



What's new in Azure Machine Learning?

Timo Klerx – Data Scientist @paiqo

Marcel Franke – Cloud Solution Architect @Microsoft

About me – Timo Klerx

paiqo

Data Scientist @paiqo



<https://www.linkedin.com/in/timoklerx/>



tklerx@paiqo.com

About paiqo

paiqo

2019

Year of foundation

Platform



Microsoft
Partner

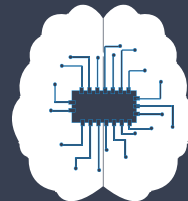
Gold Data Analytics
Gold Data Platform



10

Employees

AI



databricks

About me – Marcel Franke



Cloud Solution Architect @Microsoft Germany



<https://github.com/marcelfranke>



<https://www.linkedin.com/in/marcelfranke/>



marcel.franke@microsoft.com

meetup

#LetsTalkAboutAI

Microsoft AI User Group MeetUp

BER – CGN – HH – MUC

User Group NRW

<https://www.meetup.com/de-DE/Microsoft-AI-User-Group-NRW/>



Marek Matuszewski
Organisator
Beigetreten 27. Juli 2018



Timo Klerx
Co-Organisator
Beigetreten 28. Jan. 2020



Marcel Franke
Co-Organisator
Beigetreten 29. Aug. 2018

Where did it all start?

