



Azure MVP

  SamVanhoutte

Hi, I am Sam, CTO of Codit

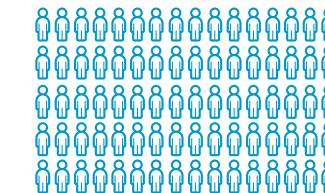
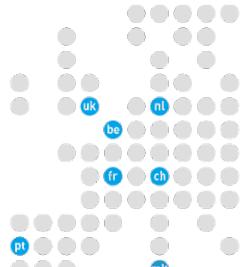
# Azure Machine Learning

Streamlining your Machine Learning projects with AzureML

# About Codit



2000 Belgium  
2004 France  
2013 Portugal  
2016 Switzerland  
2016 UK  
2016 The Netherlands  
2017 Malta  
2020 Luxemburg



195  
worldwide



Largest Microsoft partner in Europe for Data driven, integration, API management, IoT, AI and Azure Solutions



Gold Application Development  
Gold Application Integration  
Gold Cloud Platform  
Gold Data Analytics

# This session



Machine learning concepts



Azure Machine Learning



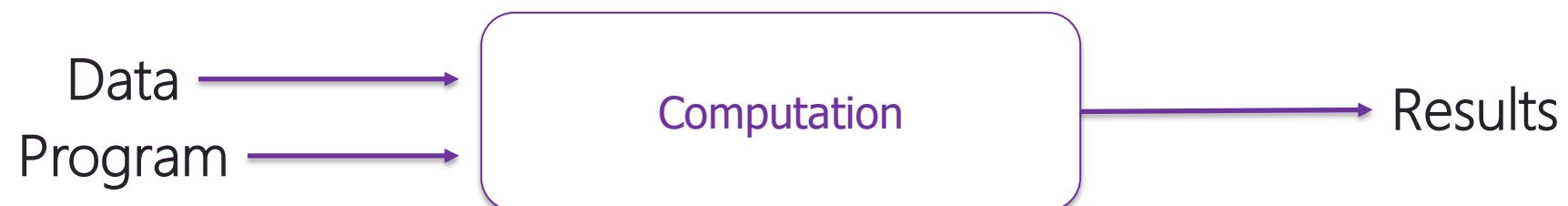
From experiment to service



The MLOps approach

# A different approach to problem solving

Traditional programming



Machine learning



# 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

## Experience

SDK, Notebooks, Drag&drop, Wizard

## MLOps

Reproducible, Reusable, Automatable, Git, CLI, REST

## Dataset mgmt

Profiling, drift, labeling



## Model training

Experiments, Runs



## Model mgmt

Models



## Model Serving

Batch, RealTime



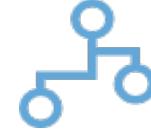
## Compute

Jobs, Clusters, Instances



## Orchestration

Security, management, deployment



## Cloud

CPU, GPU, FPGA



## Edge

CPU, GPU, NPU



# Experimenting like a pro

# The sample: MNIST

- | Handwritten digits
- | 60.000 images: 28x28px

> Classification problem



# Evaluation of a classifier

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

*Percentage of correct predictions*

$$\text{Recall} = \frac{TP}{TP + FN}$$

*How many of the actual samples in the classes were well classified (also called sensitivity)*

$$\text{Precision} = \frac{TP}{TP + FP}$$

*How many of the predicted samples in a class were well classified*

$$\text{F1 score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

*Combines recall & precision*

	p' (Predicted)	n' (Predicted)
P (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

# Free and maintained for you to use

## | Open source effort for the community

### | Arcus for Azure Machine Learning

- | Interactive experimenting (notebook), with centralized logging & snapshotting
- | Cloud training & experiments, with standard templates, logging and evaluations

