Microsoft Azure

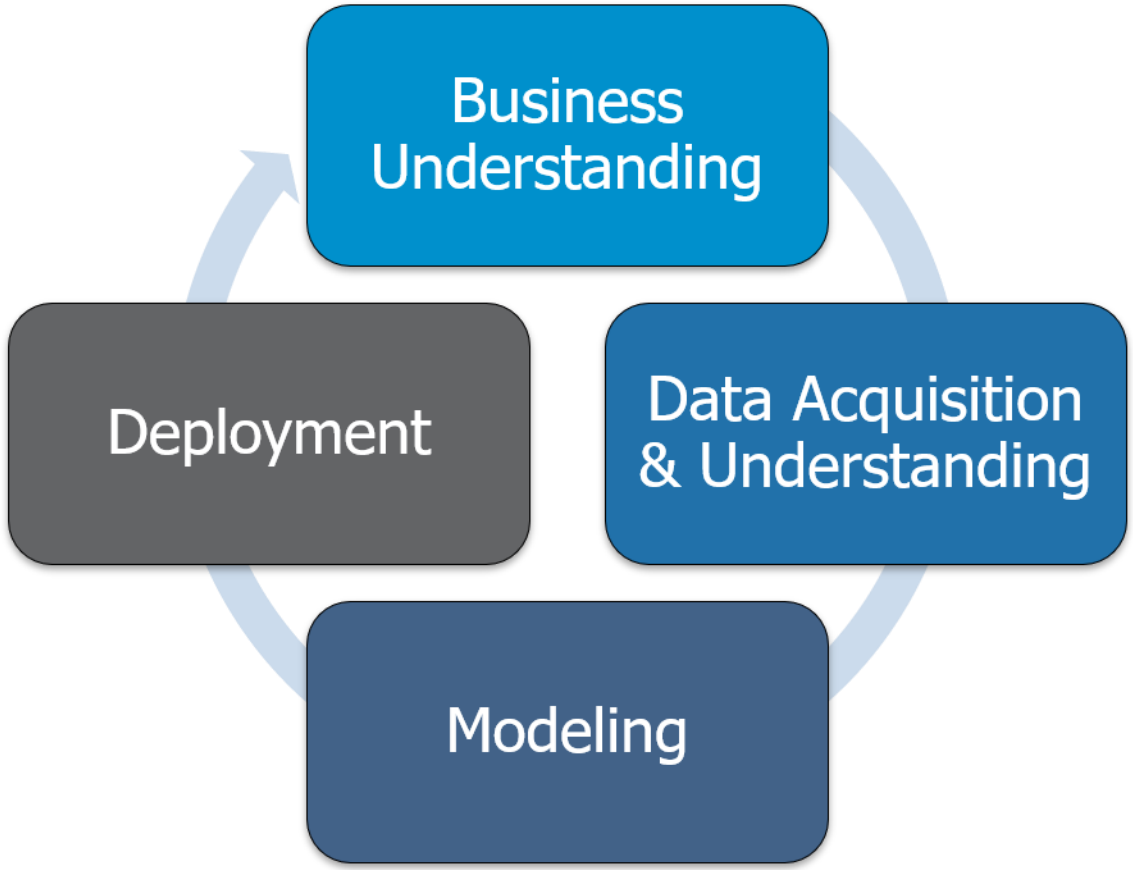
# Data Science & Machine Learning

## Developing Predictive & Prescriptive Analytic Services

**Jon Tupitza**
Sr. Cloud Solution Architect – Data & AI – Azure Data Scientist
*Jon.Tupitza@Microsoft.com*

# The Microsoft: Team Data Science Process

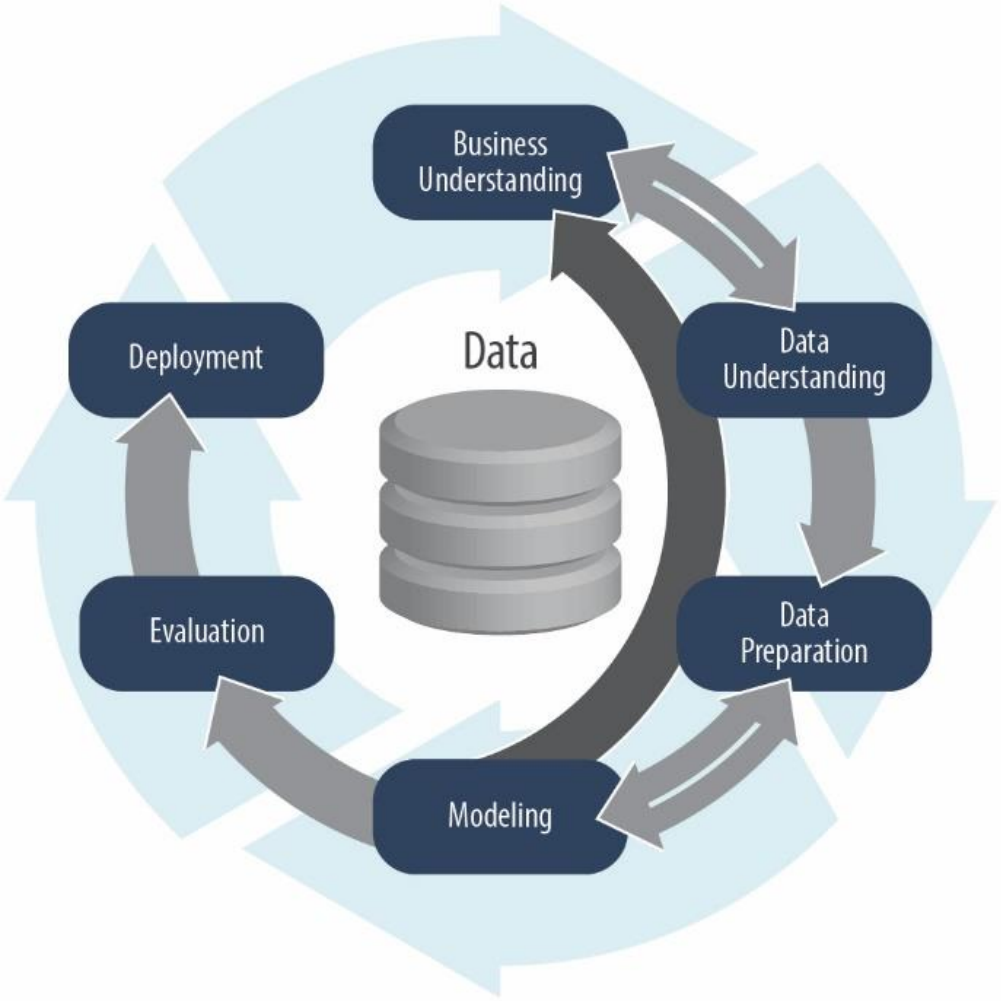
## Iterative & Exploratory: Largely Based on Conducting Experiments




| | |
|---|---|
| Business Understanding | • Identify the Problem Domain<br>• Identify the Solution Scenario |
| Acquire & Understand Data | • Load, Prepare & Explore Data<br>• Identify Influential Features |
| Develop Machine Learning Models | • Select & Engineer Features<br>• Train, Evaluate & Tune Models |
| Deployment | • Publish Models as Webservices<br>• Consume Models Visually and Programmatically |

# CRISP-DM: Cross-Industry Standard Process-Data Mining

First Introduced in 1996!