Microsoft Research

Turning ideas into reality for 27 years



8
MSR Labs

1k+
Researchers

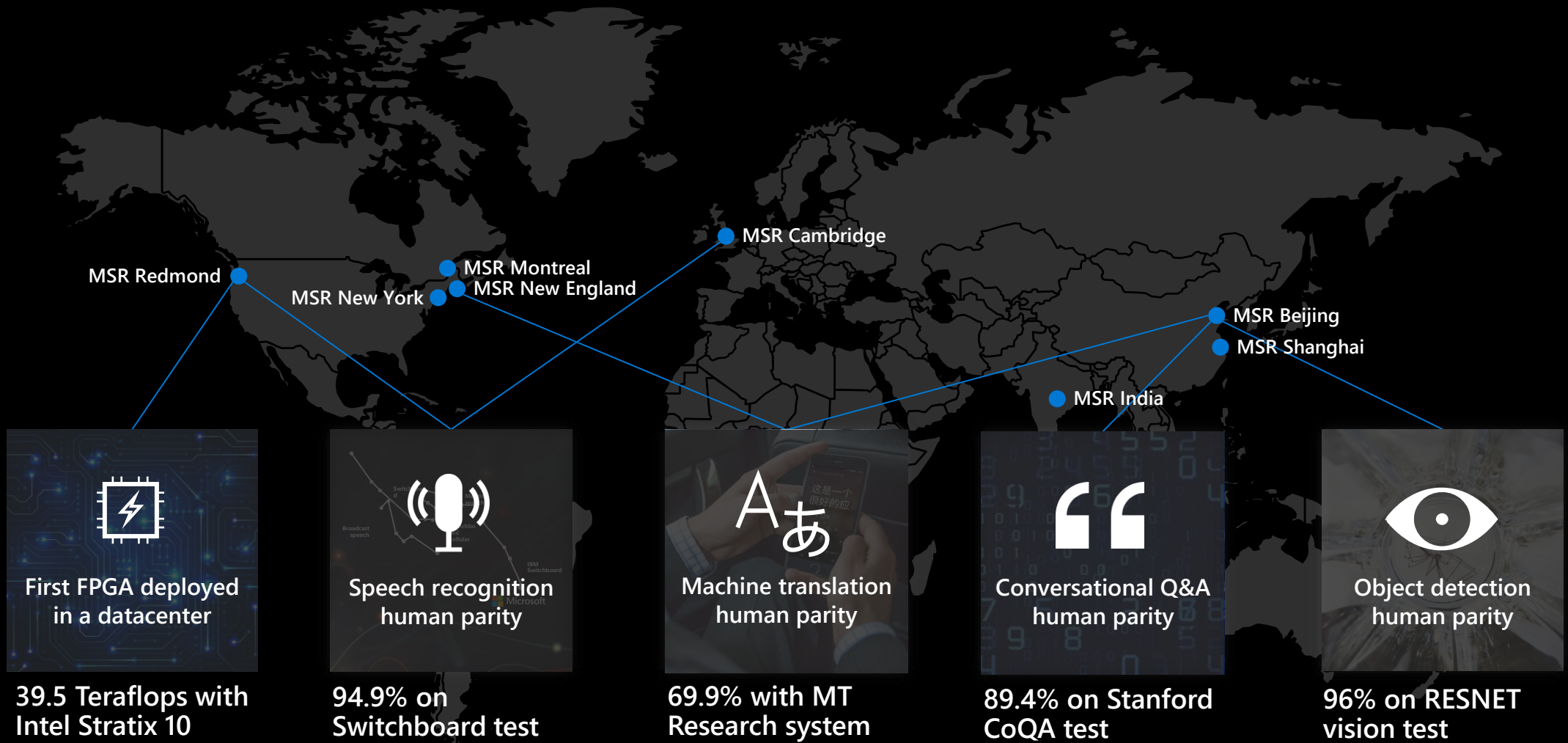
20+
Fields

4k+
WW Patents

22k+
Papers

Driving innovation

Fueled by breakthrough research



MSR Redmond

MSR New York

MSR Montreal

MSR New England

MSR Cambridge

MSR India

MSR Beijing

MSR Shanghai



First FPGA deployed
in a datacenter

39.5 Teraflops with
Intel Stratix 10



Speech recognition
human parity

94.9% on
Switchboard test



Machine translation
human parity

69.9% with MT
Research system



Conversational Q&A
human parity

89.4% on Stanford
CoQA test



Object detection
human parity

96% on RESNET
vision test

Azure Artificial Intelligence



Pre-built AI models

Azure Cognitive Services



Custom AI models

Azure Machine Learning

Demo

Cognitive Services

Azure Artificial Intelligence



Pre-built AI models

Azure Cognitive Services



Custom AI models

Azure Machine Learning

Machine Learning on Azure

Domain specific pretrained models

To simplify solution development



Vision



Speech



Language



Web search



Decision

Familiar Data Science tools

To simplify model development



Visual Studio Code



Azure Notebooks



Jupyter



Command line

Popular frameworks

To build advanced deep learning solutions



PyTorch



TensorFlow



Scikit-Learn



ONNX

Productive services

To empower data science and development teams



Azure Machine Learning



Azure Databricks



Machine Learning VMs

Powerful infrastructure

To accelerate deep learning



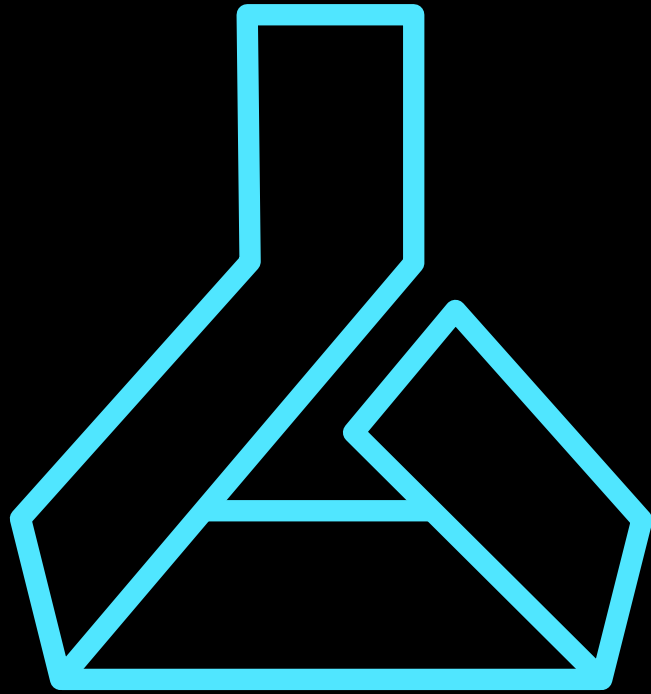
CPU



GPU



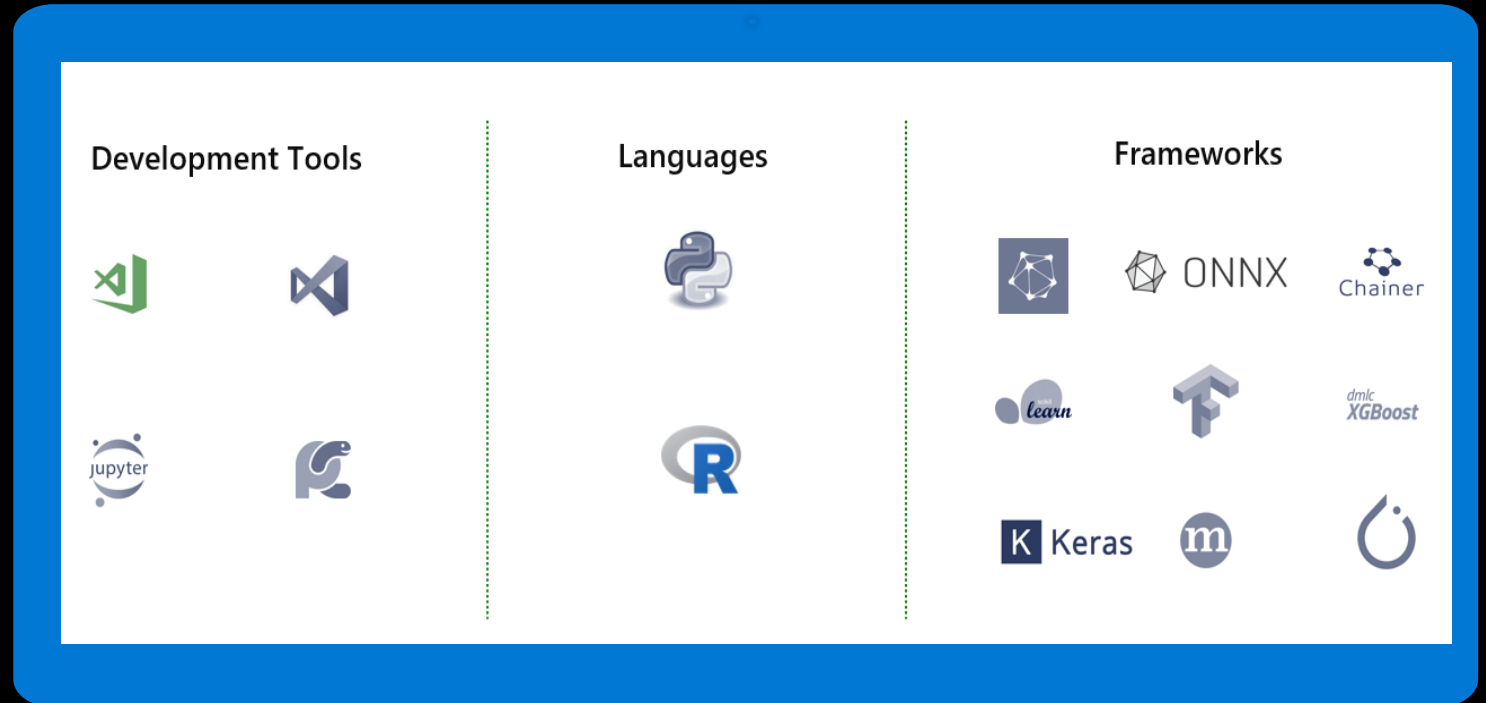
FPGA



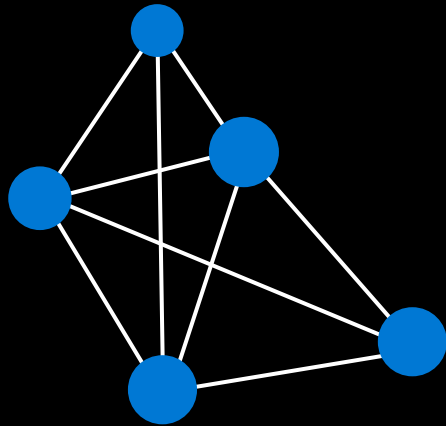
Azure Machine Learning

Azure Machine Learning

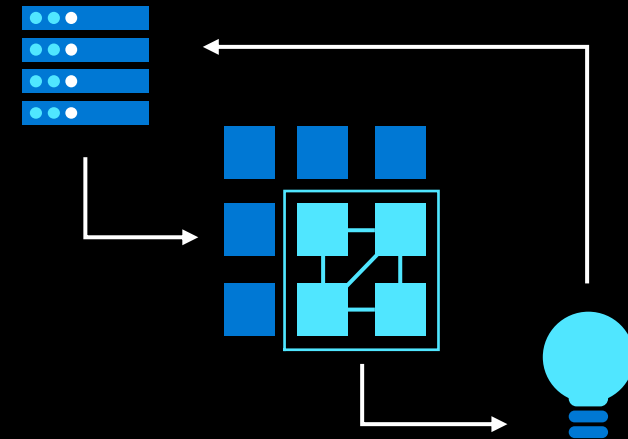
Open and interoperable



Azure Machine Learning service



Simplified machine learning



End to end lifecycle management

Open platform

Azure Machine Learning components

Experience

SDK, Notebooks, Drag-n-drop, Wizard

MLOps

Reproducible, Automatable, GitHub, CLI, REST

Datasets

Profiling, Drift, Labeling



Training

Experiments, Runs



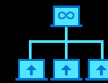
Model Registry

Models, Images



Inferencing

Batch, Realtime



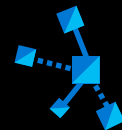
Compute

Jobs, Clusters, Instances



IoT Edge

Security, Mgmt, Deployment



Cloud

CPU, GPU, FPGA



Edge

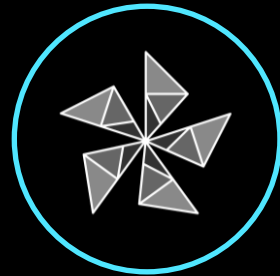
CPU, GPU, NPU



Open platform



Native MLflow support



ONNX Runtime updates



Azure Open Datasets

Demo

Azure Machine Learning

What's new?

Filter by title

Azure Machine Learning Documentation

Overview

What is Azure Machine Learning?

Azure Machine Learning vs Studio

Architecture & terms

> Tutorials

> Samples

> Concepts

> How-to guides

> Reference

Resources

Release notes

Azure roadmap

Pricing

Regional availability

Known issues

User forum

Stack Overflow

Compare our ML products

What happened to Workbench

Designer accessibility features

Download PDF

Azure Machine Learning release notes

01/21/2020 • 77 minutes to read • 12

In this article, learn about Azure Machine Learning releases. For the full SDK reference content, visit the Azure Machine Learning's [main SDK for Python](#) reference page.

See [the list of known issues](#) to learn about known bugs and workarounds.

2020-02-18

Azure Machine Learning SDK for Python v1.1.1rc0

• Bug fixes and improvements

o azure-cli-ml

- o Single instance profiling was fixed to produce a recommendation and was made available in core sdk.

o azureml-automl-core

- o The error logging has been improved.

o azureml-automl-runtime

- o Fixed the issue with forecasting when the data set contains short grains with long time gaps.
- o Fixed the issue when the auto max horizon is turned on and the date column contains dates in form of strings. We added proper conversion and sensible error if conversion to date is not possible
- o Using native NumPy and SciPy for serializing and deserializing intermediate data for FileCacheStore (used for local AutoML runs)
- o Fixed a bug where failed child runs could get stuck in Running state.

o azureml-cli-common

- o Single instance profiling was fixed to produce a recommendation and was made available in core sdk.

o azureml-core

- o Added `--grant-workspace-msi-access` as an additional parameter for the Datastore CLI for registering Azure Blob Container which will allow you to register Blob Container that is behind a VNet
- o Single instance profiling was fixed to produce a recommendation and was made available in core sdk.
- o Fixed the issue in `aks.py_deploy`
- o Validates the integrity of models being uploaded to avoid silent storage failures.

o azureml-interpret

- o added azureml-style exceptions to azureml-interpret
- o fixed DeepScoringExplainer serialization for keras models

o azureml-pipeline-core

- o Pipeline batch scoring notebook now uses ParallelRunStep

o azureml-pipeline-steps

- o Moved the `AutoMLStep` in the `azureml-pipeline-steps` package. Deprecated the `AutoMLStep` within `azureml-train-automl-runtime`.

o azureml-contrib-pipeline-steps

- o Optional parameter `side_inputs` added to `ParallelRunStep`. This parameter can be used to mount folder on the container. Currently supported types are `DataReference` and `PipelineData`.

o azureml-tensorboard

Is this page helpful?

Yes No

In this article

2020-02-18

2020-02-04

2020-01-21

2020-01-06

2019-12-23

2019-12-09

2019-11-25

2019-11-11

2019-11-04

2019-10-31

2019-10-21

2019-10-14

2019-10-08

2019-09-30

2019-09-16

2019-09-09

2019-09-03

2019-08-19

2019-08-05

2019-07-23

2019-07-09

2019-07-01

2019-06-25

2019-06-24

2019-06-10

2019-05-28

2019-05-14

2019-05-08

2019-05-06

2019-04-26

2019-04-26

2019-04-22

2019-04-17

2019-04-15

2019-04-08

2019-03-25

2019-03-11

2019-02-27

Highlights

Functionality

- Studio
- Designer
- Datasets
- Pipelines
- Notebooks
- Auto ML
- Labeling

Integration

- R SDK
- Databricks
- mlflow
- Visual Studio Code
- Event Grid
- Power BI
- Open Datasets

Enterprise Ready

- Interpretability
- RBAC
- VNET
- Monitoring
- MLOps
- Pricing

Functionality

Azure Machine Learning

ML for all skill levels

The screenshot displays the Azure Machine Learning web interface. On the left is a navigation sidebar with sections: Home, Author (Automated ML, Designer, Notebooks), Assets (Datasets, Experiments, Models, Endpoints), and Manage (Compute, Datastores, Notebook VMs). The main content area features a 'Welcome!' section with a 'Create new' dropdown and three 'Start now' buttons for Automated ML, Designer, and Notebooks. Below this is a 'My recent resources' section containing a table of runs.

Run Number	Experiment	Status Updated Time	Status
1	Sample_1_-_Regression...	9/27/2019, 1:38:37 PM	Completed
1474	category-based-prope...	9/18/2019, 4:37:10 PM	Completed
1475	category-based-prope...	9/18/2019, 3:49:21 PM	Completed
158	data-profiling	9/18/2019, 3:40:23 PM	Completed

[View all experiments →](#)

Drag and drop modeling with designer

User built in modules, data visualization, model evaluation

Automatically generate scoring files, register models and build images using AKS for scale

Custom code to run Python and R

Automobile price prediction Finished run

```
Python script
2 from __future__ import print_function
3 import pickle
4 import time
5 import math
6 import csv
7 import sys
8 from PIL import Image
9 import io
10
11 import pandas as pd
12 import pyarrow.parquet as pq
13 import torch
14 import torch.nn as nn
15 import torch.optim as optim
16 import torch.nn.functional as F
17 from torch.utils.data import DataLoader
18 from torch.utils.data import Dataset
19 import torchvision.transforms as transforms
20
21
22 class CNNSetting:
```

Start time 4/25/2019 6:38:32 PM
End time 4/25/2019 6:38:59 PM
Elapsed time 0:00:26.999
Status code Finished
Status details None
[View output log](#)

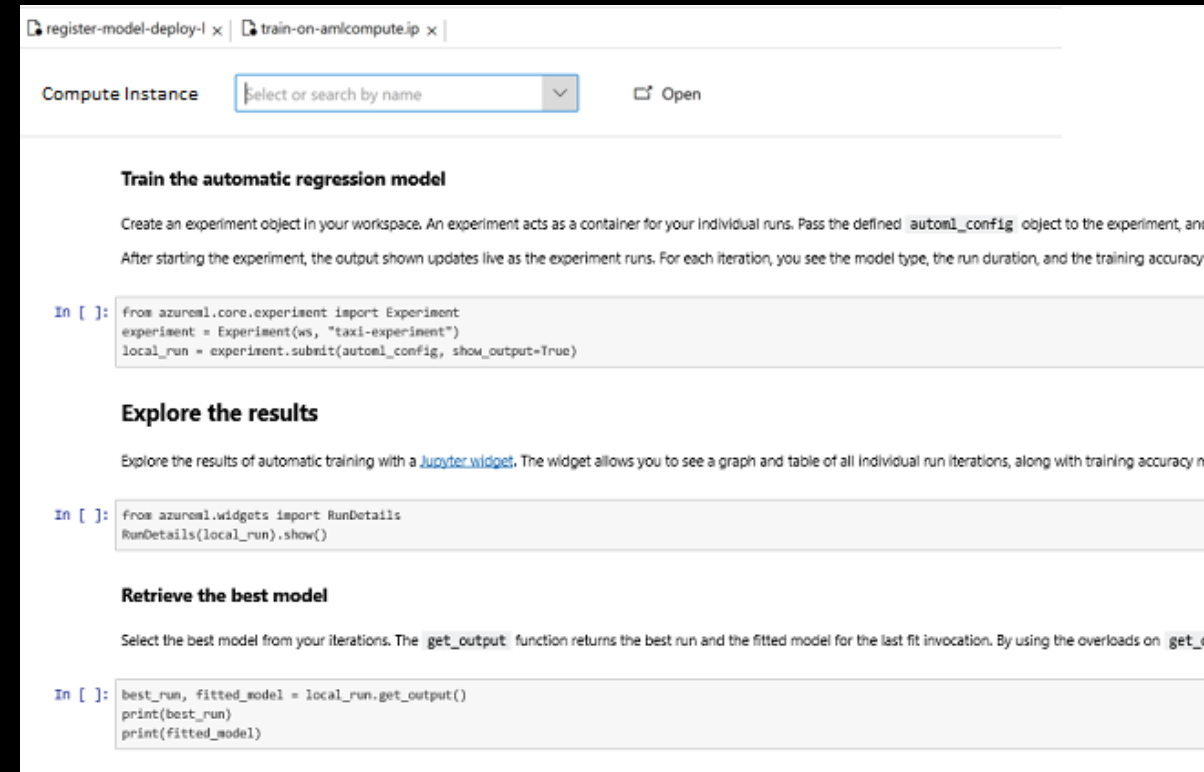
Run History Save Save As Discard Changes Run Predictive Web Service

Code-first ML with Notebooks

Build and deploy models easily using integrated notebooks and popular tools. Collaboratively debug models and share notebooks within the boundaries of workspace.

Broad support for Azure VM types and persisted low-level customization makes advanced scenarios

Pre-configured environment with up-to-date ML packages, GPU drivers and everything Data Scientists need to save time on setup tasks.