### | Arcus for Machine Learning

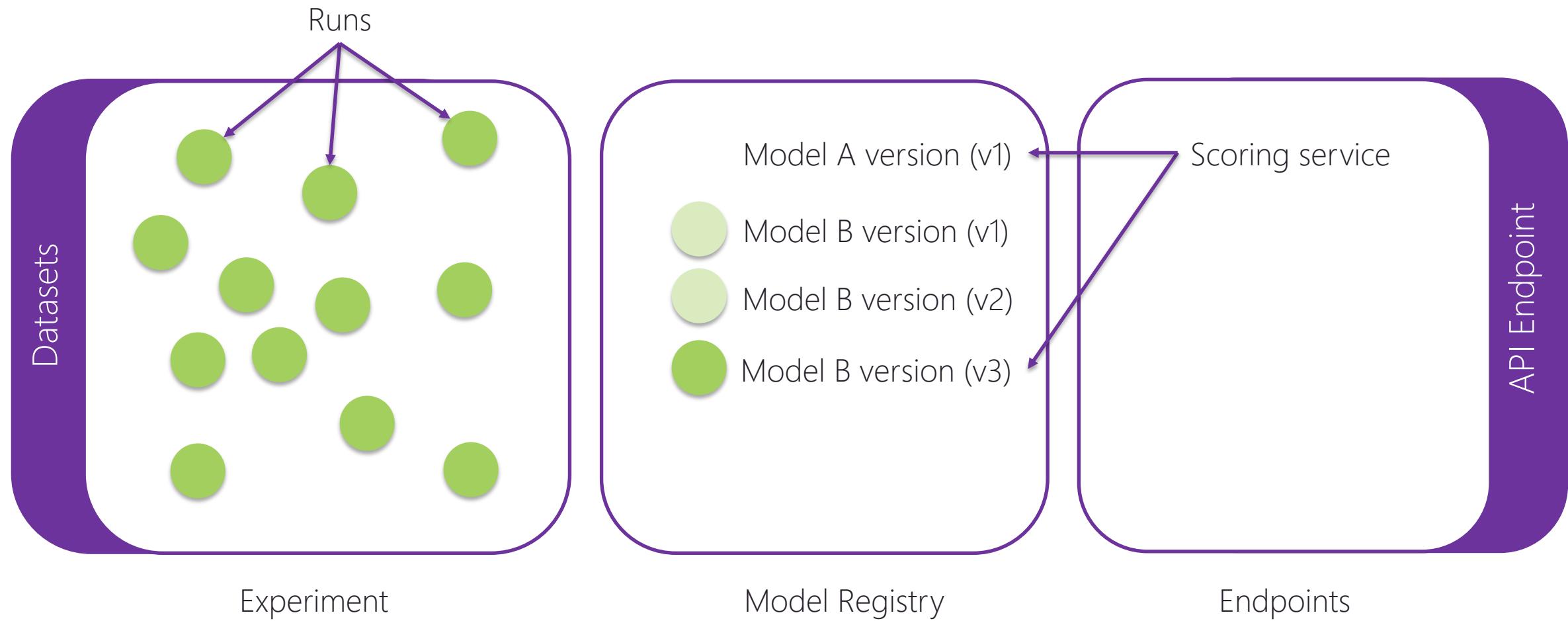
- | Common operations for DataFrames
- | Images
  - | IO: load from dataframe, Load from Url, Slice Images, Caching, Transform
  - | Explorer: Visualize Images, Compare images (expected/In/out)
- | Time Series: Time Windowing



# Training & experimenting approach

- | Experiment locally (Jupyter Notebook)
  - | Smaller datasets
  - | Focus on syntax and feasibility
  - | Immediate feedback, short training cycles
- | Script generation
- | Schedule trainings in the cloud
  - | Hyper parameters as script arguments
  - | Multiple runs in parallel, through scheduling queue by AzureML
  - | Docker image generation, based on required packages and sizing needs
  - | Every experiment gets logged & tracked

# Experiments, models & endpoints



# Azure Machine Learning experimenting

```
C:\Users\samva>demo01.exe_
```

# Training a model (train.py)

- | File that will be used as entry point for training image:
  - | Docker image, built using requirements (pip/conda)
  - | Train.py file started with training
- | Leverage arguments for dynamic input/behavior
  - | Hyper parameters
  - | Limit size of training data
- | Output metrics (to AzureML)
- | Save trained model to **outputs** folder.
- | [Example file](#)