**Business Understanding**
- Identify the Problem Domain
- Identify the Solution Scenario

**Data Understanding**
- Load and Explore Data
- Identify Influential Features

**Data Preparation**
- Remove Duplicates & Nulls
- Impute Missing Values
- Select & Engineer Features

**Modeling**
- Train Models Using a Variety of Algorithms
- Tune Hyper-parameters

**Evaluation**
- Test Models' Performance & Predictive Power
- Cross-Validate to Appraise Goodness-of-Fit
- Select Most Effective Model for Deployment

**Deployment**
- Publish Models On-premises or in the Cloud
- Consume Models Visually & Programmatically

# Machine Learning: Key Topics & Activities

Azure Machine Learning Features that Enable Machine Learning Productivity

**Exploratory Data Analysis**

- Univariate Analysis
- Feature Engineering
- Feature Importance

**Explainability (Responsible AI)**

- Partial Dependence
- Out-of-Sample Accuracy (Lift Chart)
- Sensitivity Analysis (Feature Impact)

**Training and Evaluation**

- Model Selection
- Cross-Validation
- Hyperparameter Tuning

**Deployment & Inferencing**

- Register Model
- Register & Deploy Image
- Monitor Model Predictions

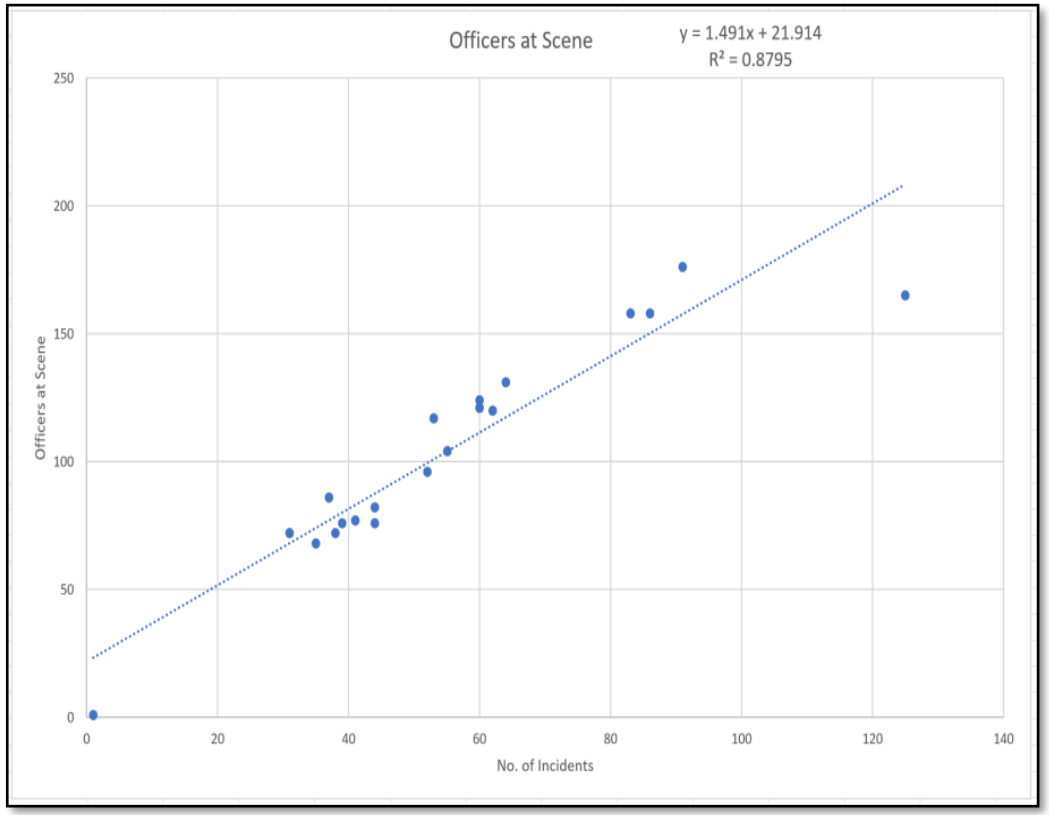
# Modeling

## Training & Evaluating Machine Learning Models

# Machine Learning Process: Algorithms

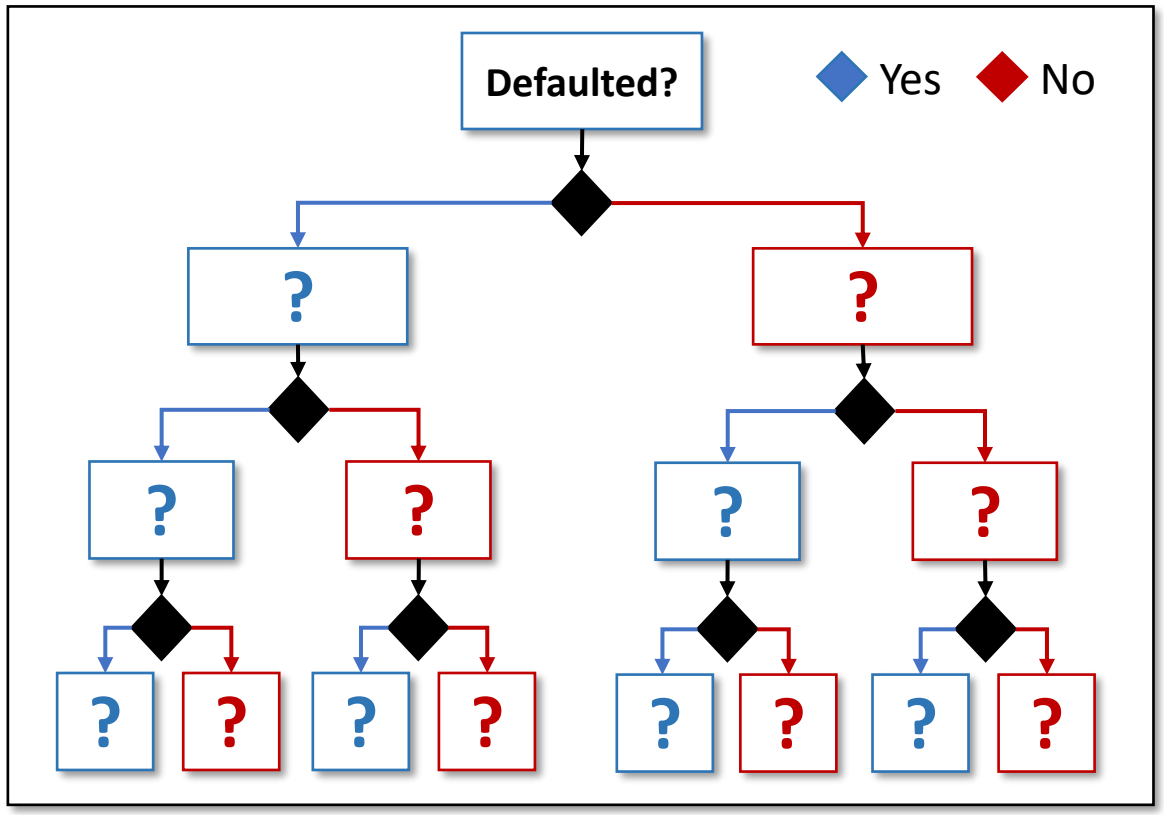
The Type of Question Typically Drives the Choice of Modelling Algorithms