The screenshot shows a Jupyter notebook interface within the Azure ML workspace. At the top, there are two browser tabs: 'register-model-deploy-l x' and 'train-on-amlcompute.ip x'. Below the tabs, there is a 'Compute Instance' section with a dropdown menu labeled 'select or search by name' and an 'Open' button. The notebook content is organized into sections:

- Train the automatic regression model**: A text block explaining that an experiment object is created in the workspace to act as a container for individual runs. It mentions passing an `auto_ml_config` object and that output updates live as the experiment runs.
- Code cell 1**:

```
In [ ]: from azureml.core.experiment import Experiment
experiment = Experiment(ws, "taxi-experiment")
local_run = experiment.submit(auto_ml_config, show_output=True)
```
- Explore the results**: A text block explaining that a `Jupyter.widget` is used to explore the results of automatic training, showing a graph and table of individual run iterations.
- Code cell 2**:

```
In [ ]: from azureml.widgets import RunDetails
RunDetails(local_run).show()
```
- Retrieve the best model**: A text block explaining that the `get_output` function returns the best run and the fitted model for the last fit invocation.
- Code cell 3**:

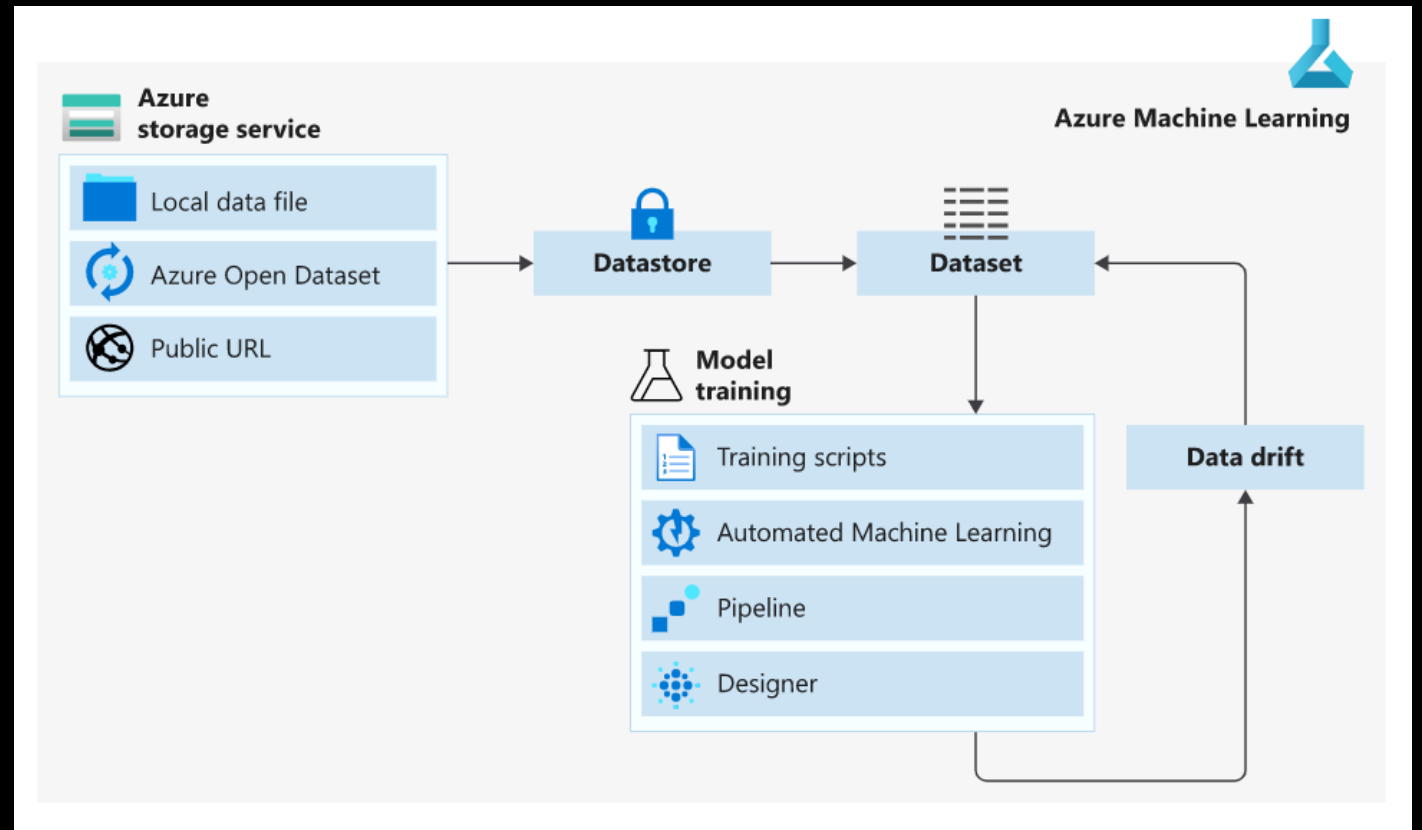
```
In [ ]: best_run, fitted_model = local_run.get_output()
print(best_run)
print(fitted_model)
```

