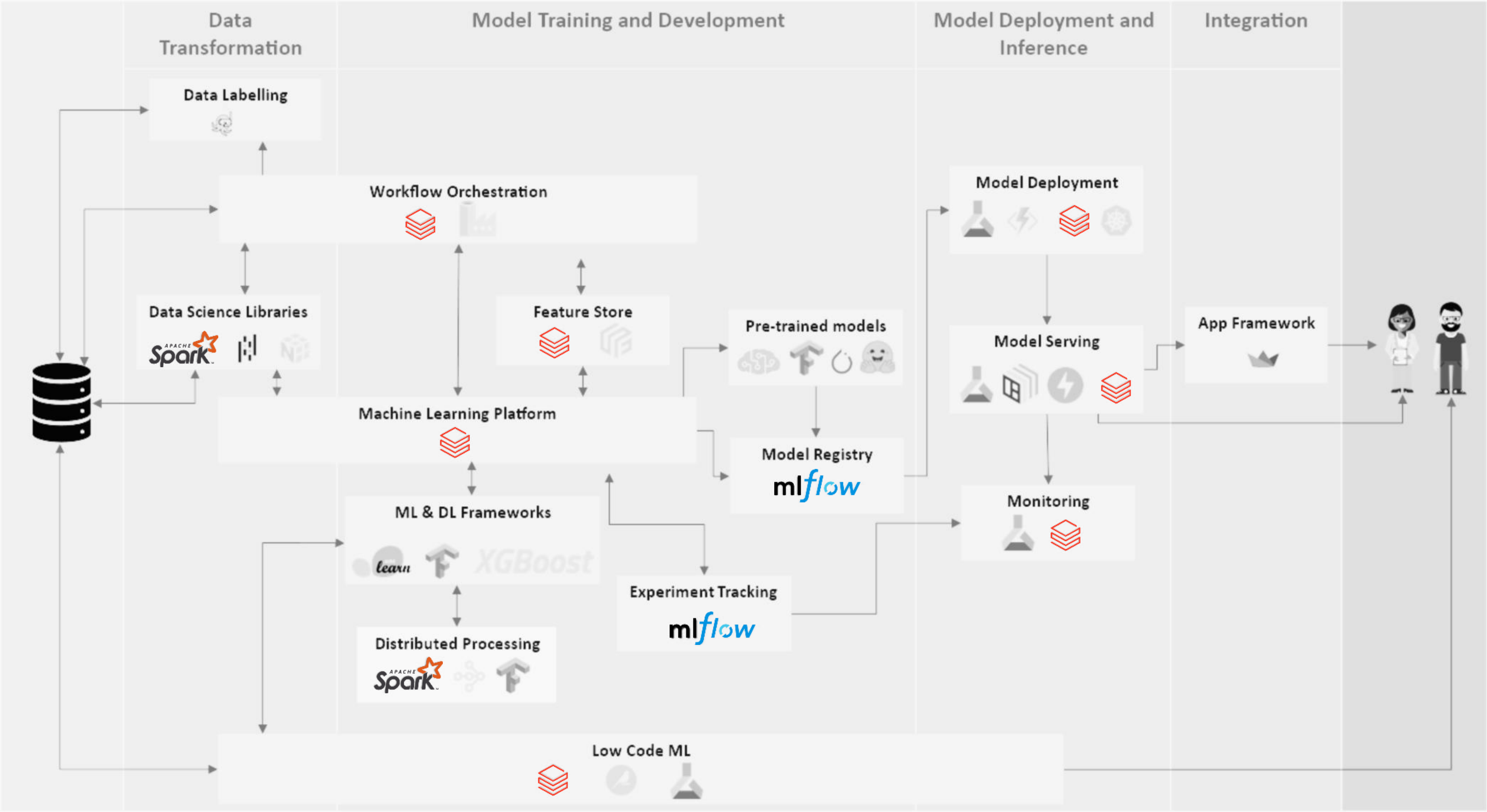
- Monitoring phase to ensure models remain accessible and accurate
- MLOps should monitor models for performance and data drift
- This can include tracking input and output data, performing drift analysis and automatic alerts

# MLOPS TOOLS



Data Transformation
Model Training and Development
Model Deployment and Inference
Integration
Data Labelling
Workflow Orchestration
Data Science Libraries
Spark
Feature Store
Pre-trained models
Machine Learning Platform
Model Registry
mlflow
ML & DL Frameworks
learn
XGBoost
Distributed Processing
Spark
Experiment Tracking
mlflow
Model Deployment
Model Serving
Monitoring
App Framework
Low Code ML

# MLOPS TOOLS IN DATABRICKS



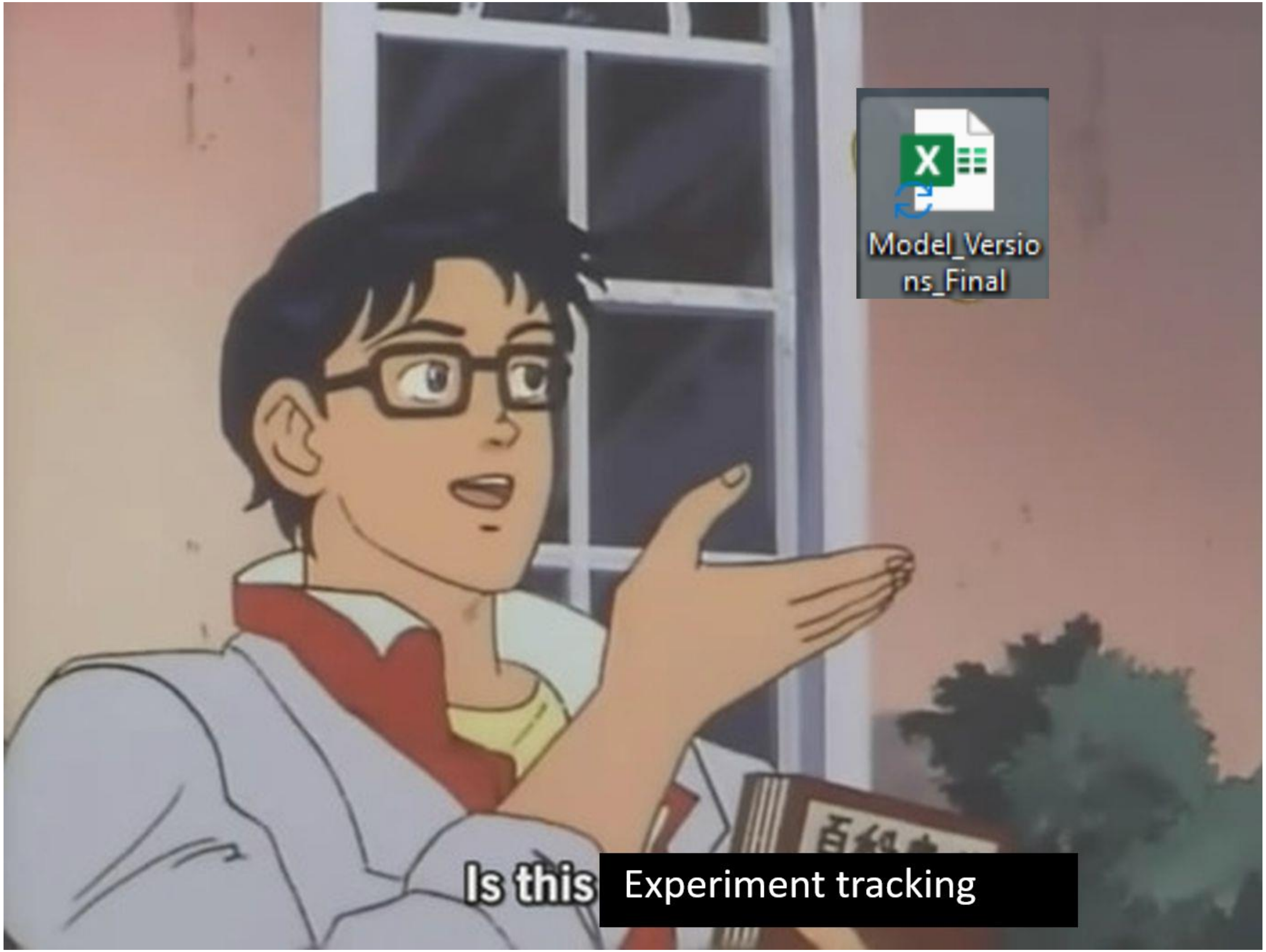
Explore

Evaluate

# MODEL AND EXPERIMENT TRACKING

Shap

mlflow


Model_Versions_Final
Is this Experiment tracking

# EXPLORE

- Which dataset was used?
- Which ML algorithm did we use to train the model?
- What hyperparameter values were used?
- What were the performance like?
- How can we convey all these info to our team easily?