## Linear Models (e.g., Regression)




***How do X and y Correlate?***

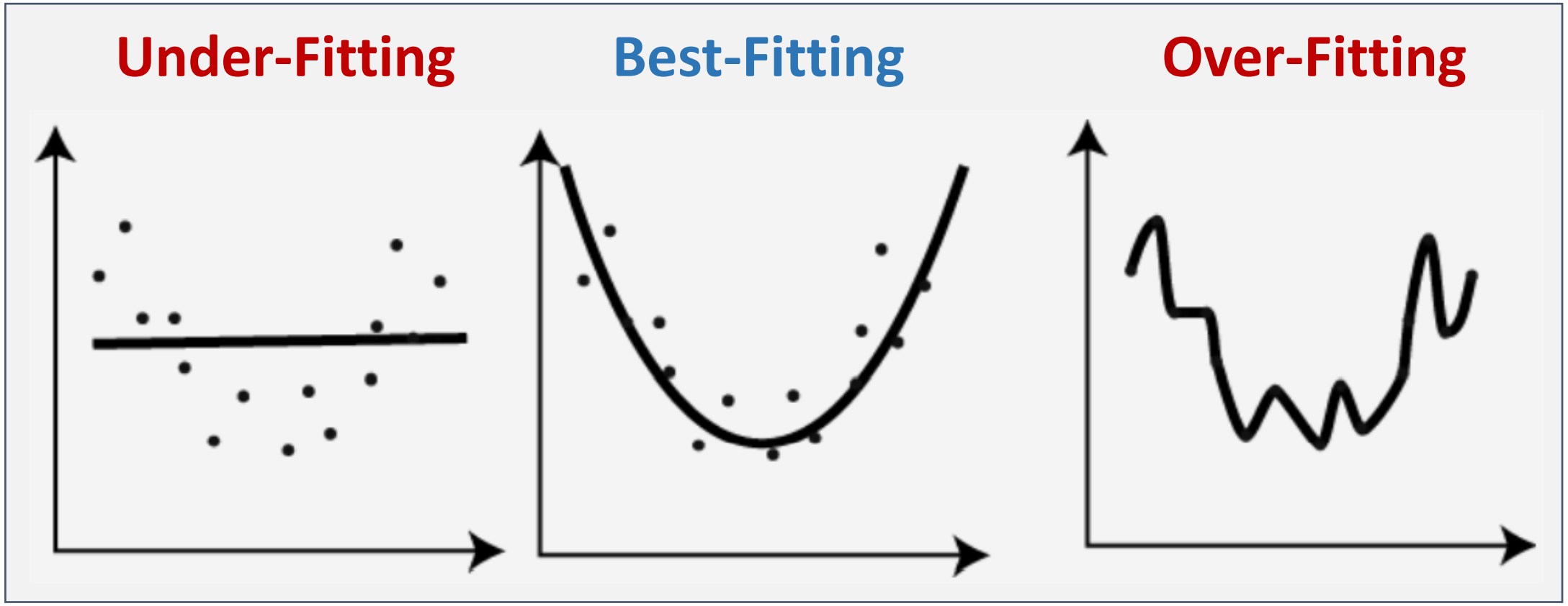
## Tree-Based Models (e.g., Classification)




***The outcome of a series of decisions?***

# Machine Learning Process: Generalization

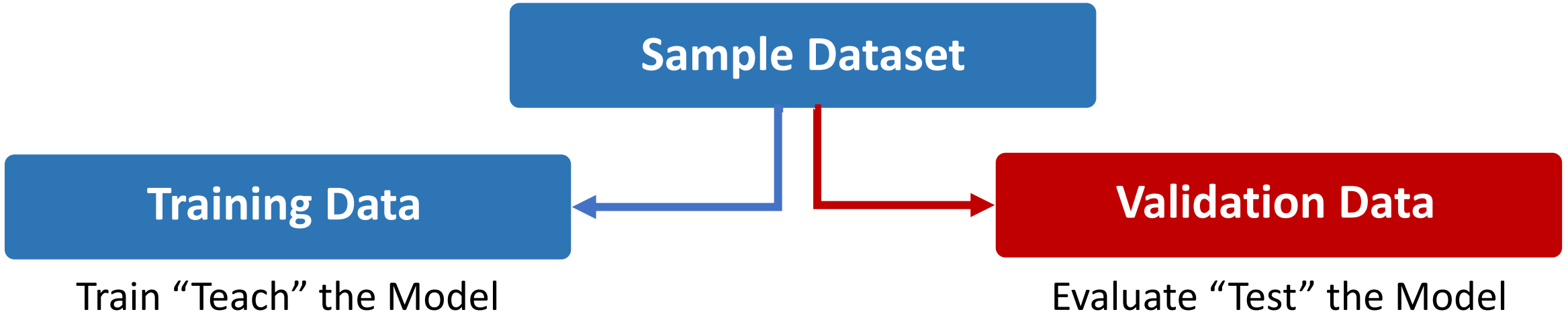
Creating ML Models that Predict Accurately When Exposed to Data “In the Wild”