Demo

Studio, Designer, Notebooks

Datasets

Easily connect with
your data

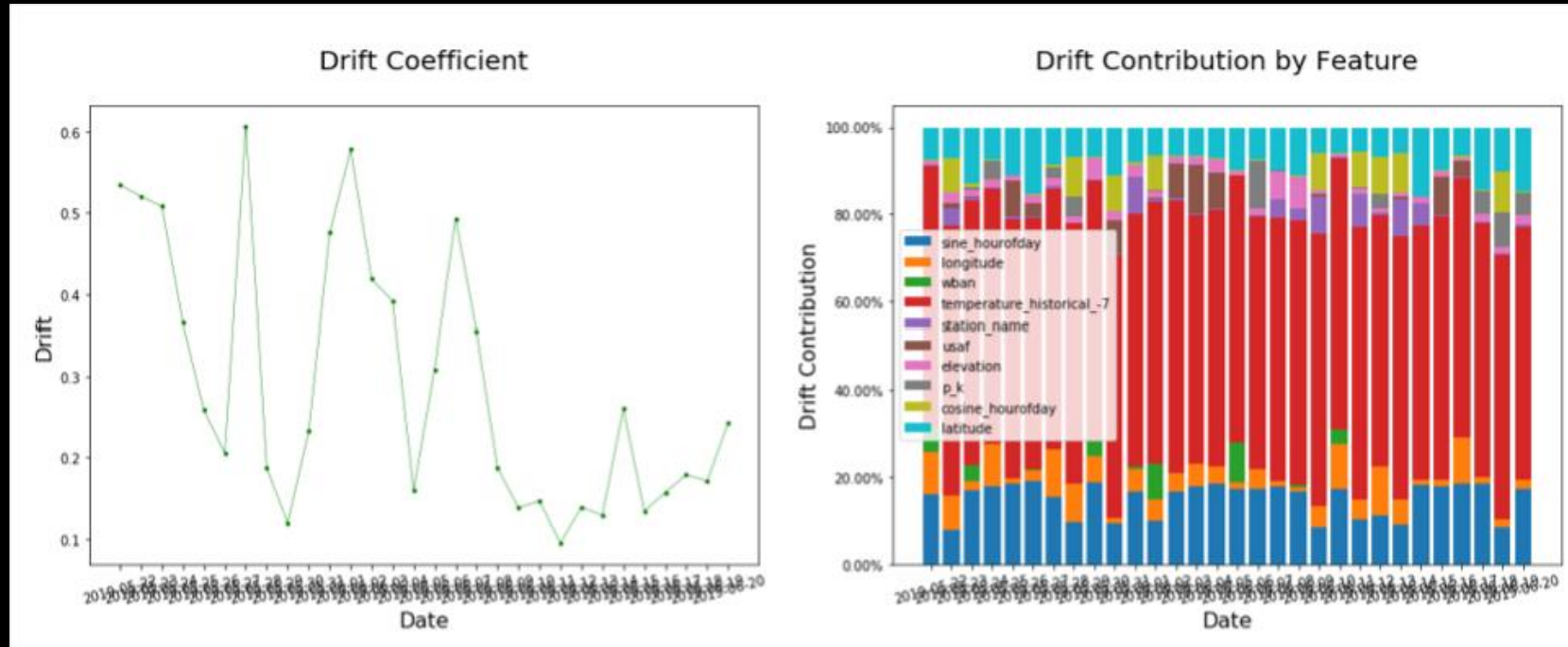


Demo

Datasets & Data drift

Data Drift Monitoring

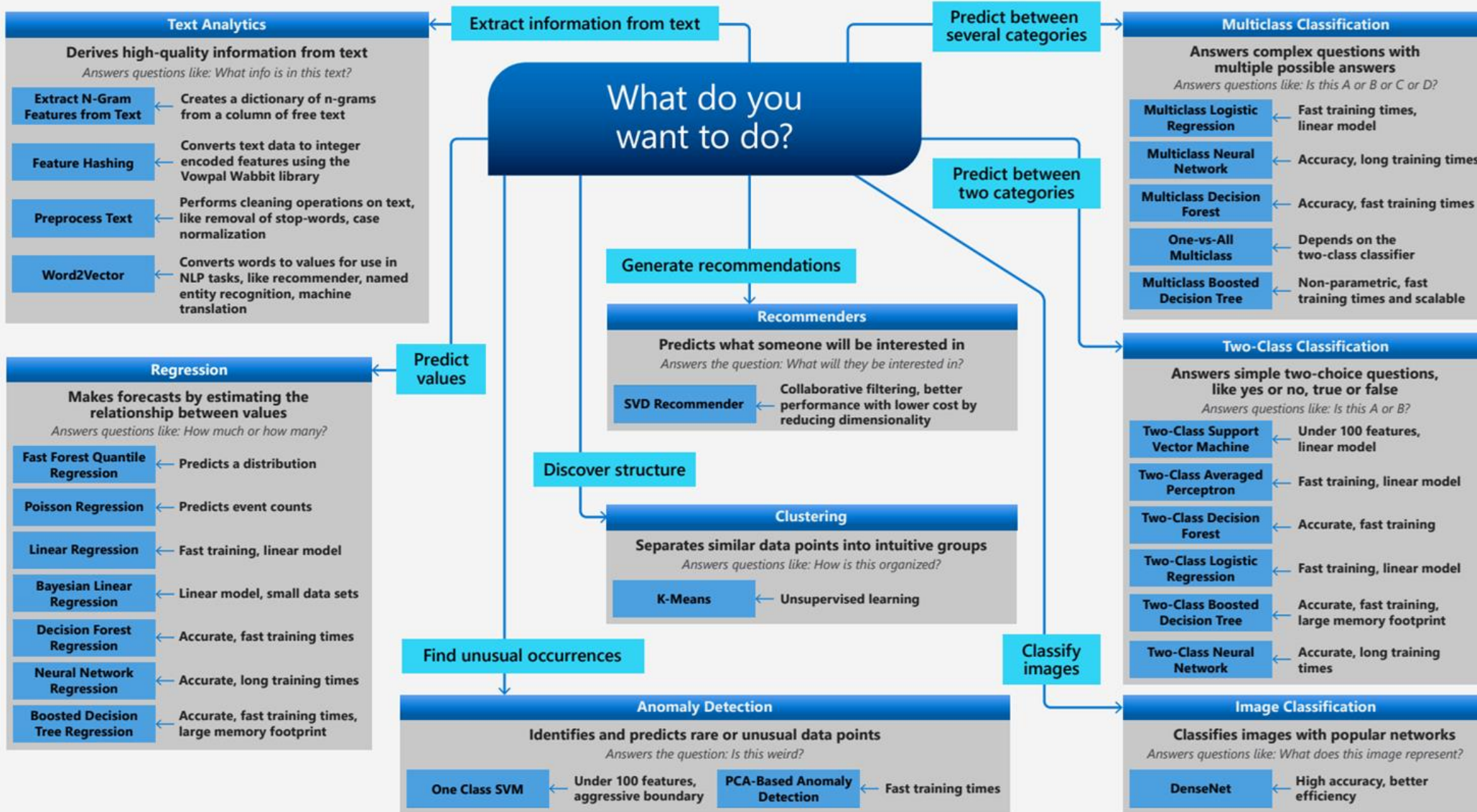
Monitor Data Drift on deployed models





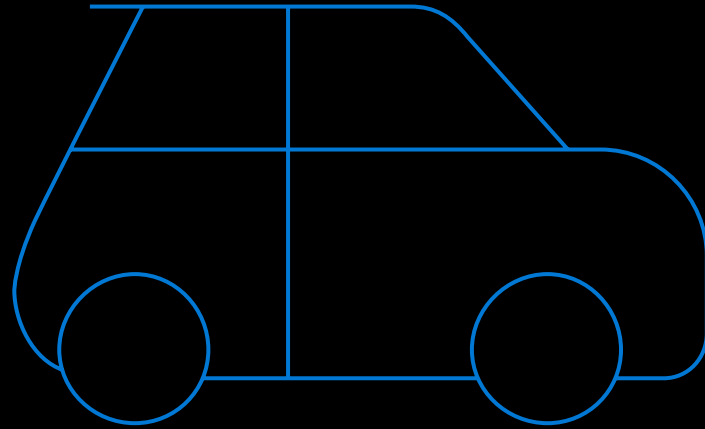
Microsoft Azure Machine Learning Algorithm Cheat Sheet

This cheat sheet helps you choose the best machine learning algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the goal you want to achieve with your data.



Azure Machine Learning

Automated machine learning



How much is this car worth?

Model creation is typically a time-consuming process

Which features?

Mileage

Condition

Car brand

Year of make

Regulations

...

Gradient Boosted

Nearest Neighbors

SVM

Bayesian Regression

LGBM

...

Which algorithm?

Parameter 1

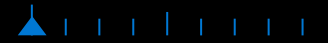
Parameter 2

Random Forest Split

Random Forest Leaf

Others

Which parameters?



30%

Model

Model creation is typically a time consuming process

Which features?

Mileage

Condition

Car brand

Year of make

Regulations

...

Which algorithm?

Gradient Boosted

Nearest Neighbors

SVM

Bayesian Regression

LGBM

...

Which parameters?

Neighbors

Weights

Max Samples Split

Min Samples Leaf

Others

| | | | | | | | | | ▲ | |

| | | ▲ | | | | | | |

| | | | | ▲ | | | | |

| | ▲ | | | | | | | |

| | | | | | | | | | ▲ | |

30%

Model

Iterate

Model creation is typically a time consuming process

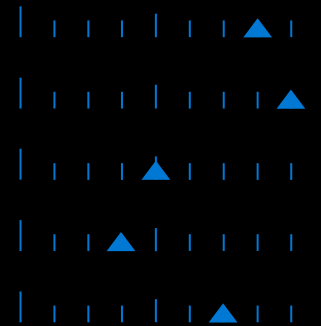
Which features?



Which algorithm?



Which parameters?



30%

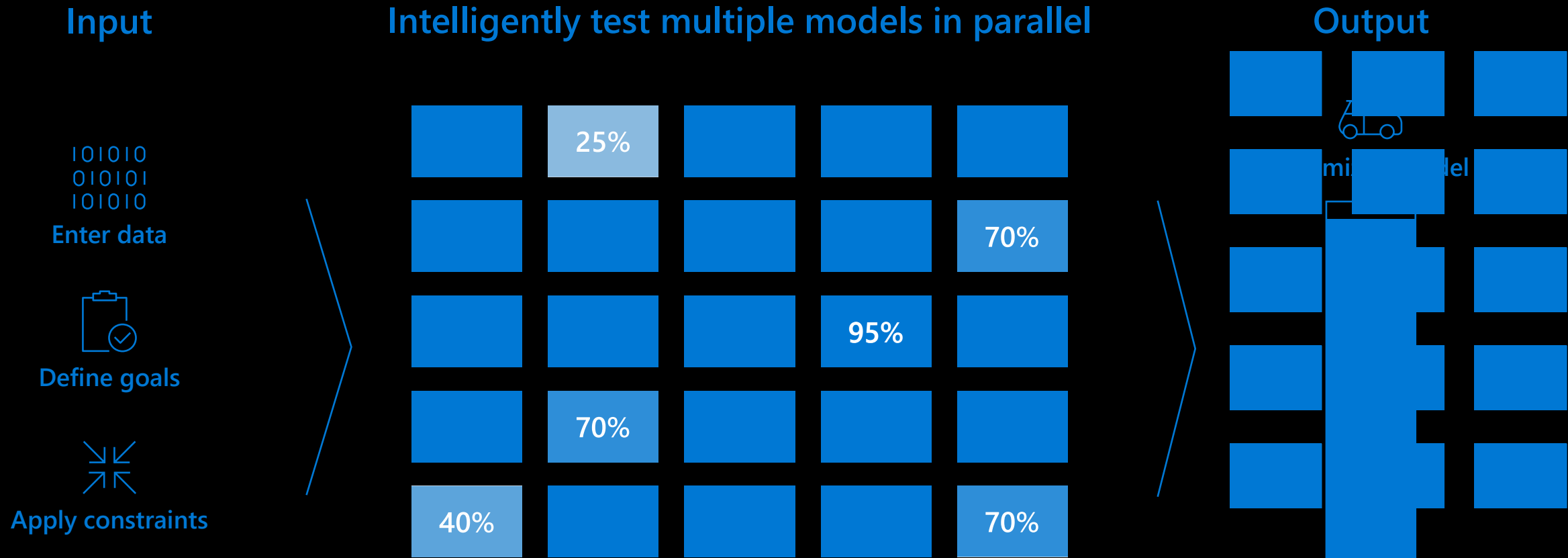
15%



Iterate

Azure Machine Learning accelerates model selection

with automated machine learning



Demo

Auto ML

Supported Algorithms

Classification	Regression	Time Series Forecasting
Logistic Regression	Elastic Net	Elastic Net
Light GBM	Light GBM	Light GBM
Gradient Boosting	Gradient Boosting	Gradient Boosting
Decision Tree	Decision Tree	Decision Tree
K Nearest Neighbors	K Nearest Neighbors	K Nearest Neighbors
Linear SVC	LARS Lasso	LARS Lasso
Support Vector Classification (SVC)	Stochastic Gradient Descent (SGD)	Stochastic Gradient Descent (SGD)
Random Forest	Random Forest	Random Forest
Extremely Randomized Trees	Extremely Randomized Trees	Extremely Randomized Trees
Xgboost	Xgboost	Xgboost
DNN Classifier	DNN Regressor	DNN Regressor
DNN Linear Classifier	Linear Regressor	Linear Regressor
Naive Bayes	Fast Linear Regressor	Auto-ARIMA
Stochastic Gradient Descent (SGD)	Online Gradient Descent Regressor	Prophet
Averaged Perceptron Classifier		ForecastTCN
Linear SVM Classifier		

Data Preprocessing

data is automatically scaled or normalized to help algorithms perform well

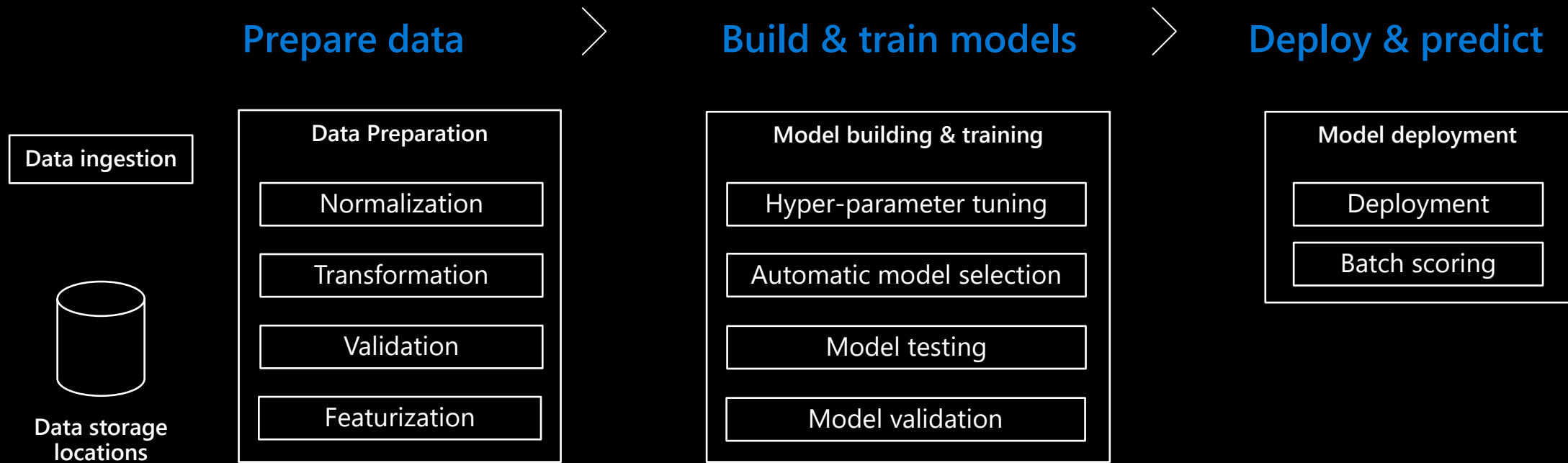
Automatic Preprocessing

Scaling & normalization	Description
StandardScaleWrapper	Standardize features by removing the mean and scaling to unit variance
MinMaxScaler	Transforms features by scaling each feature by that column's minimum and maximum
MaxAbsScaler	Scale each feature by its maximum absolute value
RobustScaler	This Scaler features by their quantile range
PCA	Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space
TruncatedSVDWrapper	This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). Contrary to PCA, this estimator does not center the data before computing the singular value decomposition, which means it can work with <code>scipy.sparse</code> matrices efficiently
SparseNormalizer	Each sample (that is, each row of the data matrix) with at least one non-zero component is rescaled independently of other samples so that its norm (l1 or l2) equals one

Advanced Preprocessing

Preprocessing steps	Description
Drop high cardinality or no variance features	Drop these from training and validation sets, including features with all values missing, same value across all rows or with extremely high cardinality (for example, hashes, IDs, or GUIDs).
Impute missing values	For numerical features, impute with average of values in the column. For categorical features, impute with most frequent value.
Generate additional features	For DateTime features: Year, Month, Day, Day of week, Day of year, Quarter, Week of the year, Hour, Minute, Second. For Text features: Term frequency based on unigrams, bi-grams, and tri-character-grams.
Transform and encode	Numeric features with few unique values are transformed into categorical features. One-hot encoding is performed for low cardinality categorical; for high cardinality, one-hot-hash encoding.
Word embeddings	Text featurizer that converts vectors of text tokens into sentence vectors using a pre-trained model. Each word's embedding vector in a document is aggregated together to produce a document feature vector.
Target encodings	For categorical features, maps each category with averaged target value for regression problems, and to the class probability for each class for classification problems. Frequency-based weighting and k-fold cross validation is applied to reduce over fitting of the mapping and noise caused by sparse data categories.
Text target encoding	For text input, a stacked linear model with bag-of-words is used to generate the probability of each class.
Weight of Evidence (WoE)	Calculates WoE as a measure of correlation of categorical columns to the target column. It is calculated as the log of the ratio of in-class vs out-of-class probabilities. This step outputs one numerical feature column per class and removes the need to explicitly impute missing values and outlier treatment.
Cluster Distance	Trains a k-means clustering model on all numerical columns. Outputs k new features, one new numerical feature per cluster, containing the distance of each sample to the centroid of each cluster.