Explore

mlflow

**Reproducible**

**Traceable**

# MLFLOW - COMPONENTS

**mlflow Projects**

Packaging format for reproducible runs on any compute platform

**mlflow Tracking**

Record and query experiments: code, metrics, parameters, artifacts, models

**mlflow Models**

General model format that standardizes deployment options

**mlflow Model Registry**

Centralized and collaborative model lifecycle management

Explore

# DATABRICKS MLFLOW

| | Open Source MLflow | Managed MLflow on Databricks |
|---|---|---|
| **Experiment Tracking** | | |
| MLflow tracking API | ✓ | ✓ |
| MLflow tracking server | Self-hosted | Fully managed |
| Notebooks integration | ✗ | ✓ |
| Workspace integration | ✗ | ✓ |
| **Reproducible Projects** | | |
| MLflow Projects | ✓ | ✓ |
| Git and Conda integration | ✓ | ✓ |
| Scalable cloud/clusters for project runs | ✗ | ✓ |
| **Model Management** | | |
| MLflow Model Registry | ✓ | ✓ |
| Model versioning | ✓ | ✓ |
| ACL-based stage transition | ✗ | ✓ |
| CI/CD workflow integrations | ✓ | ✓ |

Explore

# DATABRICKS MLFLOW

| | | |
|---|---|---|
| **Flexible Deployment** | | |
| MLflow Models | ✓ | ✓ |
| Built-in batch inference | ✗ | ✓ |
| Built-in streaming analytics | ✗ | ✓ |
| **Security and Management** | | |
| High availability | ✗ | ✓ |
| Automated updates | ✗ | ✓ |
| Role-based access control | ✗ | ✓ |

Integrates with Feature Store
Facilitates Databricks Serving Endpoints

Explore

# EVALUATE

Evaluate

- Built in metrics:
  - **Regressor models:** example_count, mean_absolute_error, mean_squared_error, root_mean_squared_error, sum_on_target, mean_on_target, r2_score, max_error, mean_absolute_percentage_error.
  - **Binary classifiers:** true_negatives, false_positives, false_negatives, true_positives, recall, precision, f1_score, accuracy_score, example_count, log_loss, roc_auc, precision_recall_auc.
  - **Multiclass classifiers:** accuracy_score, example_count, f1_score_micro, f1_score_macro, log_loss
- Custom metrics
- Explainability metrics
  - SHAP
- Fairness metrics
  - Fairlearn

# DEMO

- Mlflow
  - Tracking experiments
  - Logging parameters
  - Logging metrics

# INTERPRETABILITY

Evaluate

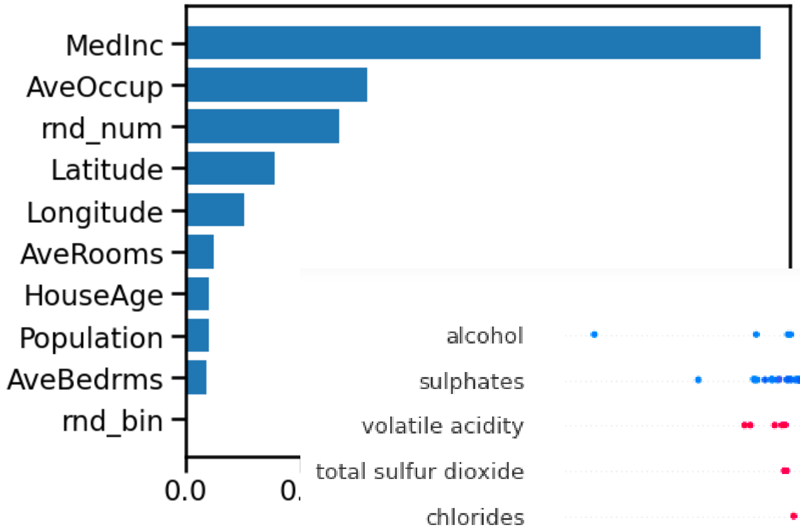
You're fired!

WHY!

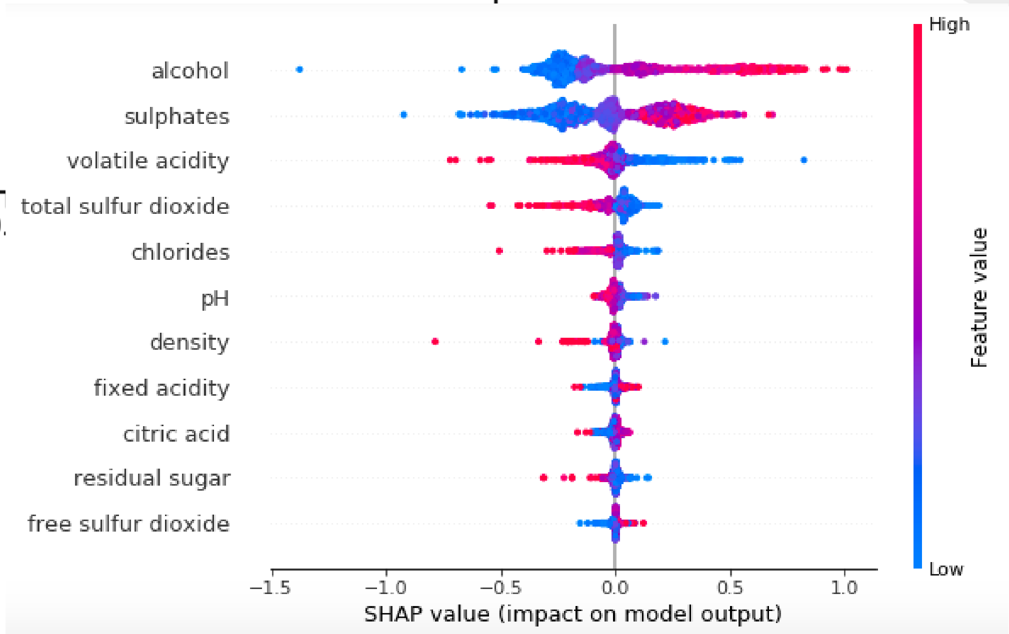
Computer says so

# HOW CAN YOU INTERPRET MODEL

- Feature Importance
  - Ranking features in order of importance for determining outcome




- SHAP Values
  - **Sh**apely **A**dditive ex**P**lanations
  - Visualisations of how features interact with the outcome

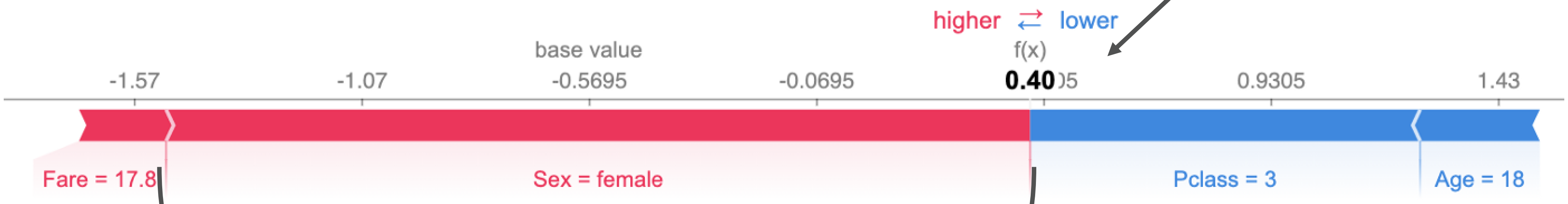



Evaluate

# SHAP VALUES (SHAPLEY ADDITIVE EXPLANATIONS)

Evaluate

Final model prediction




The larger the width, the more of a contribution it had on the output.