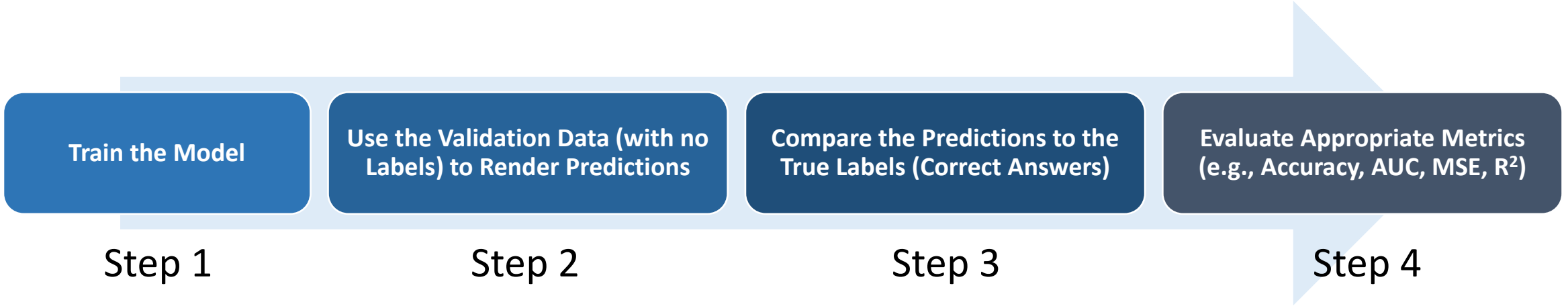



***So then… How do we achieve the best fit?***

# ML Model Validation: Train/Test Split

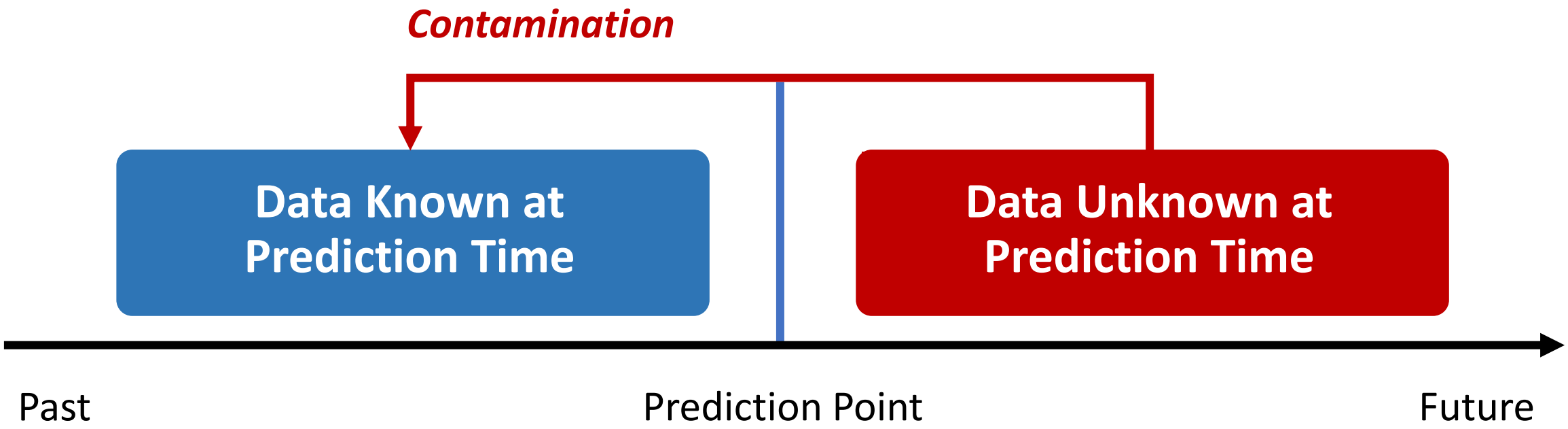
## Data Partitioning: Train on 70% of the Data and then Test Using the Remaining 30%