Azure Machine Learning pipelines



Integration

Databricks Integration

- Customers who use Azure Databricks for advanced analytics can now use the same cluster to run experiments with or without automated machine learning
- Databricks as a Compute Target from AML Pipelines

Integration with mlflow



Open source machine learning platform

- Works with any ML library, algorithm, language, etc
- *Open interface* design (use with any code you already have)

mlflow
Tracking

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

mlflow
Projects

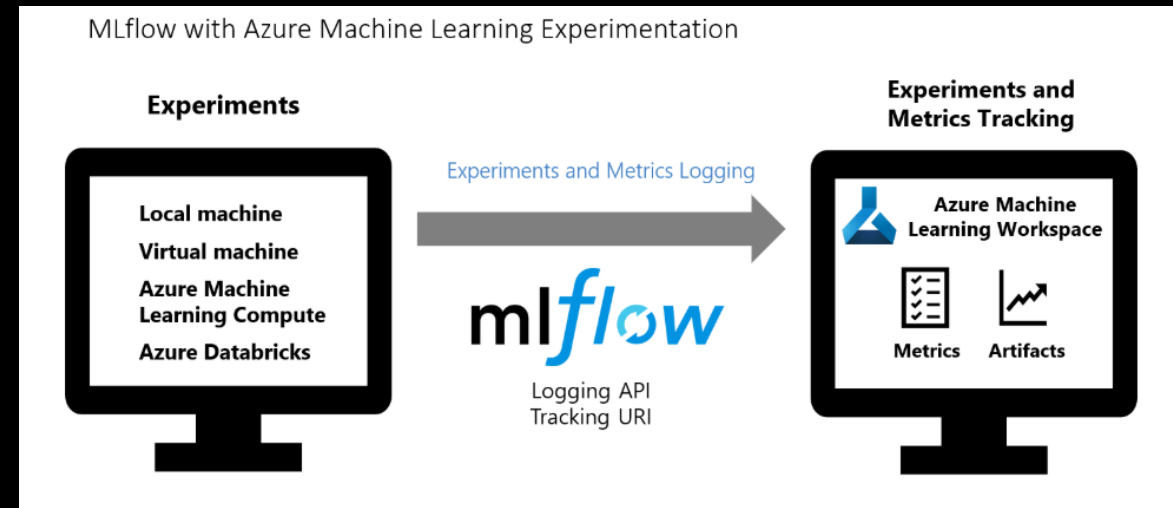
Packaging format for reproducible runs and workflows

mlflow
Models

General format that standardizes deployment paths

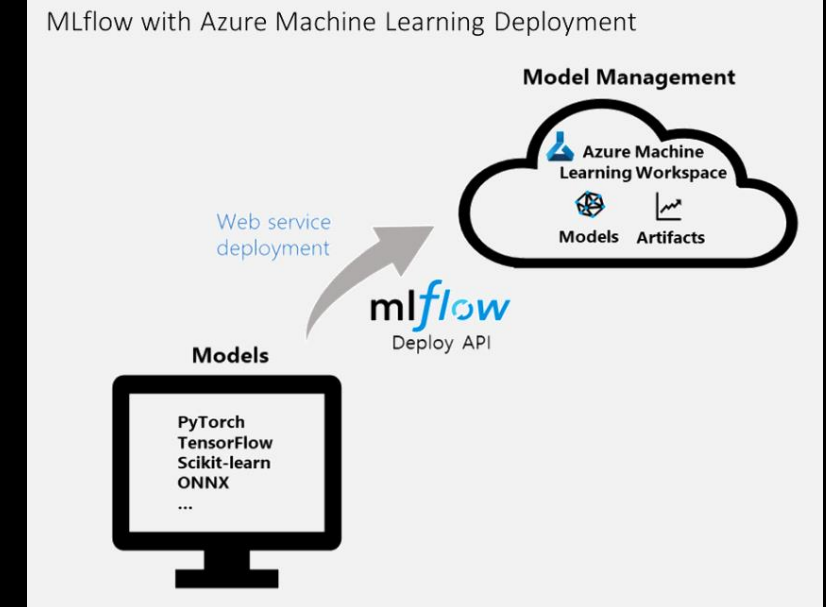
new
mlflow
Model Registry

Centralized model management, review & sharing



Integration with mlflow

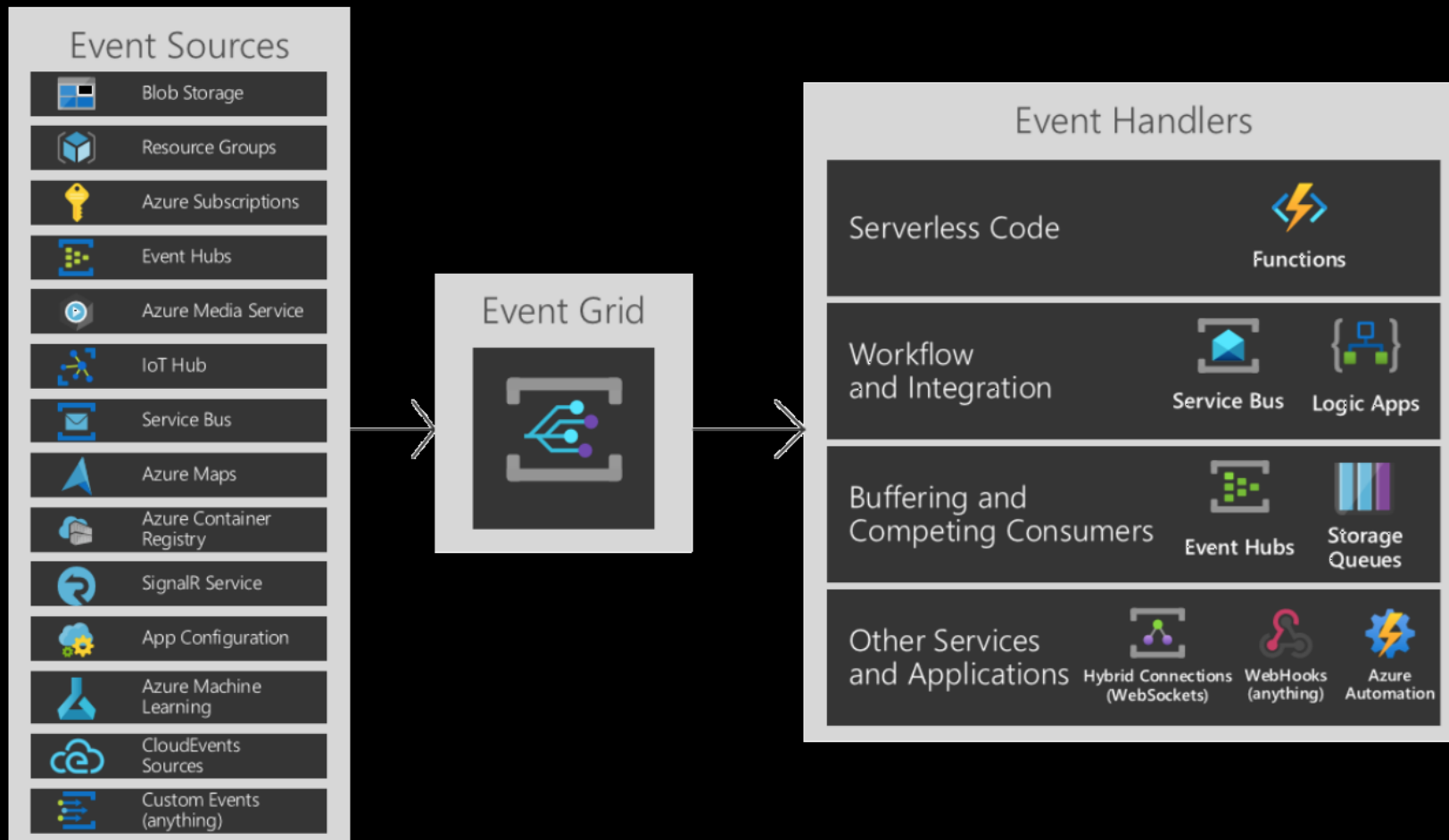
	MLflow Tracking	Azure Machine Learning Python SDK	Azure Machine Learning CLI	Azure Machine Learning Studio
Manage workspace		✓	✓	✓
Use data stores		✓	✓	
Log metrics	✓	✓		
Upload artifacts	✓	✓		
View metrics	✓	✓	✓	✓
Manage compute		✓	✓	✓