Pushes the score higher

Pushes the score lower

# DEMO

- SHAP!

www.advancinganalytics.co.uk

# FEATURE STORES


Storing training data in a table
Feature Stores

# WHAT ARE FEATURES?

Develop

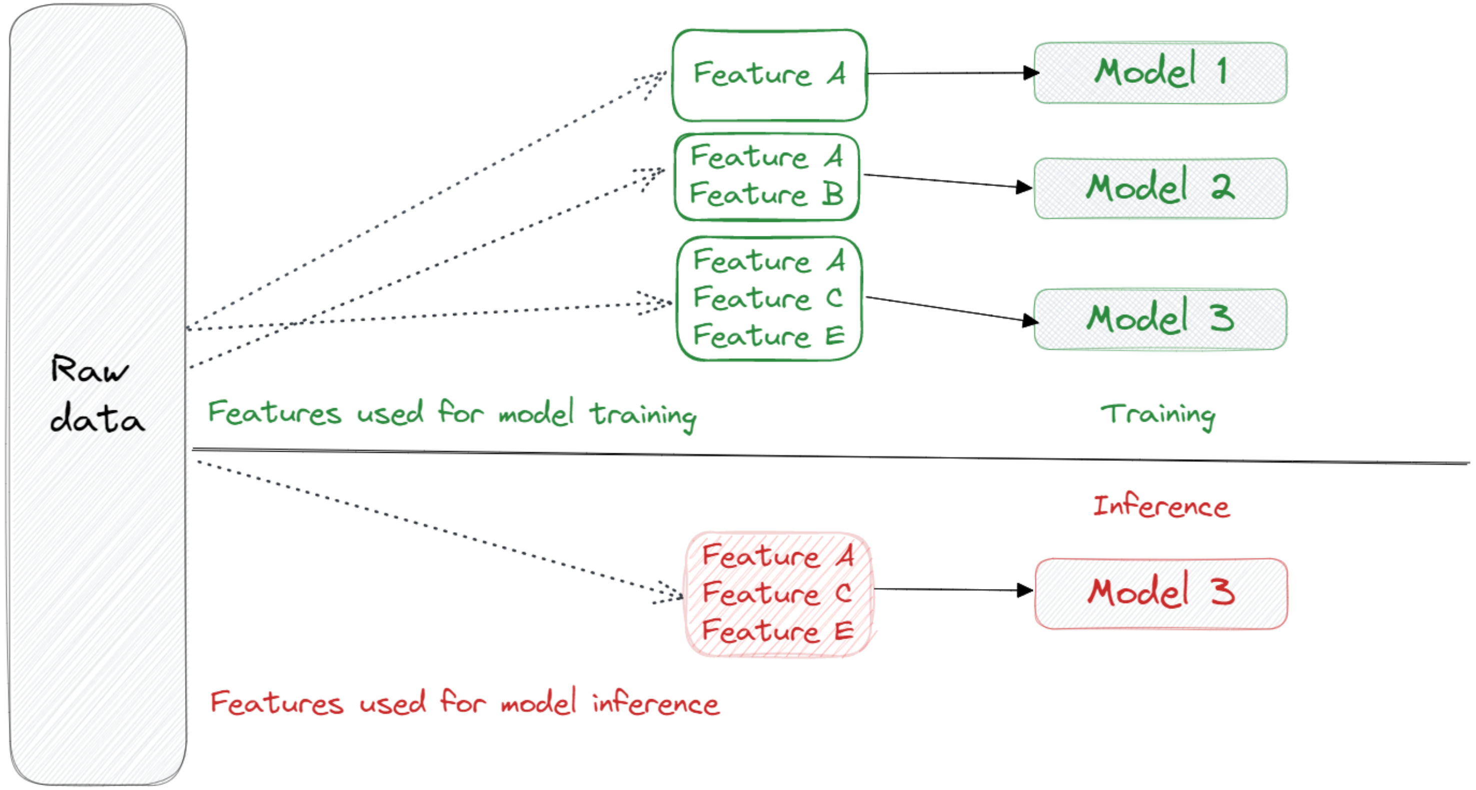
A "**feature**" refers to the entire column in the dataset

| Transaction_id | in_foreign_country | size_compared_to_avg_transaction | fraud? |
|---|---|---|---|
| 7485 | False | 0.8x | False |
| 46854 | True | 21.2x | True |
| 3521 | True | 1.1x | False |

A "**feature value**" refers to a single value of a feature column

# WHAT IS A FEATURE STORE?

- Primary purpose to store and organise features
- Relatively new concept




Develop

# FEATURE STORE CONCEPTS

- Registry – Central interface for managing feature definitions + metadata
- Offline Store – Store large volumes of feature data used train and test
- Online Store – Low latency database for real-time inference

Develop

# WHY DO YOU NEED A FEATURE STORE?

- Easily reuse new features
- Easily explore features
- Data pipelines can be shared across both training and serving
- Constant features across teams
- Provides lineage
- Feature tracking

Develop

# DATABRICKS FEATURE STORE

- Backed by Delta Lake
- Feature Store Library
- Feature Store UI
- Offline Store
- Online Store Compatability
- Training Set Functionality