# Servicing a model (score.py)

| File that will be hosted in the model endpoint (docker container) and should contain two methods:

| | **def init():**

- | | | Called once, when deployment is started
- | | | Used to load the model (often pkl file) and any other required objects
- | | | Using the get\_model\_path to locate model inside the docker container

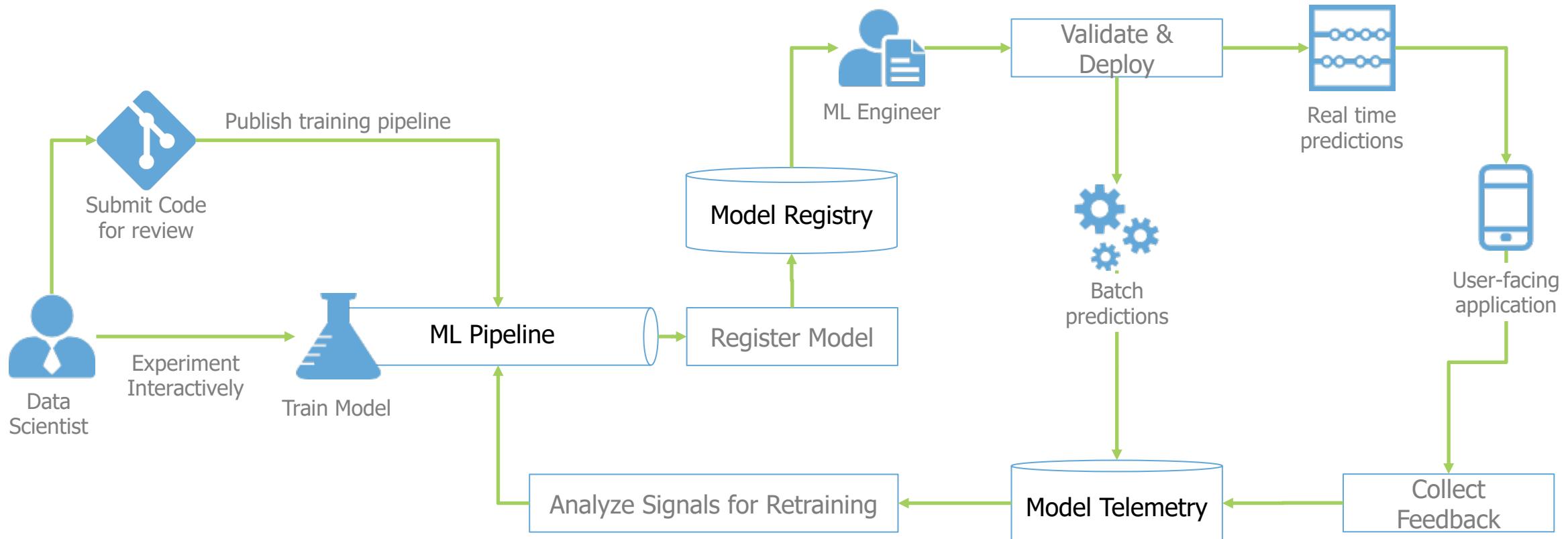
| | **def run(raw\_data):**

- | | | Called interactively when data samples are to be predicted
- | | | Raw\_data is json structure (with 'data' property)