Demo

mlflow Integration

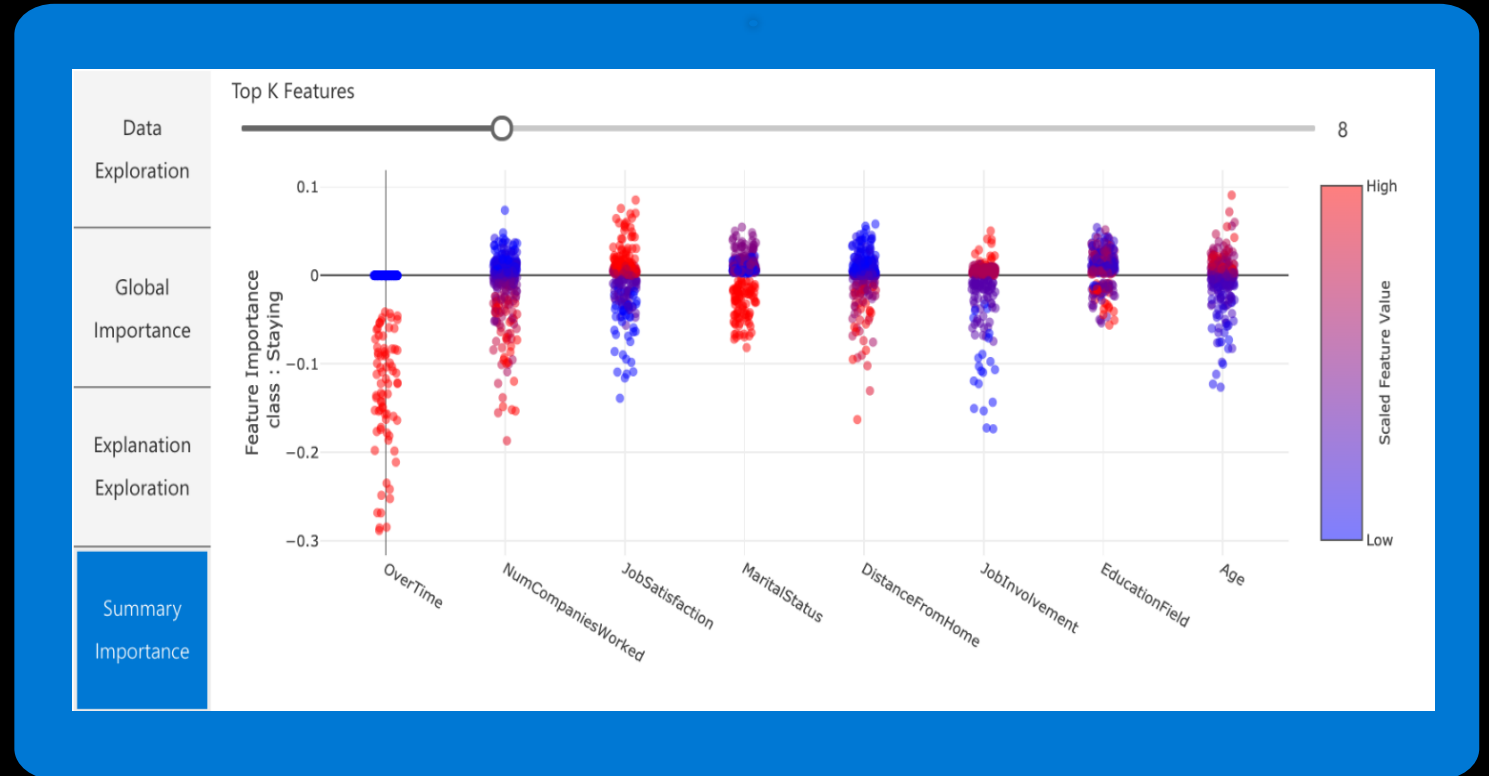
Event Grid Integration



Enterprise Ready

Azure Machine Learning

Trusted ML



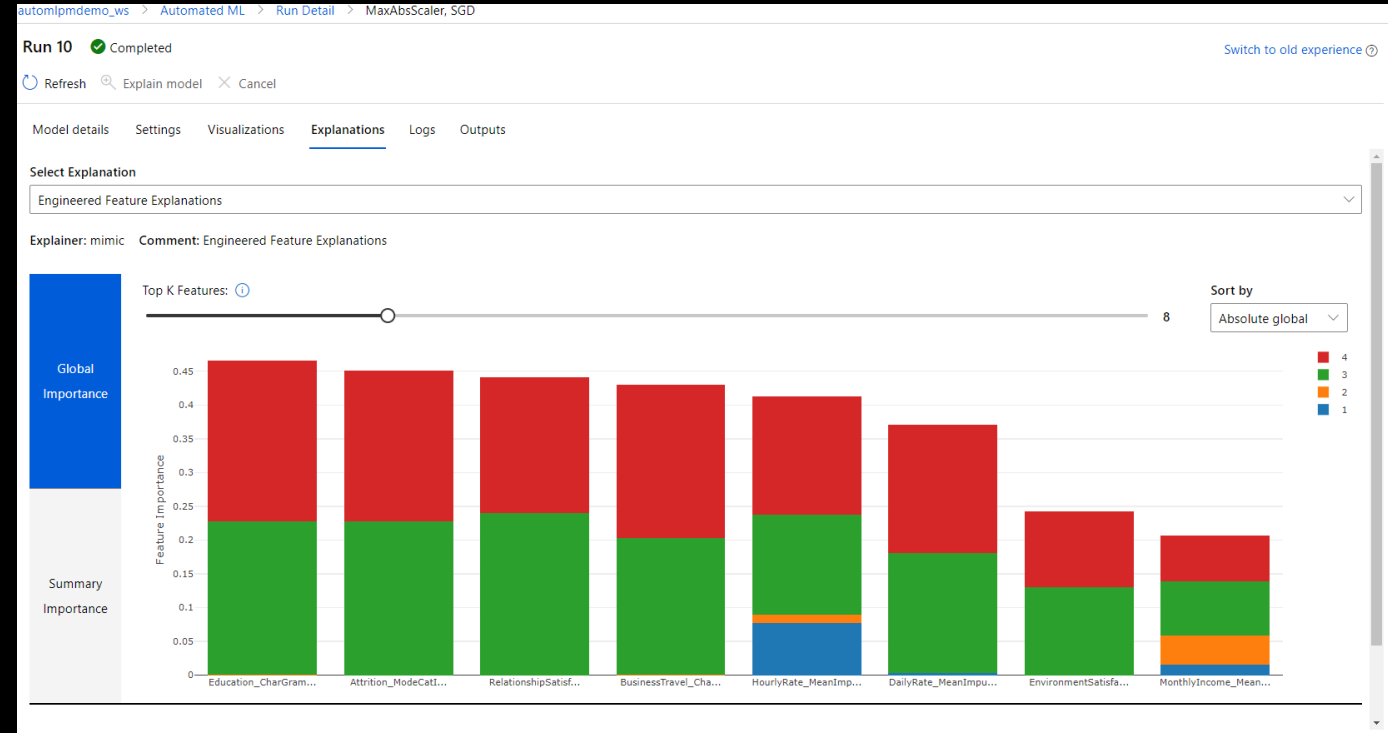
Understanding models and data with Interpretability

Built-in model explanations available at training and inferencing

Glass box and black box explainers for your custom needs

Data understanding, feature importance and local or global explanations for a data point or dataset

Out-of-box visualizations for model predictions and explanations

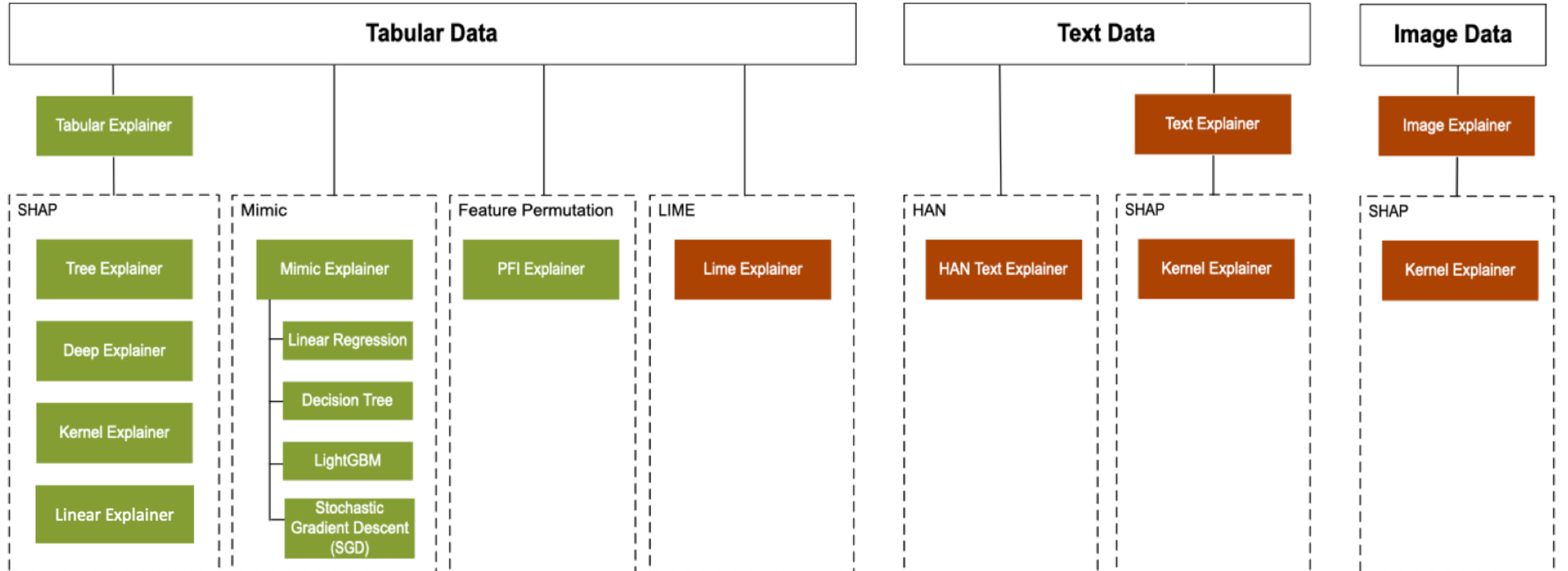


Understanding models and data

Machine Learning Interpretability

Legend:

- Main Package
- Contrib Package



Demo

Understanding models and data

Editions

Azure Machine Learning editions

FEATURES	BASIC	ENTERPRISE
	For open source development at cloud scale with a code-first experience.	Basic + UI capabilities + secure and comprehensive machine learning lifecycle management for all skill levels.
Automated machine learning		
Create and run experiments in notebooks	✓	✓
Create and run experiments in studio web experience		✓
Industry leading forecasting capabilities		✓
Support for deep learning and other advanced learners		✓
Large data support (up to 100GB)		✓
Interpretability in UI		✓
Machine Learning Pipelines		
Create, run, and publish pipelines using the Azure ML SDK	✓	✓
Create pipeline endpoints using the Azure ML SDK	✓	✓
Create, edit, and delete scheduled runs of pipelines using the Azure ML SDK	✓	✓
Create and publish custom modules using the Azure ML SDK	✓	✓
View pipeline run details in studio	✓	✓
Create, run, visualize, and publish pipelines in Azure ML designer		✓
Create pipeline endpoints in Azure ML designer		✓
Create, edit, and delete scheduled runs of pipelines in Azure ML designer		✓
Create and publish custom modules in Azure ML designer		✓
Integrated notebooks		
Workspace notebook and file sharing	✓	✓
R and Python support	✓	✓

Build enterprise grade ML

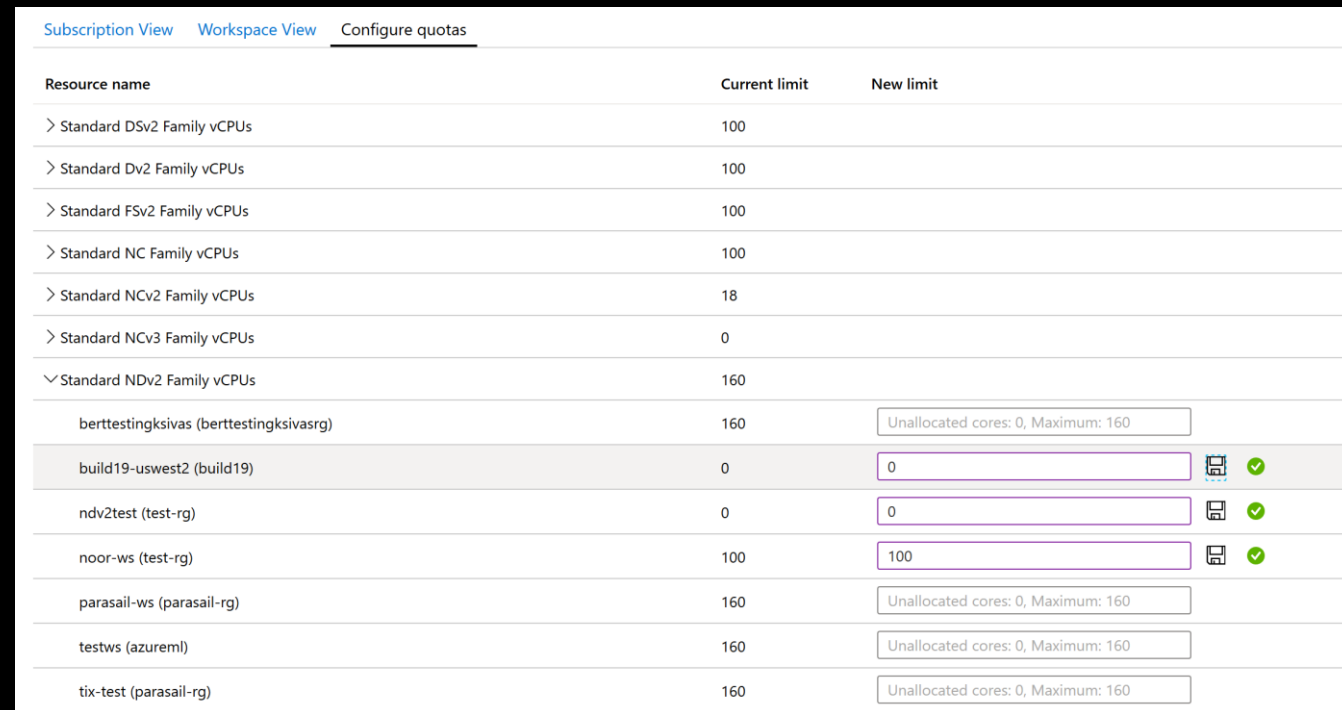
with security and governance capabilities

Granular RBAC (Role Based Access Controls) give permissions to team members based on operations they can perform







VNET (virtual network) acts as a security boundary, isolating ML resources from the public internet

Scoring endpoint authentication

Capacity management helps with efficient resource distribution, using capacity limits



The screenshot shows the 'Configure quotas' page in the Azure portal. It displays a table of resource names, current limits, and new limits. The table is organized into sections for different vCPU families and specific resource groups. The 'New limit' column contains input fields for setting new limits, with some fields showing 'Unallocated cores: 0, Maximum: 160' and others showing '0' or '100'. There are also icons for saving and confirming changes.