Develop

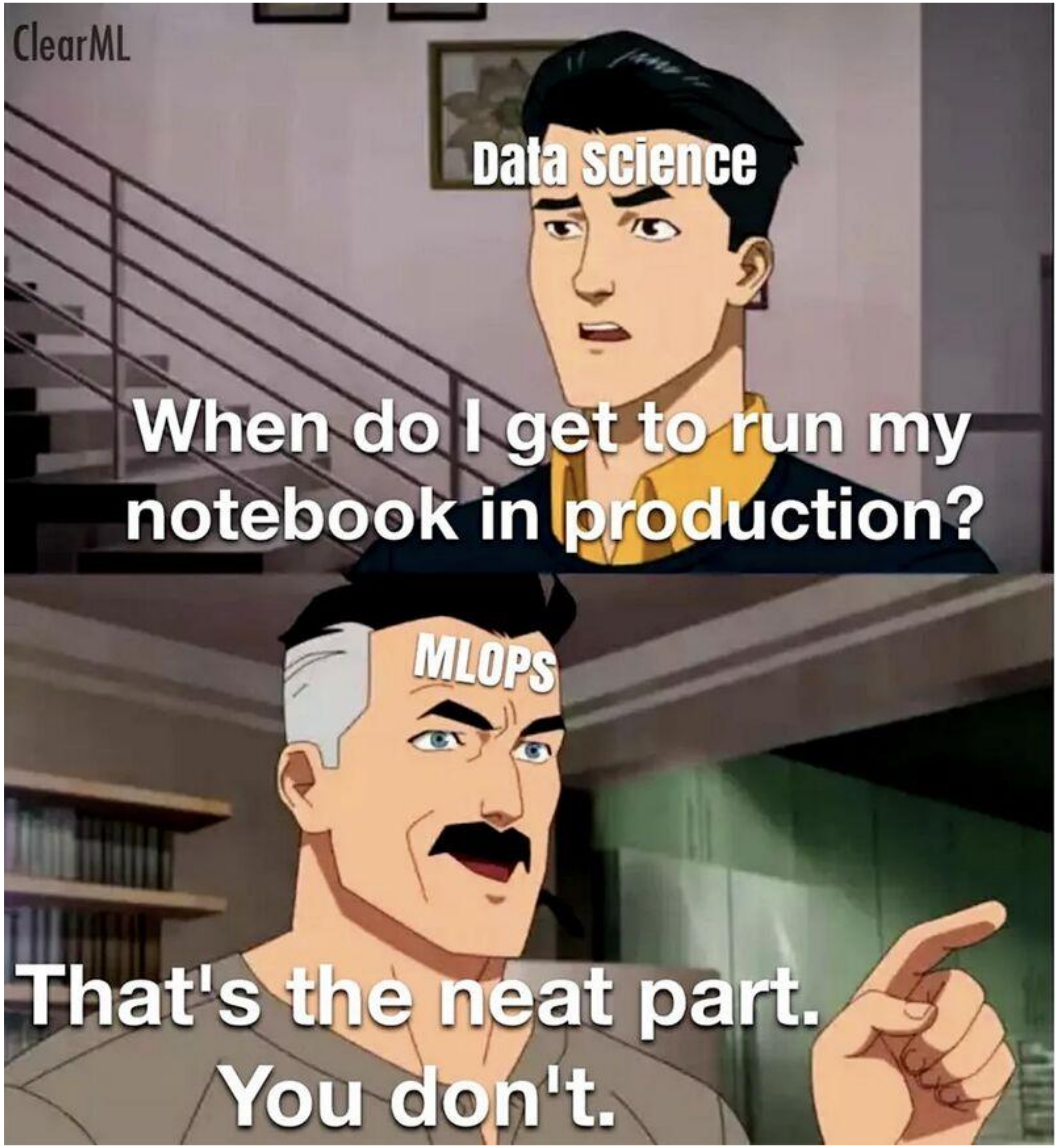
# DEMO

- Feature Store
  - –Create a feature table
  - –Save features
  - –Create a training dataset

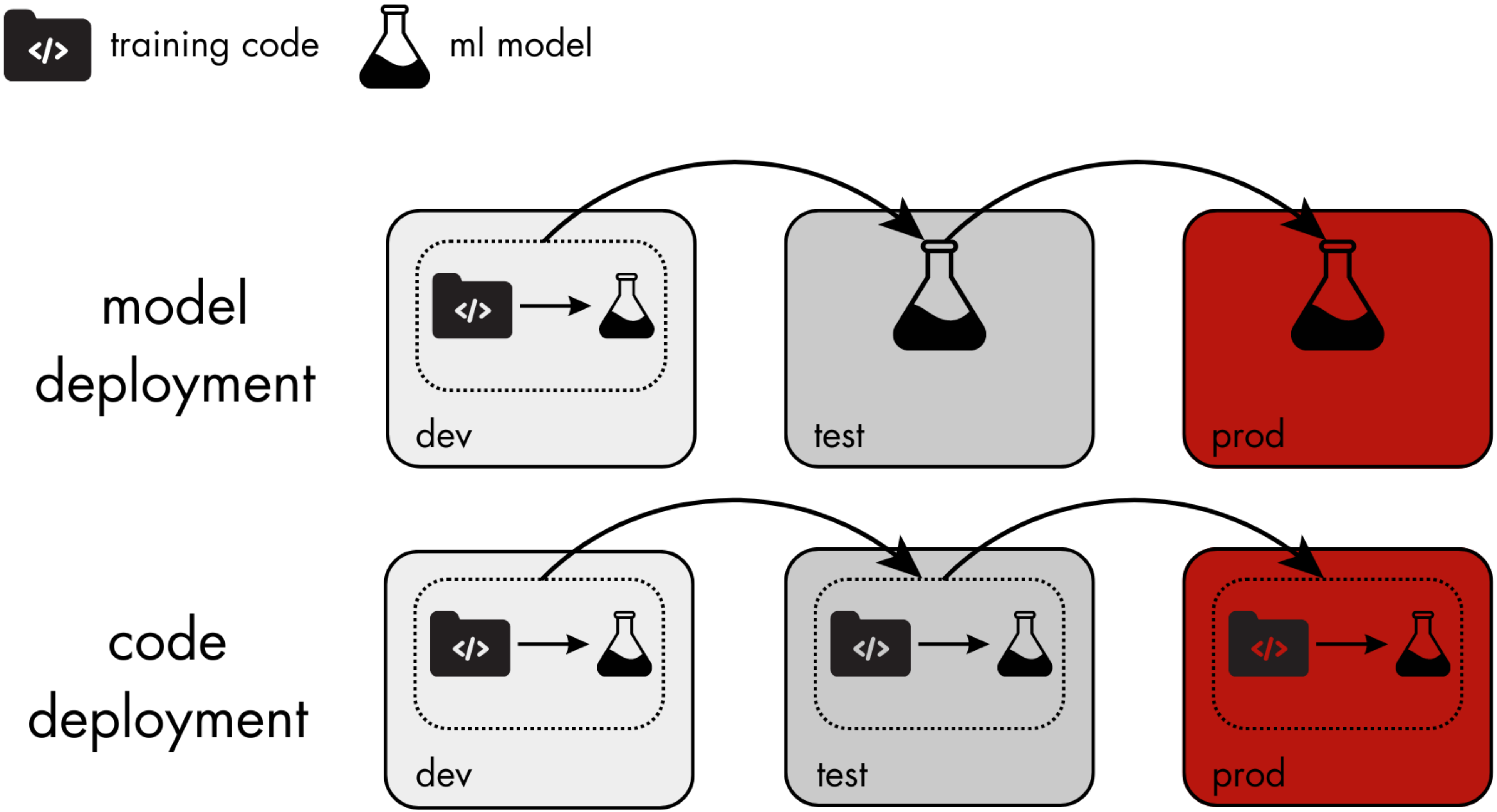
# BREAK!