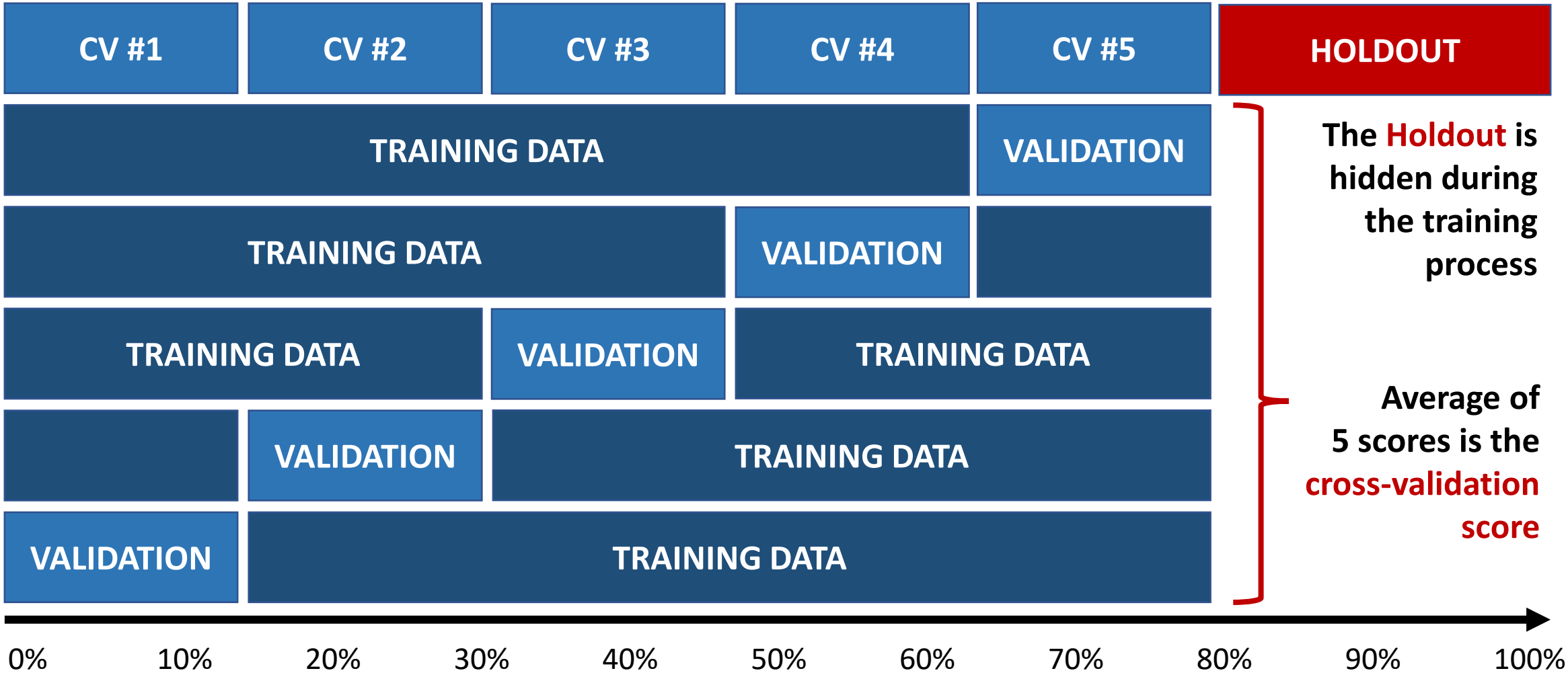
# Data Understanding: Target Leakage

Data Not Known at the Time of Prediction – *Cheating*



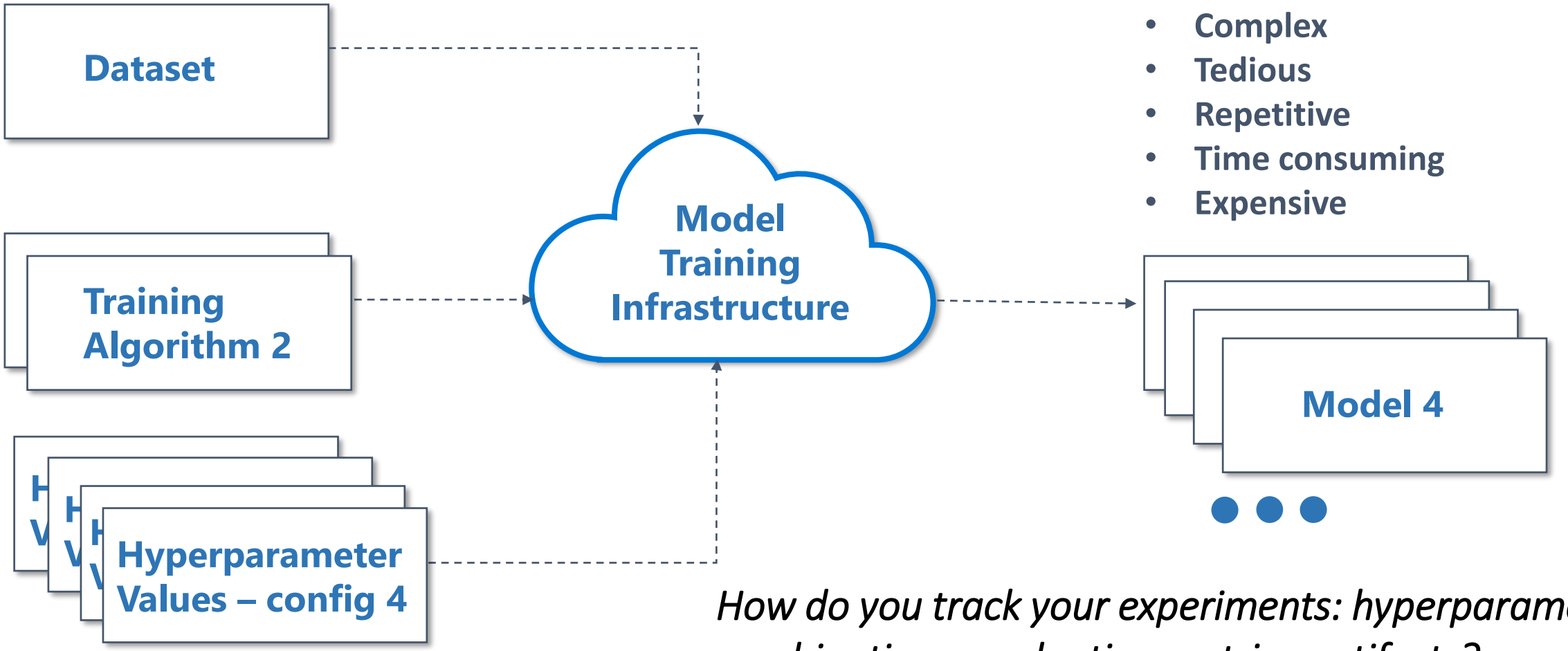
# ML Model Validation: Cross-Validation

Key to Determining the Best Hyper-Parameter Combinations & Selecting the Best Model