Resource name	Current limit	New limit
> Standard Dsv2 Family vCPUs	100	
> Standard Dv2 Family vCPUs	100	
> Standard FSv2 Family vCPUs	100	
> Standard NC Family vCPUs	100	
> Standard NCv2 Family vCPUs	18	
> Standard NCv3 Family vCPUs	0	
∨ Standard NDv2 Family vCPUs	160	
berttestingkivas (berttestingkivasrg)	160	Unallocated cores: 0, Maximum: 160
build19-uswest2 (build19)	0	0  
ndv2test (test-rg)	0	0  
noor-ws (test-rg)	100	100  
parasail-ws (parasail-rg)	160	Unallocated cores: 0, Maximum: 160
testws (azureml)	160	Unallocated cores: 0, Maximum: 160
tix-test (parasail-rg)	160	Unallocated cores: 0, Maximum: 160

MLOps



Model reproducibility

Model validation

Model deployment

Model retraining

DevOps



Code reproducibility



Code testing



App deployment

MLOps



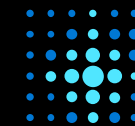
Model reproducibility



Model validation



Model deployment



Model retraining

MLOps with Azure Machine Learning

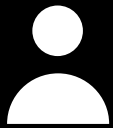
Collaborate

Build app

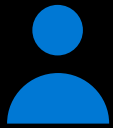
Test app

Release app

Monitor app



App developer
using Azure DevOps



Data scientist using
Azure Machine Learning

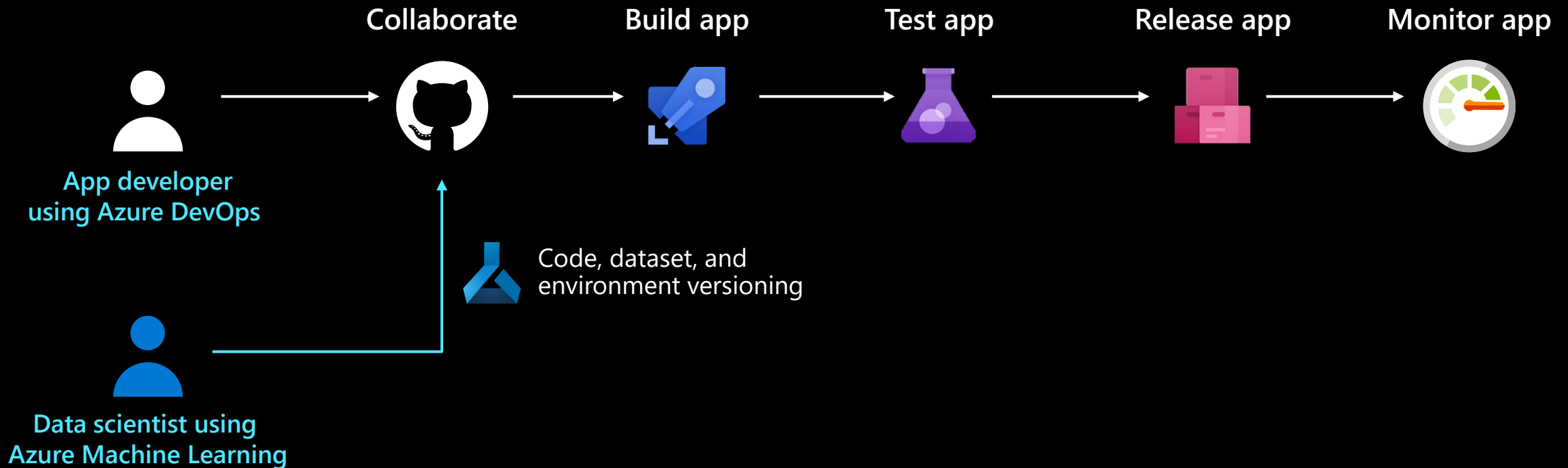
Model reproducibility

Model validation

Model deployment

Model retraining

MLOps with Azure Machine Learning



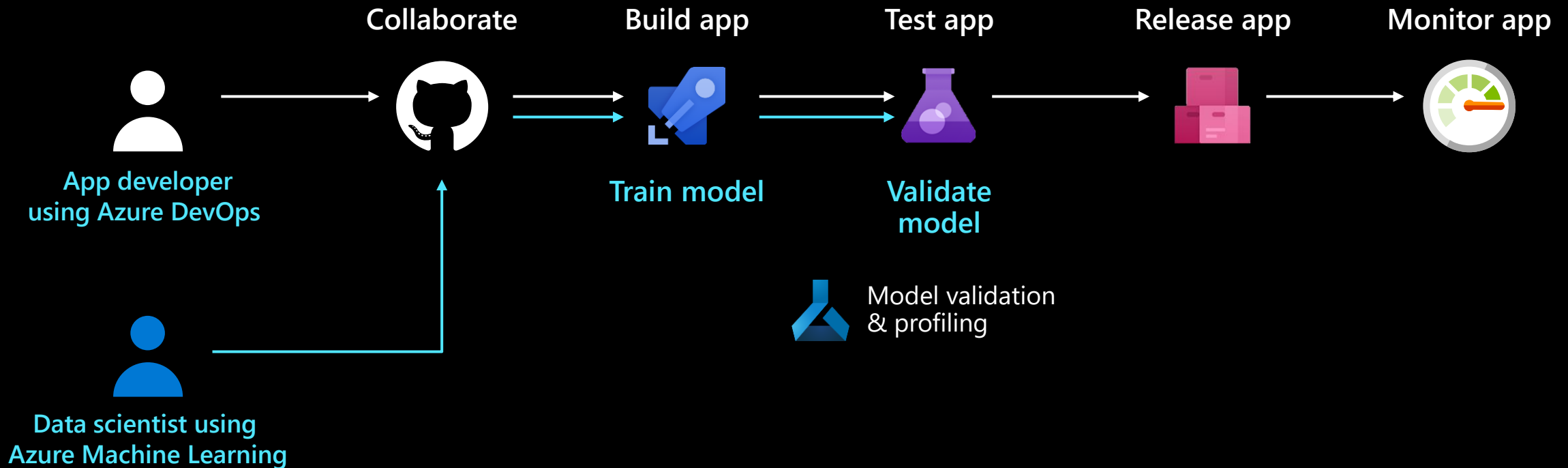
Model reproducibility

Model validation

Model deployment

Model retraining

MLOps with Azure Machine Learning



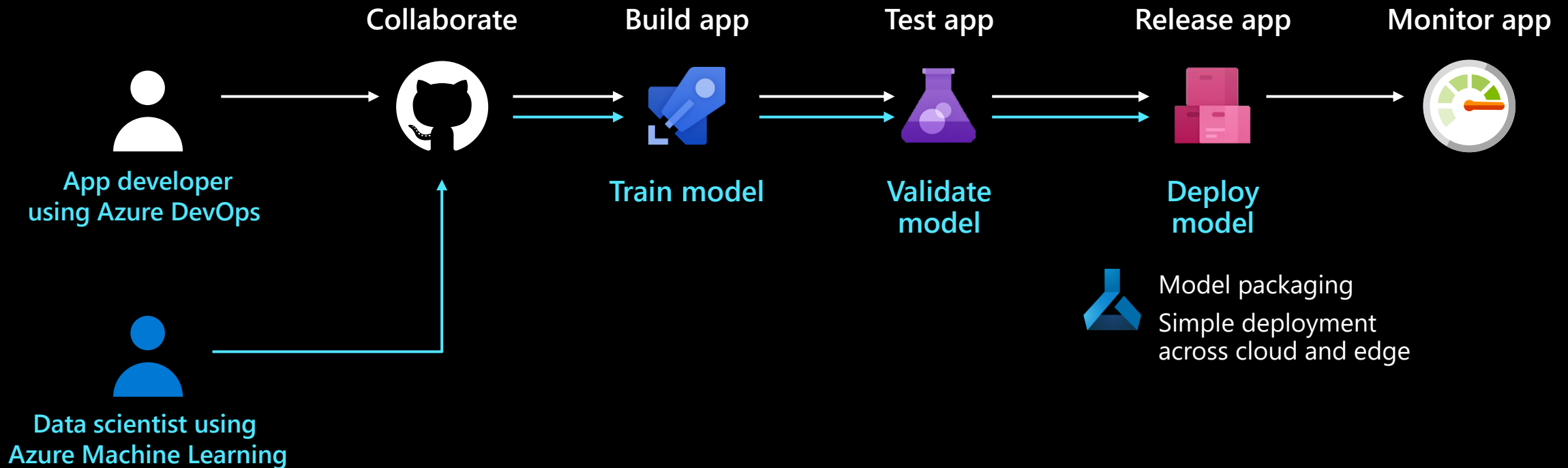
Model reproducibility

Model validation

Model deployment

Model retraining

MLOps with Azure Machine Learning



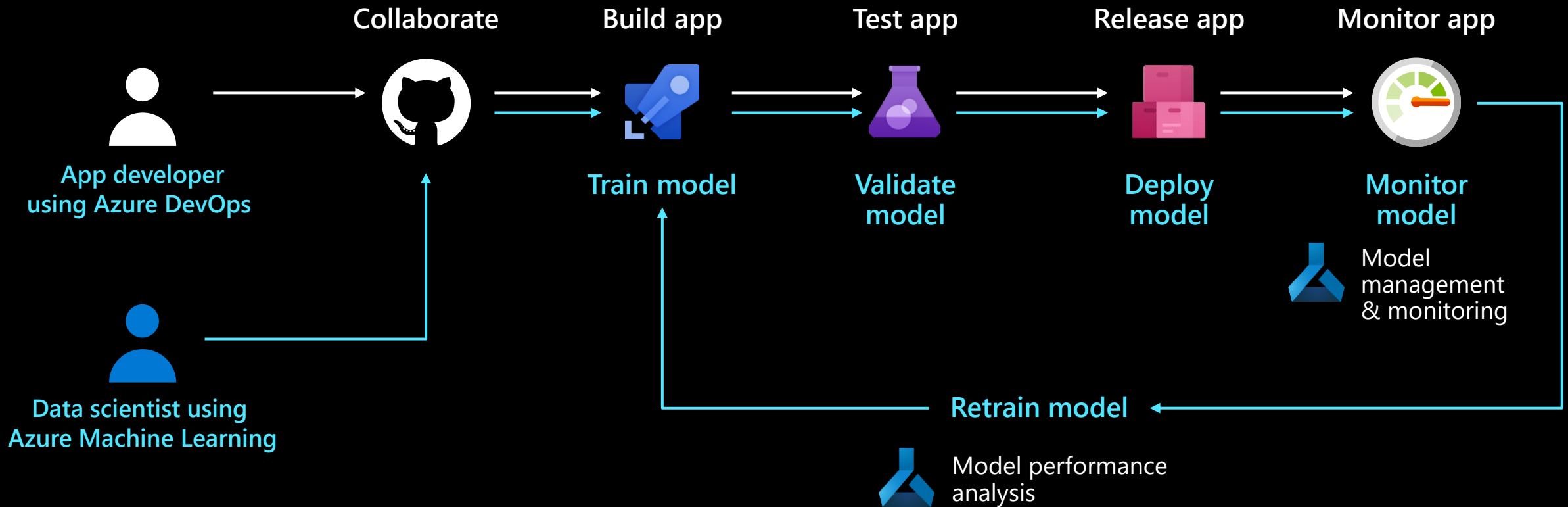
Model reproducibility

Model validation

Model deployment

Model retraining

MLOps with Azure Machine Learning



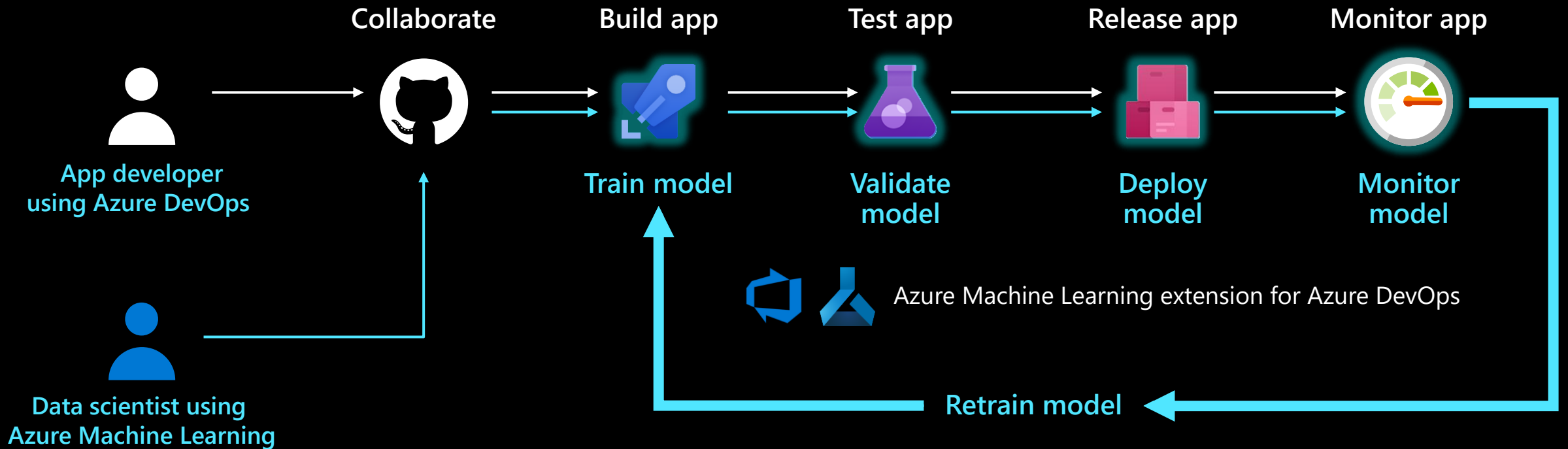
✓ Model reproducibility

✓ Model validation

✓ Model deployment

✓ Model retraining

MLOps with Azure Machine Learning



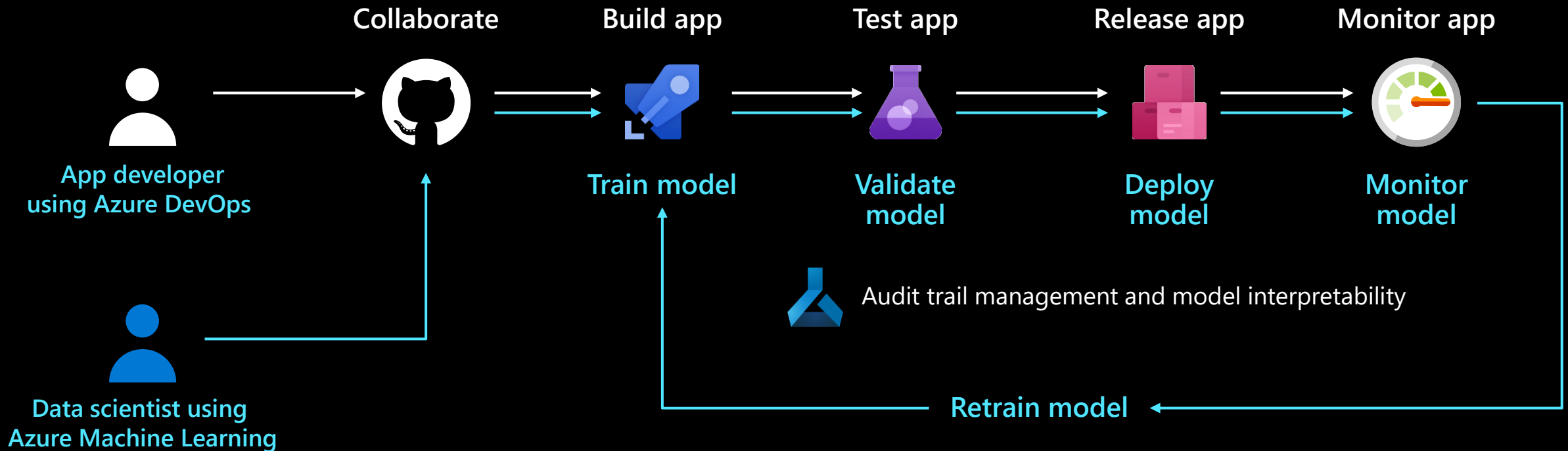
✓ Model reproducibility

✓ Model validation

✓ Model deployment

✓ Model retraining

MLOps with Azure Machine Learning



✓ Model reproducibility

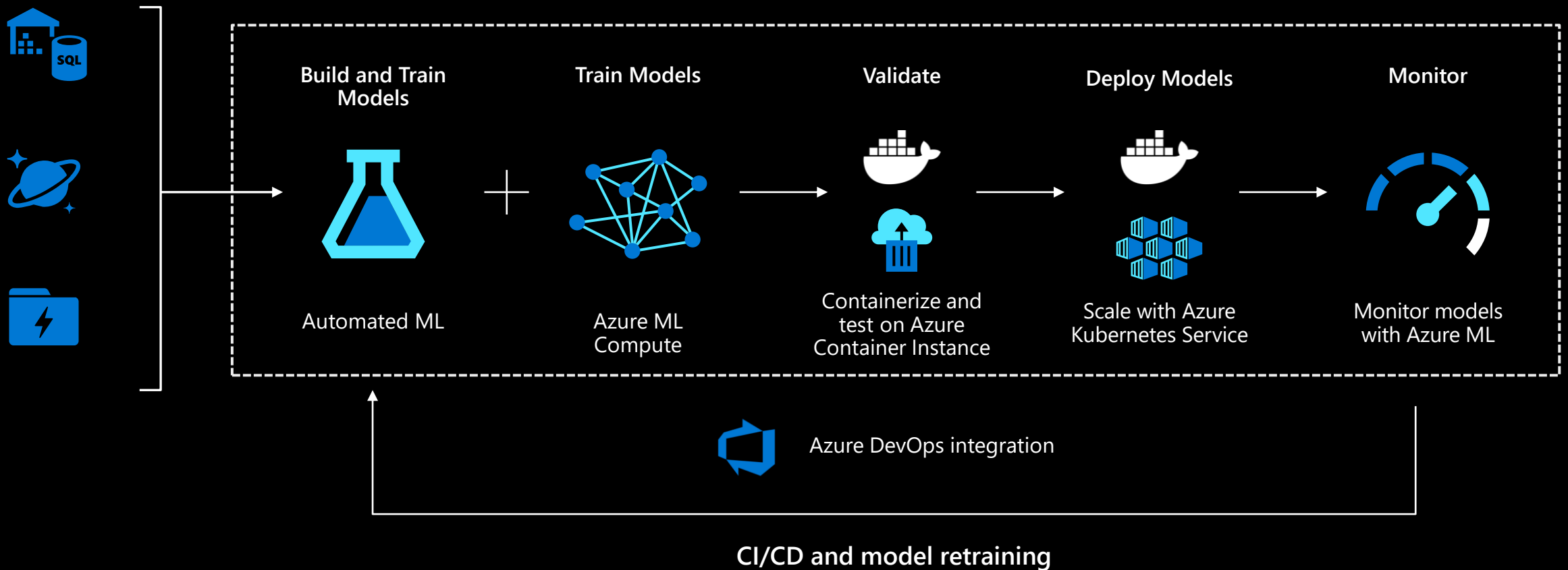
✓ Model validation

✓ Model deployment

✓ Model retraining

Deploy machine learning models at scale

Azure Machine Learning service



Demo

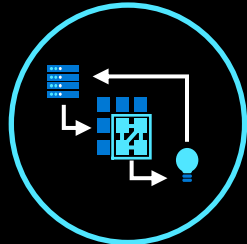
Deployment auf ACI

Summary

Azure Machine Learning



For all skill levels



Industry leading MLOps