Package

# DEPLOYMENT STRATEGIES

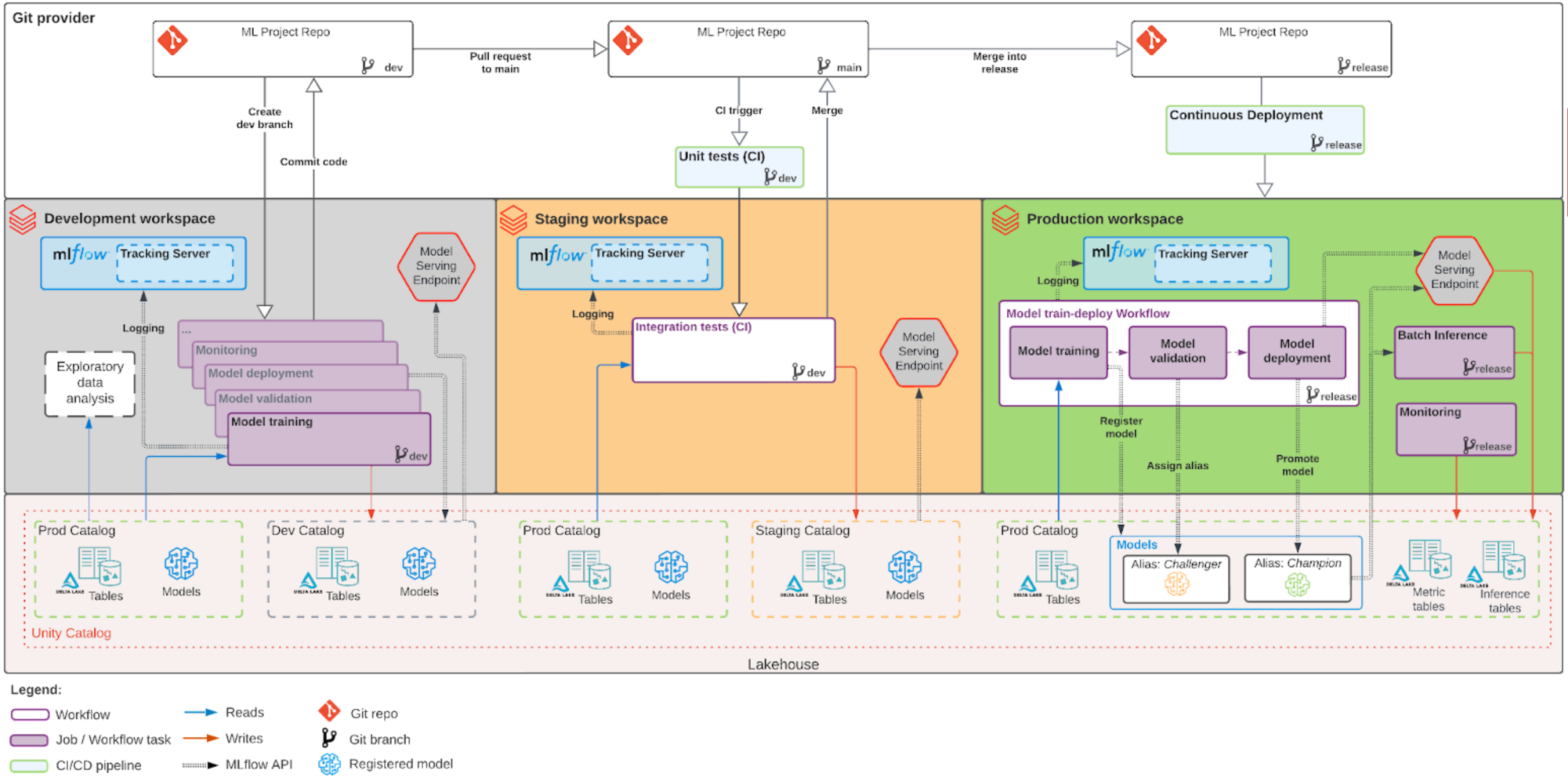

ClearML
Data Science
When do I get to run my notebook in production?
MLOPS
That's the neat part. You don't.

# CODE-FIRST VS MODEL-FIRST




Package

# CODE FIRST DEPLOYMENT PATTERN




Package

Deploy

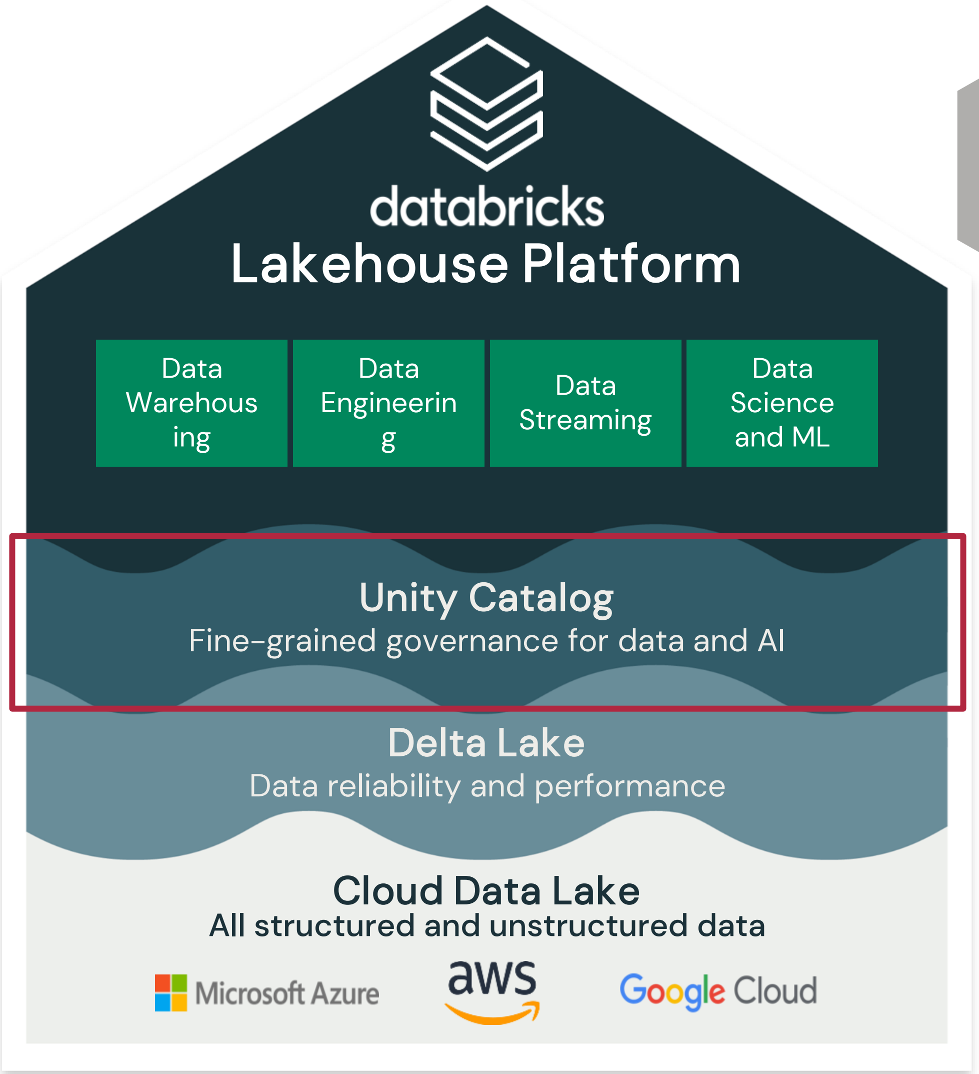
# MODEL DEPLOYMENT


I DON'T KNOW
HOW TO DEPLOY A MODEL
AND AT THIS POINT
I'M TOO AFRAID TO ASK
imgflip.com

# DATA BRICKS UNITY CATALOG

## Unified governance for all data and AI assets

- Centralized governance for data and AI
- Built-in data search and discovery
- Performance and scale
- Automated lineage for all workloads
- Integrated with your existing tools



# REGISTER MODELS IN UNITY CATALOG

- Reference models using their three-level naming
  - <catalog>.<schema>.<model>

- Models in Unity Catalog is compatible with the MLflow Python client.