# Azure Machine Learning: Hyperparameter Tuning

The typical “Manual” Approach to Hyperparameter Tuning

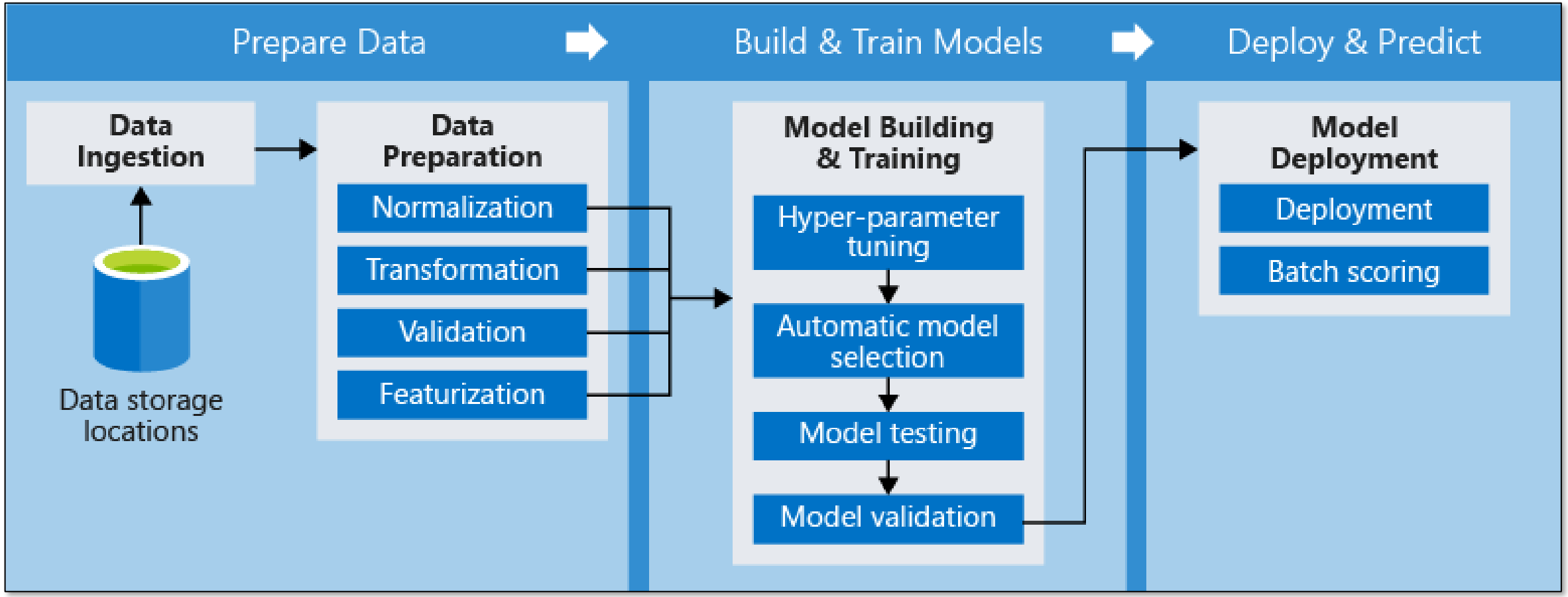



- Complex
- Tedious
- Repetitive
- Time consuming
- Expensive

*How do you track your experiments: hyperparameter combinations, evaluation metrics, artifacts?*

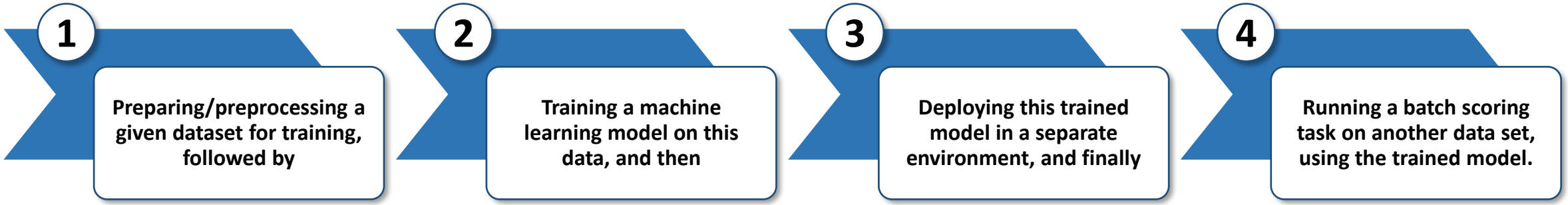
# Machine Learning Pipelines: Motivation

The “Manual” Activities Involved with Implementing Machine Learning Services



# Machine Learning Pipelines: Workflows

ML Pipelines Combine Multiple Activities into a Single Configurable Entity




**Azure's Machine Learning Pipelines:**

- Enable you to combine multiple steps into one configurable workflow
- Provide a consistent and reproducible mechanism for building, evaluating, deploying, and running ML systems.
- Support reuse by allowing multiple users to share and reuse workflows

# Machine Learning Pipelines: Steps

Individual Units of Execution that Implement Activities in the Machine Learning Process

| | |
|---|---|
| **PythonScriptStep**: | Adds a step to run a Python script in a Pipeline. |
| **AdlaStep**: | Adds a step to run U-SQL script using Azure Data Lake Analytics. |
| **DataTransferStep**: | Transfers data between Azure Blob and Data Lake accounts. |
| **DatabricksStep**: | Adds a Databricks notebook as a step in a Pipeline. |
| **HyperDriveStep**: | Creates a Hyper Drive step for Hyper Parameter Tuning in a Pipeline. |
| **AzureBatchStep**: | Creates a step for submitting jobs to Azure Batch |
| **EstimatorStep**: | Adds a step to run Estimator in a Pipeline. |
| **MpiStep**: | Adds a step to run a MPI job in a Pipeline. |
| **AutoMLStep**: | Creates a AutoML step in a Pipeline. |

# Demo: Train Models using Azure ML Services

Train Models in Azure ML using Jupyter Notebooks from an Azure Compute Cluster

# Explainability

## Interpreting & Understanding Machine Learning Models

# Explainability: Model Interpretability

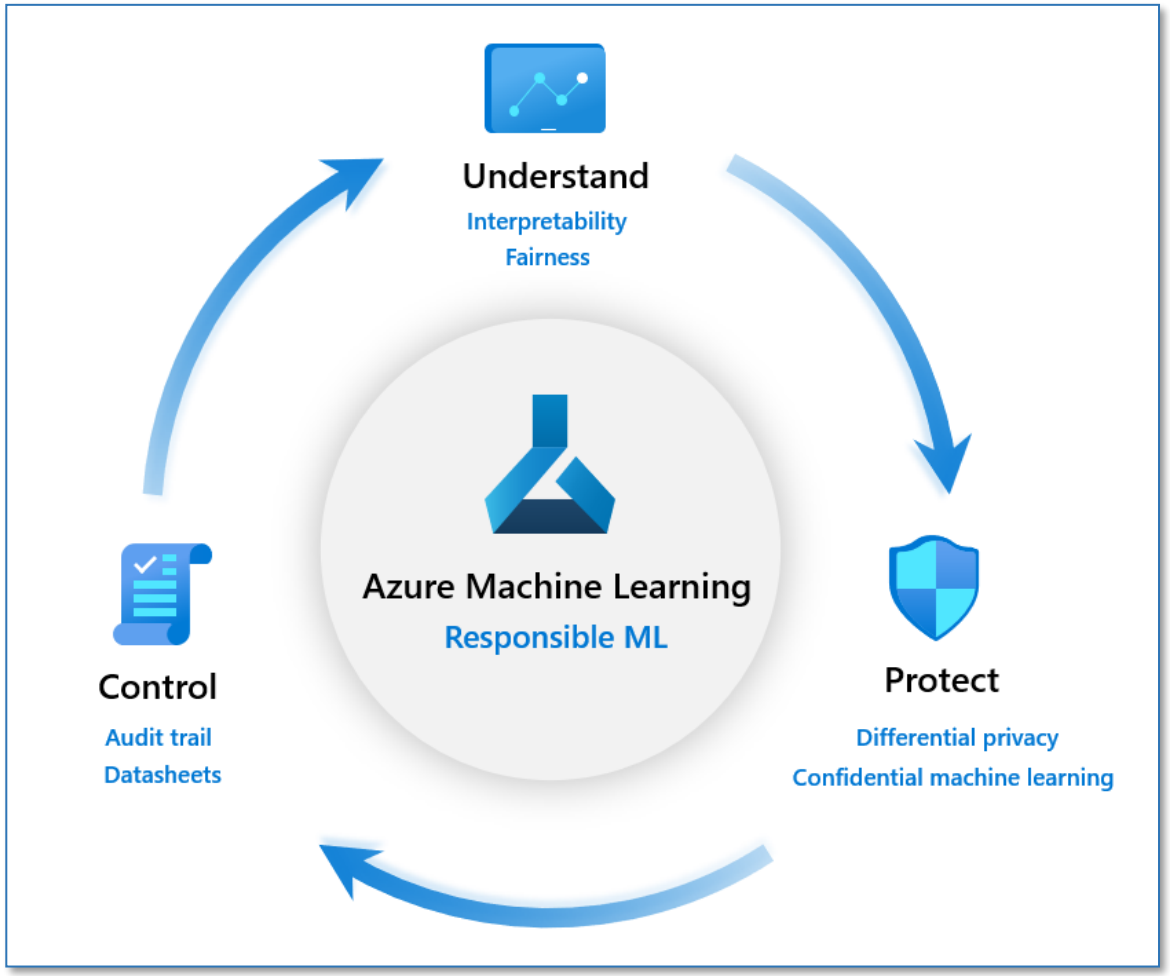
Interpret ML Models to Explain How & Why Decisions Were Made