Open & Interoperable



Trusted

Access machine learning for all skills and boost productivity.

Rapidly build and deploy machine learning models using tools that meet your needs regardless of skill level. Use the no-code designer to get started with machine learning or use built-in Jupyter notebooks for a code first experience. Accelerate model creation with the automated machine learning UI and access built-in feature engineering, algorithm selection, and hyperparameter sweeping, to develop high accuracy models.

Operationalize at scale with robust MLOps.

MLOps or DevOps for machine learning, streamlines the machine learning lifecycle, from building models to deployment and management. Use ML pipelines to build repeatable workflows and use a rich model registry to track your assets. Manage production workflows at scale using advanced alerts and automation capabilities. Profile, validate and deploy machine learning models anywhere from the cloud to the edge.

Innovate on an open and interoperable platform.

Take advantage of built-in support for popular open-source tools and frameworks for model training and inferencing. Use familiar frameworks like PyTorch, TensorFlow, scikit-learn and more, or the open and interoperable ONNX format. Choose the development tools that best meet your needs, including popular IDEs, Jupyter notebooks and CLIs or languages like Python and R. After you've built and trained your model, use ONNX Runtime to optimize and accelerate inferencing across cloud and edge devices.

Build responsible AI solutions on a secure trusted platform.

Access state-of-the-art technology for fairness and model transparency. Use model interpretability for explanations about predictions, to better understand model behavior. Reduce model bias by applying common fairness metrics, automatically making comparisons and using recommended mitigations. Enterprise-grade security with role based access control, and virtual network support to protect your assets. Audit trail, quota and cost management capabilities for advanced governance and control.

Links

- Azure AI Platform: <https://www.azure.ai>
- Azure Machine Learning: <https://azure.microsoft.com/en-us/services/machine-learning/>
- Cognitive Services: <https://azure.microsoft.com/en-us/services/cognitive-services/>
- Machine Learning Notebooks:
<https://github.com/Azure/MachineLearningNotebooks>

Thank you very much for your attention.



Vielen Dank für Eure Aufmerksamkeit.