- To upgrade ML workflows to target Unity Catalog, simply:
  - Configure MLflow client to target Unity Catalog
    - `import mlflow`
    - `mlflow.set_registry_uri("databricks-uc")`

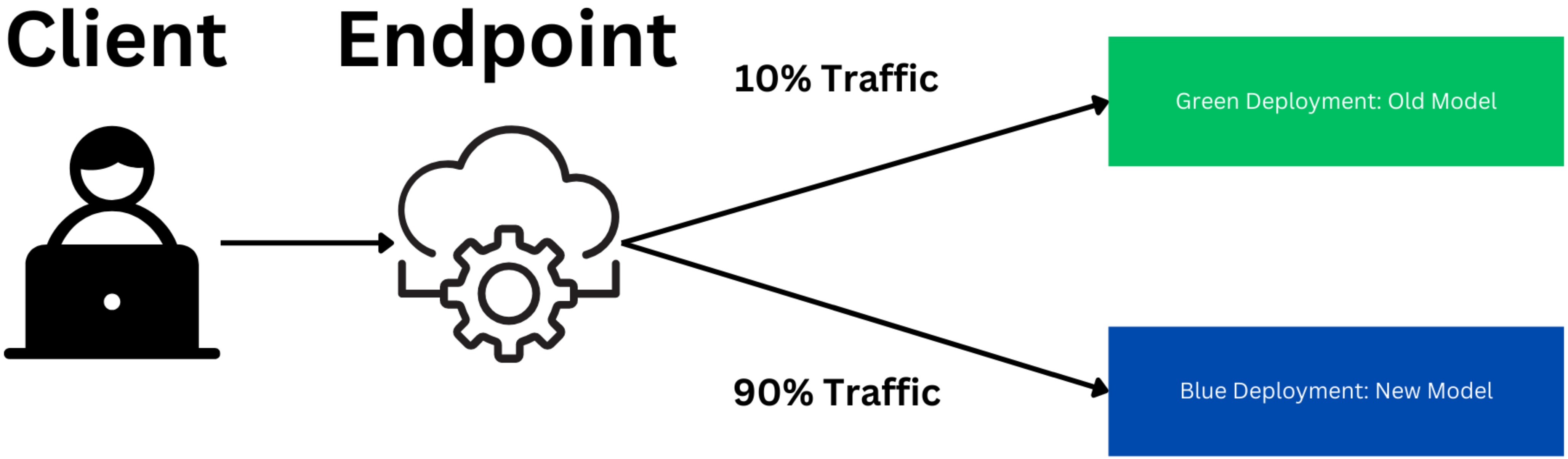
Deploy

# MODEL SERVING

- Real time scoring provides near real-time results on small records
  - Fraud Detection
  - Predictive Maintenance
  - Recommender systems

- Batch scoring provides results for a large volume of records in a single operation
  - Churn Prediction
  - Credit Risk Analysis
  - Forecasting

# GREEN BLUE DEPLOYMENT




Deploy

# ALIASES AND MODEL VERSIONS IN UC

Deploy

Overview | Details | Permissions

**Description**

Add description

**Versions**

| Version | Time registered | Tags | Aliases | Registered by | Comment |
|---|---|---|---|---|---|
| Version 3 | 2023-08-16 10:27:10 | | | eric.golinko@databricks.com | |
| Version 2 | 2023-07-17 22:34:35 | | | eric.golinko@databricks.com | |
| Version 1 | 2023-07-17 22:29:23 | | | eric.golinko@databricks.com | |

Overview | Details | Permissions

**Description**

Add description

**Versions**

| Version | Time registered | Tags | Aliases | Registered by | Comment |
|---|---|---|---|---|---|
| Version 3 | 2023-08-16 10:27:10 | | @ Champion | eric.golinko@databricks.com | |
| Version 2 | 2023-07-17 22:34:35 | | | eric.golinko@databricks.com | |
| Version 1 | 2023-07-17 22:29:23 | | | eric.golinko@databricks.com | |

# DATABRICKS SERVING/SERVERLESS ENDPOINTS

- Serverless doesn’t mean no servers!

- The cloud providers take care of underlying infrastructure

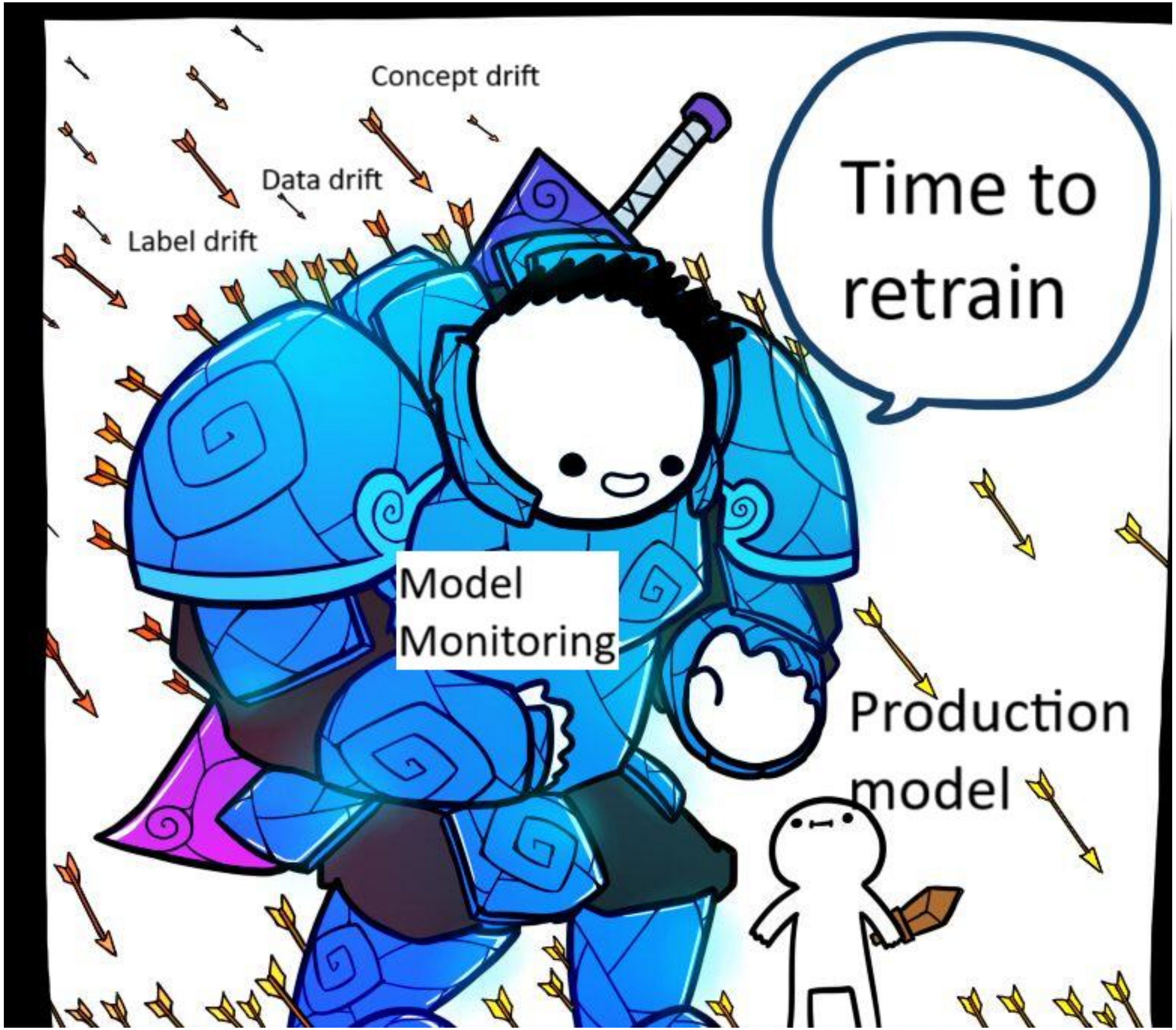
- Scalable
- Cost-effective
- Flexible
- Easy to deploy & manage