| | |
|---|---|
| Explain Predictions | • Build trust by justifying the model's predictions<br>• Assess and mitigate unfair decisions |
| Assess Model Quality | • Compare models to validate fit |
| Identify the Most Important Features | • Which features had the most influence on predictions?<br>• How did each feature influence each individual prediction? |
| Identify Feature Relationships | • What feature interactions had the greatest influence on each prediction? |

# Responsible AI: Model Interpretability & Fairness

Mitigate Inequity by "Seeing Into" ML Models to Explain How & Why Decisions Were Made

- Understand ML Models:
  - Interpret & Explain Model Behavior
  - Assess & Mitigate Model Unfairness
- Bias in AI Systems Can Result in Unintended Consequences:
  - Withholding Opportunities, Resources or Information from Groups and/or Individuals
  - Reinforcing Inequity & Stereotypes



# Feature Influence: Partial Dependence

**A Univariate Method:** The influence of each feature's influence is measured before modeling

Iteratively sets all observations in the column to each unique value contained in that column.

*Then*

Then observe the correlation each value of the column has to the response variable (target).

## Original Values

| Loan | Income | IsBad |
|---|---|---|
| $11,200 | $108,000 | False |
| $10,000 | $65,000 | False |
| $8,000 | $20,000 | True |
| $16,000 | $110,000 | True |
| $4,000 | $155,000 | False |

## All Values Set to $20k

| Loan | Income | IsBad |
|---|---|---|
| $11,200 | **$20,000** | False |
| $10,000 | **$20,000** | False |
| $8,000 | **$20,000** | True |
| $16,000 | **$20,000** | True |
| $4,000 | **$20,000** | False |

## All Values Set to $65k

| Loan | Income | IsBad |
|---|---|---|
| $11,200 | **$65,000** | False |
| $10,000 | **$65,000** | False |
| $8,000 | **$65,000** | True |
| $16,000 | **$65,000** | True |
| $4,000 | **$65,000** | False |