Deploy

# DEMO!

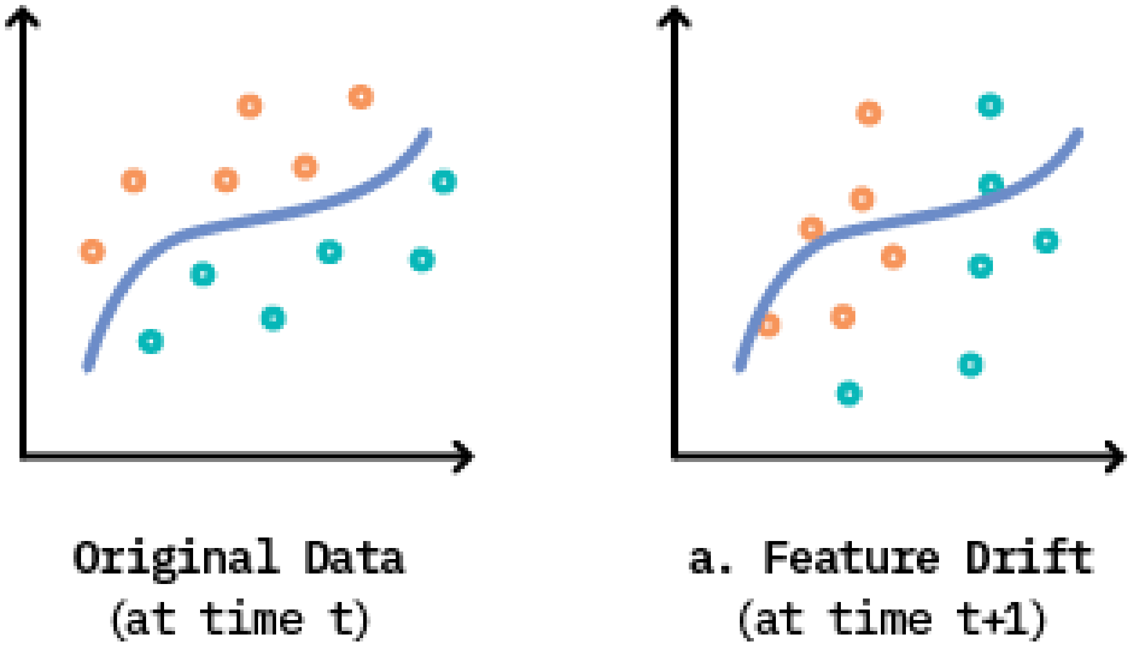
- Databricks Endpoints
  - –Deploying a model
  - –Blue/Greeb Deployment
  - –Calling the endpoint

# MODEL MONITORING


Concept drift
Data drift
Label drift
Time to retrain
Model Monitoring
Production model

# MODEL DRIFT

- Data drift
- Label drift
- Prediction drift
- Concept drift




Monitor

# CHI-SQUARED TEST

Monitor

Bounded between (0, ∞)

$$\chi^2(P|Q) = \sum_{x \in X} \frac{(P(x) - Q(x))^2}{Q(x)}$$

**Libraries**

Scipy

# K-L DIVERGENCE (RELATIVE ENTROPY)

Bounded between (0, ∞)

Monitor

First Probability distribution

$$D_{KL}(P|Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

Second Probability distribution

**Libraries**
Divergence
Scipy

# POPULATION STABILITY INDEX (PSI)

Monitor

Bounded between (0, ∞)

Symmetrical

First Probability distribution

$$D_{PSI}(P|Q) = \sum_{x \in X} (P(x) - Q(x)) \log\left(\frac{P(x)}{Q(x)}\right)$$

Second Probability distribution

- PSI < 0.1: no significant population change
- 0.1 < PSI < 0.2: moderate population change
- PSI > 0.2: significant population change

# JS (JENSEN-SHANNON) DIVERGENCE

Bounded between (0, 1)

**Monitor**

First Probability distribution

$$D_{JS}(P|Q) = \frac{1}{2}\sum_{x \in X} P(x)\log\left(\frac{P(x)}{M(x)}\right) + Q(x)\log\left(\frac{Q(x)}{M(x)}\right)$$

Second Probability distribution

$$M(x) = \frac{1}{2}(P(x) + Q(x))$$

**Libraries**

Divergence

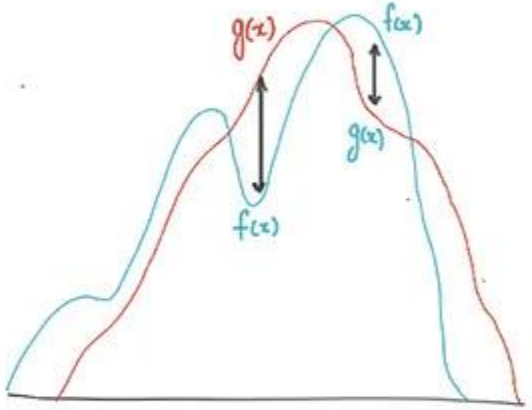
Scipy

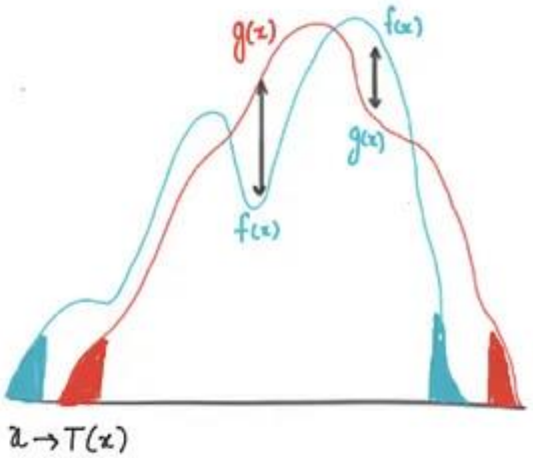
# WASSERSTEIN DISTANCE

First cumulative distribution

$$W(\mu, \nu) = \left( \int_R \left| F_1^{-1}(q) - F_2^{-1}(q) \right|^p dq \right)$$

Monitor




**Libraries**

Scipy

Wasserstein Distance, Contraction Mapping, and Modern RL Theory | by Kowshik chilamkurthy | Medium