# Demo: Responsible AI

Using Microsoft Fairlearn to Mitigate Unintended Bias and Unfair Predictions

# Deployment

## Machine Learning Deployment Scenarios & Architectures

# Azure MLOps: What's Involved in the ML Lifecycle?

**Develop & train model** with reusable Machine Learning Pipelines

**Package model** using containers to capture runtime dependencies for inference

**Validate model behavior**— functionally, in terms of responsiveness and regulatory compliance

**Deploy model**— to cloud and edge, for use in real-time – streaming – batch processing

**Monitor model** behavior **and** business value; know **when to replace/deprecate a stale model**

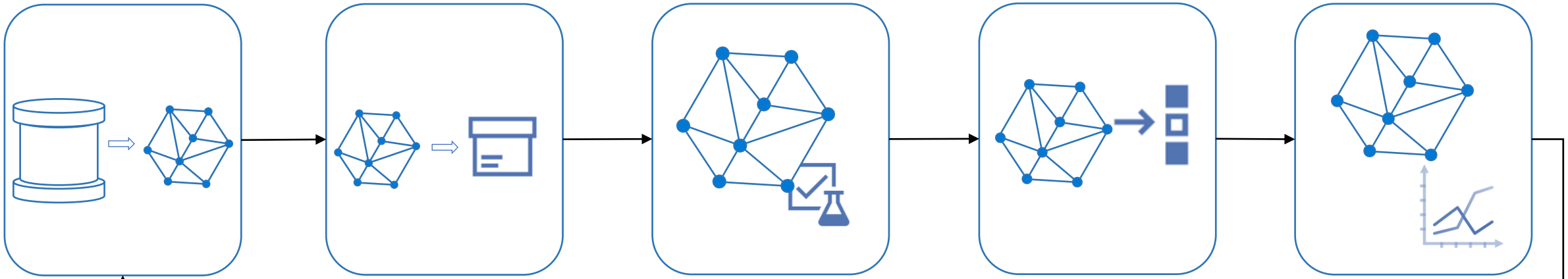
Train Model

Package Model

Validate Model

Deploy Model

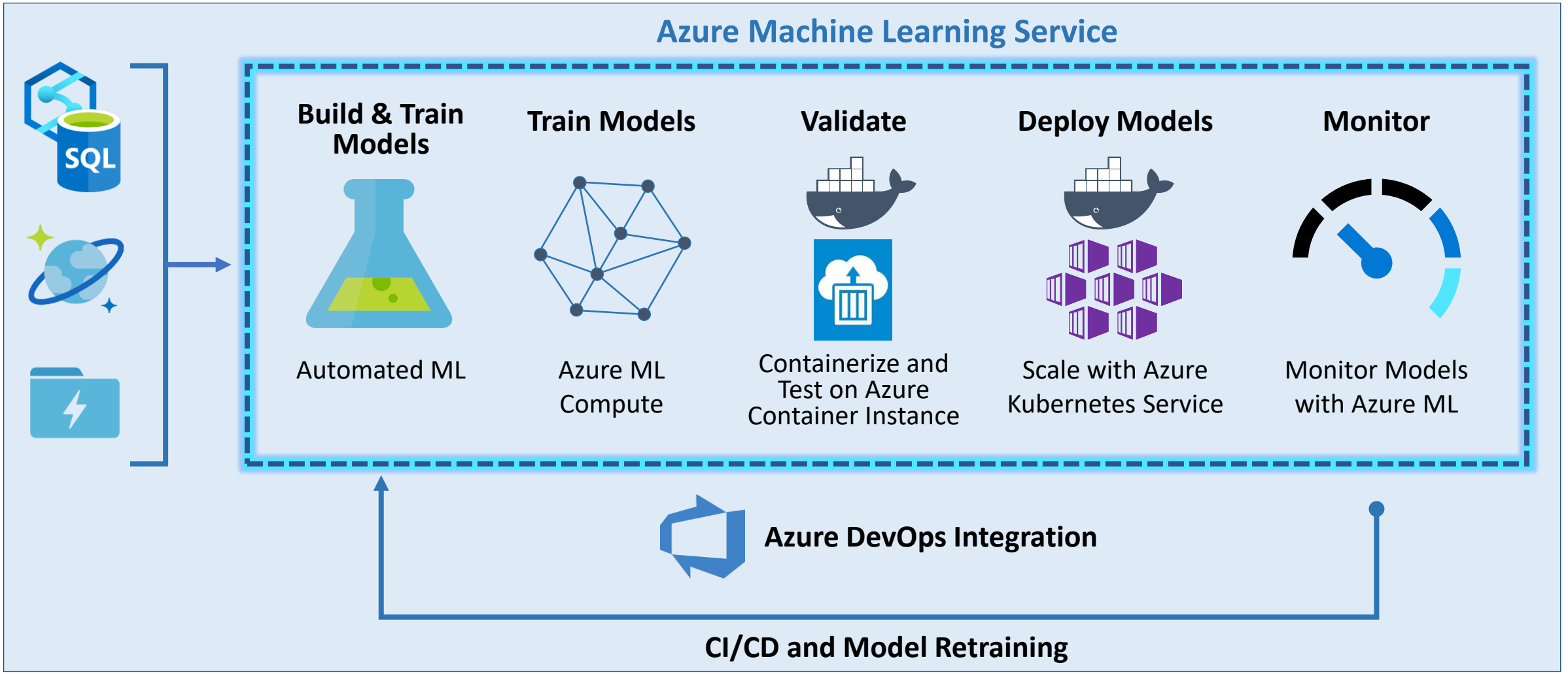
Monitor Model



Retrain Model

# Azure ML Services: Machine Learning Lifecycle

## Deploy Machine Learning Models at Scale with Azure DevOps Integration



# Azure Machine Learning: Model Management

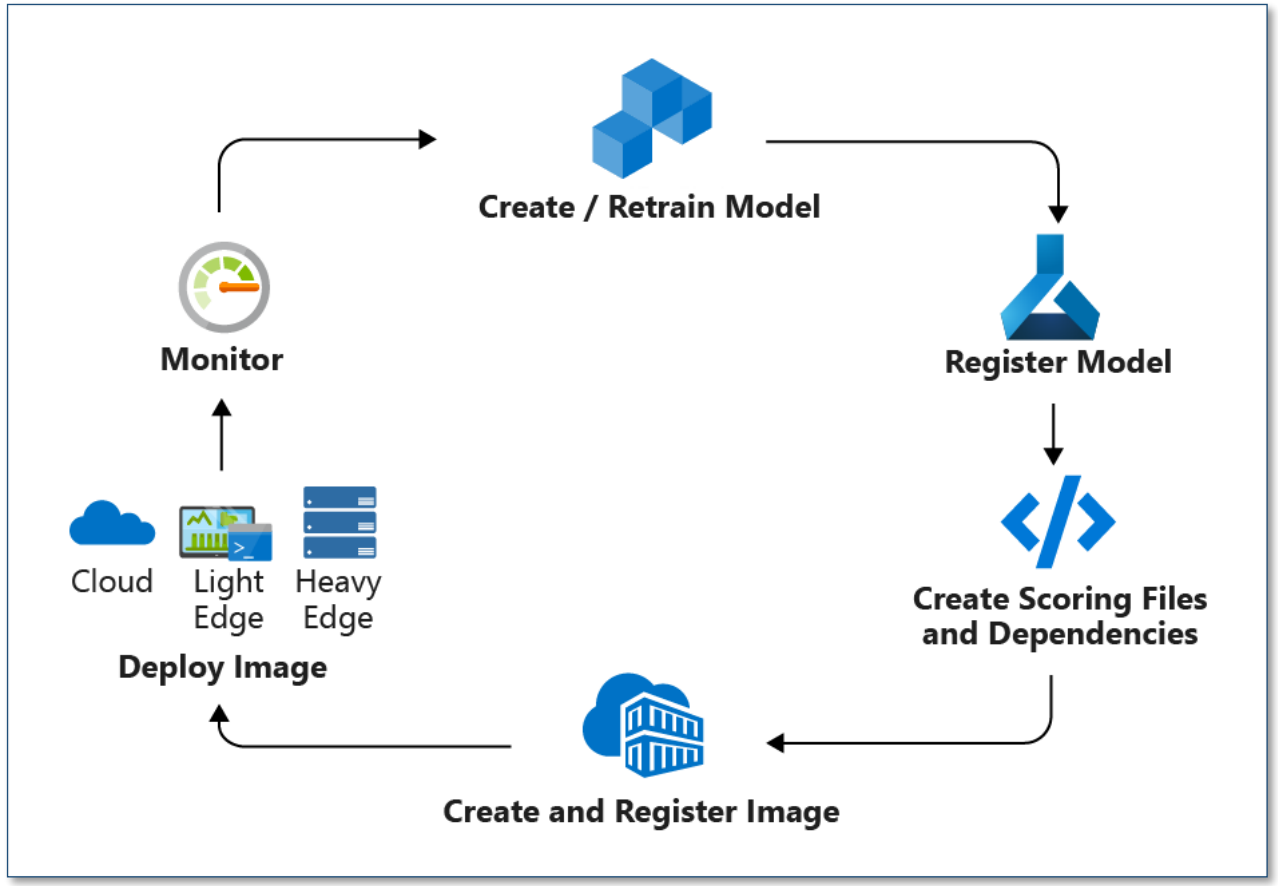
## Model Management in Azure ML usually involves these four steps:

**Step 1:** Register Model using the Model Registry

**Step 2:** Register Image using the Image Registry (the Azure Container Registry)

**Step 3:** Deploy Image to Cloud, or Edge Devices

**Step 4:** Monitor models—you can monitor input, output, and other relevant data from your model.



# Managing Drift: Model Monitoring & Maintenance

How can predictive models be tracked at run-time? How can their veracity be assured?

## Since the only constant *is change…*

**Concept Drift**

When the Target variable we're trying to predict changes

**Data Drift**

When the Input variables to our model change

**Model Drift**

When the design of the input variables change; i.e., feature engineering

## …Monitoring for drift is essential to maintaining the veracity of ML models