# WHAT DRIFT METRICS DO I USE?

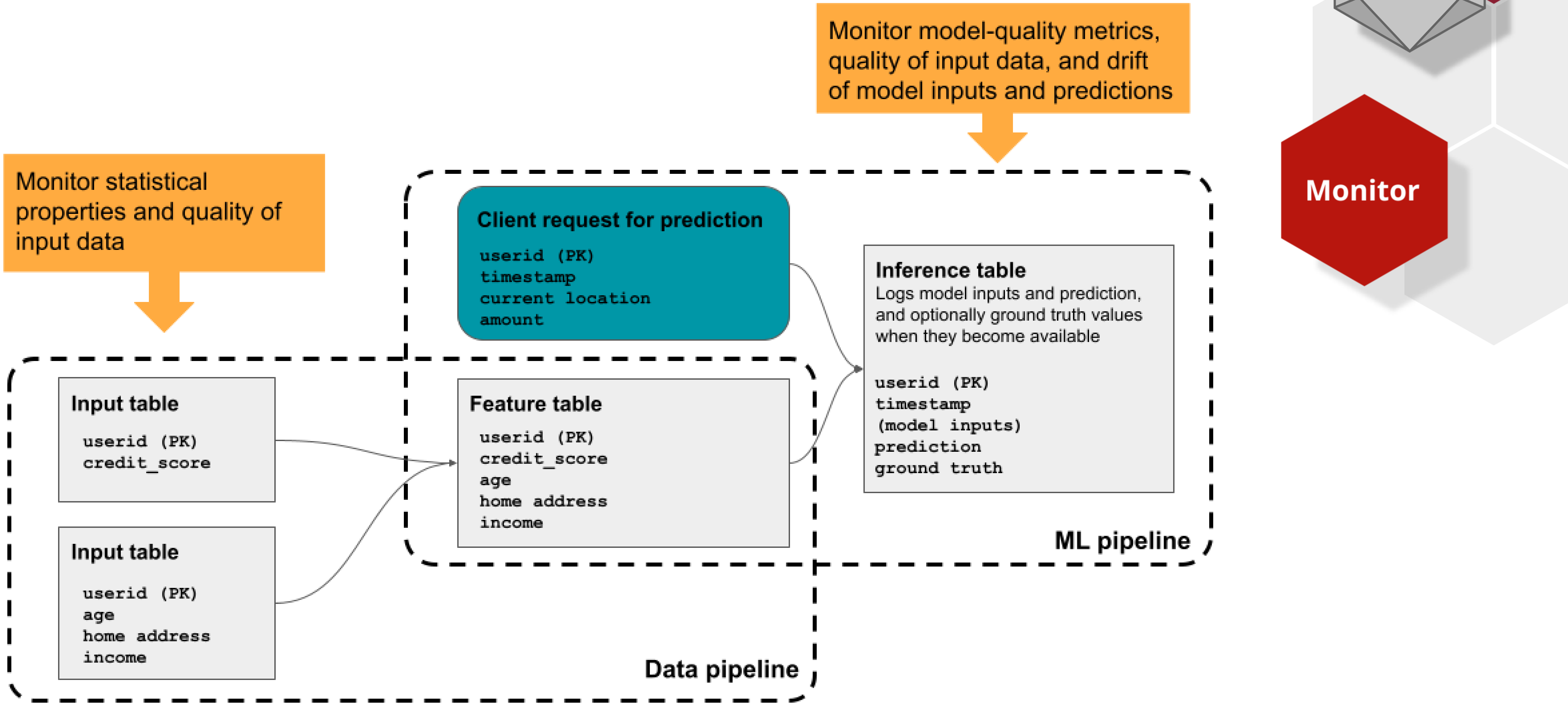
| Name | Pros | Cons |
| --- | --- | --- |
| Chi-Squared Test | Simple | Invalid for small frequencies within a category. |
| K-L Divergence | Drift is interpretable (excess surprise of second distribution) | Non-symmetric, doesn’t like zero bins |
| PSI | Symmetric, standard indicator values | Unreliable when frequencies approach zero |
| JS Divergence | Smoother, bounded version of K-L Divergence, can handle zero bins | Baseline isn’t completely static |
| Wasserstein Distance | Accounts for the distance along the variable space as well as the displacement value | Complex |

Monitor

# DEMO

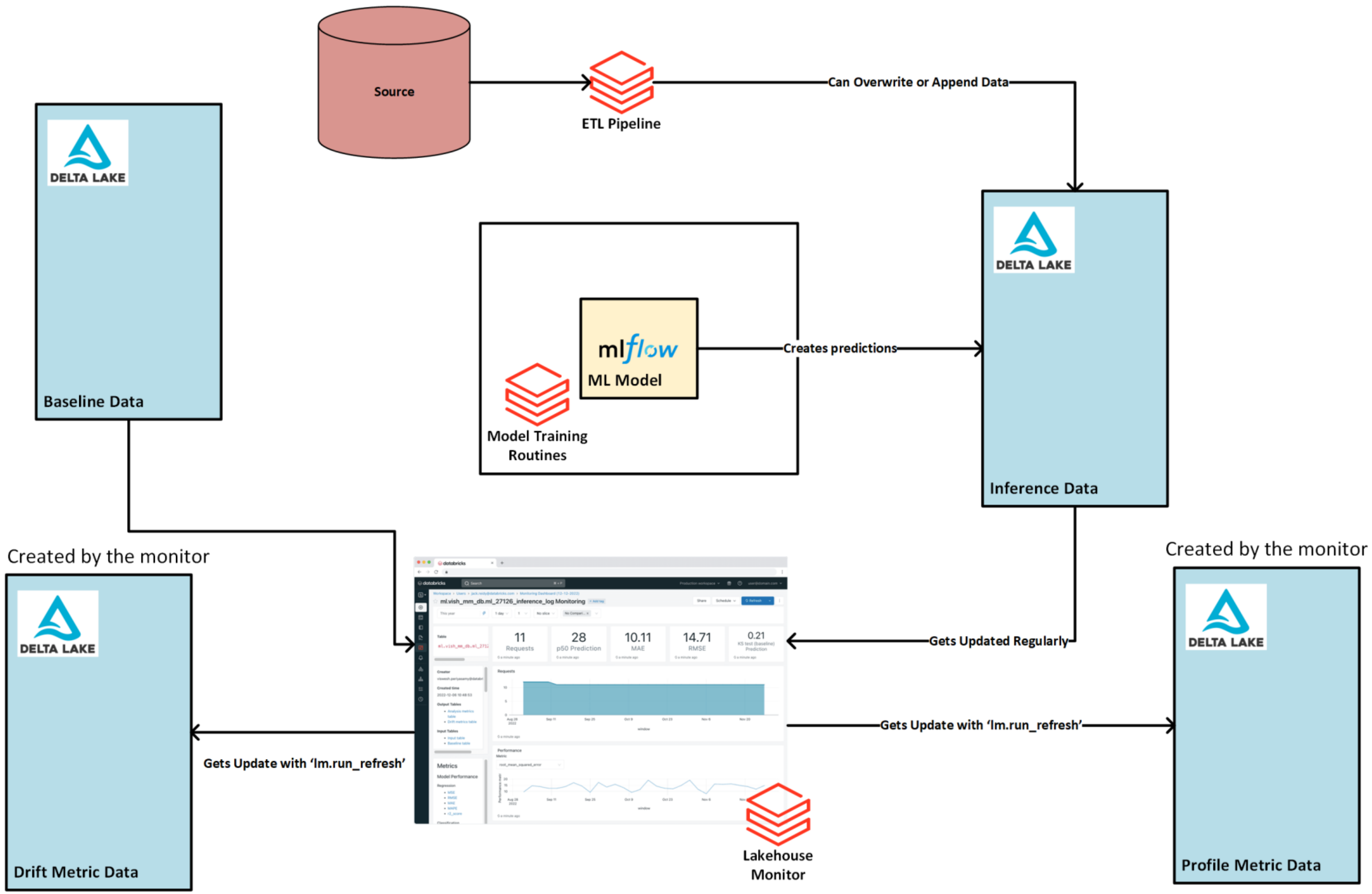
- Model Drift Metrics
  - Implementing metrics in databricks

# LAKEHOUSE MONITORING



# LAKEHOUSE MONITORING

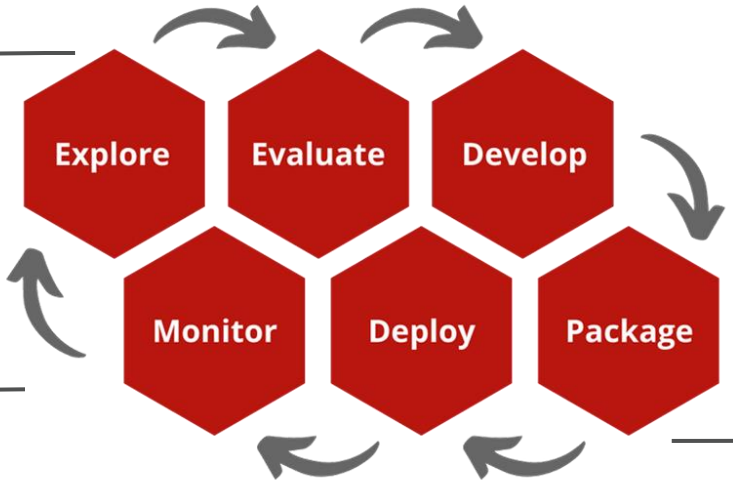
Monitor



# CONCLUSION

# DATABRICKS AS AN END-TO-END PLATFORM

Collaborative Workspace




Databricks Feature Store

Custom Drift Notebooks

Code vs Model Deployment

Databricks Serverless Endpoints

# THANKS FOR LISTENING!