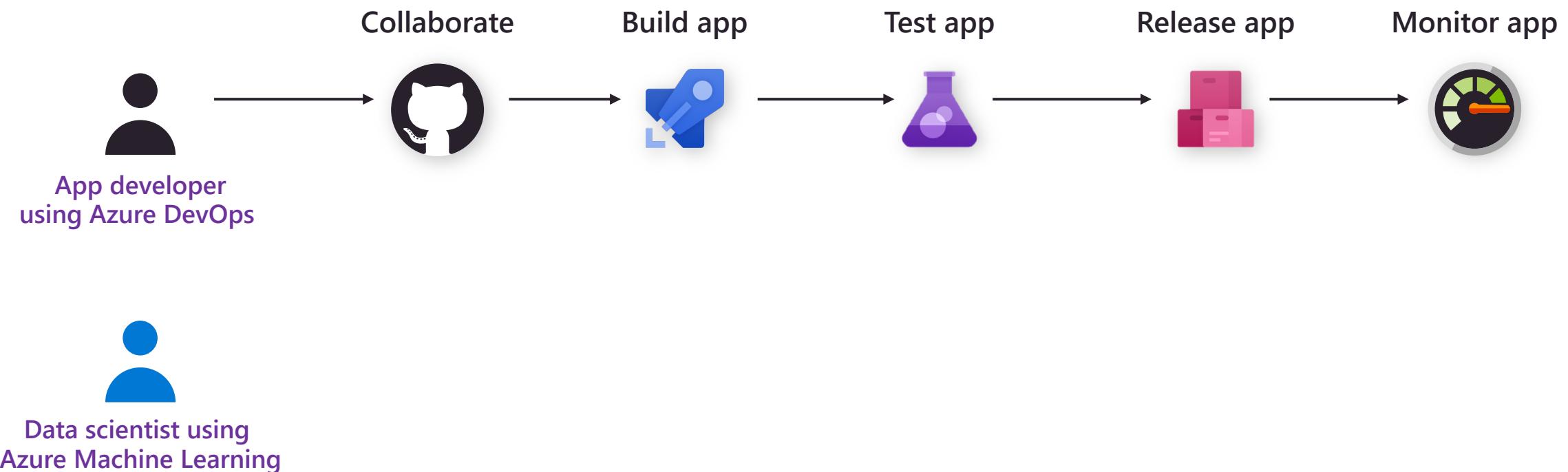
| Example file

# ML Ops

# MLOps process (generalized)



# MLOps with Azure Machine Learning



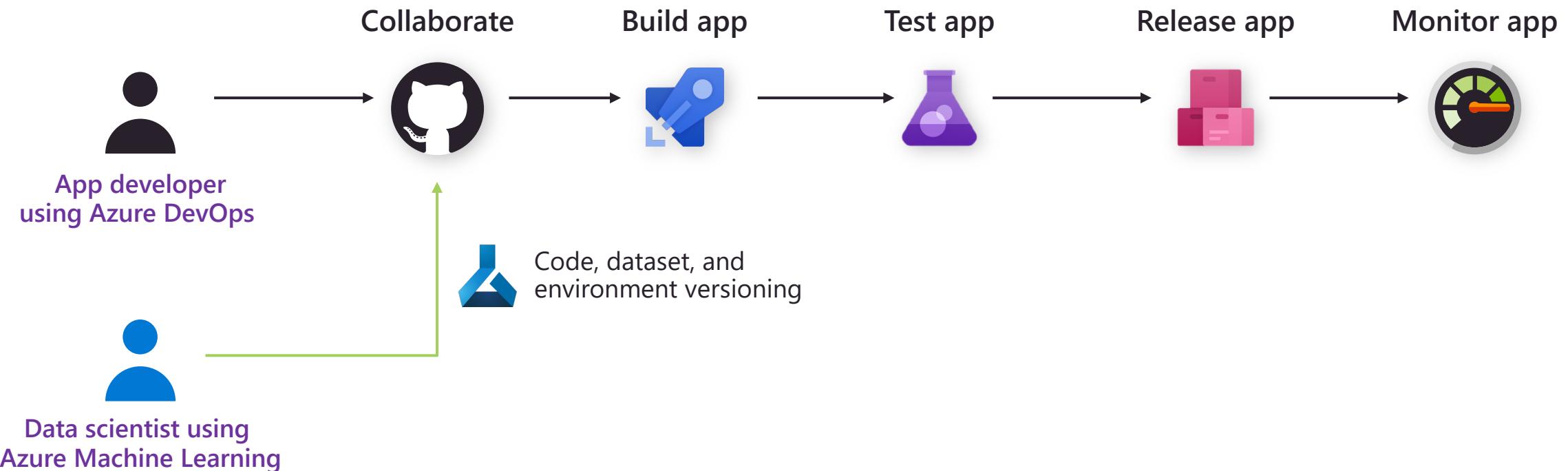
Model reproducibility

Model validation

Model deployment

Model retraining

# MLOps with Azure Machine Learning



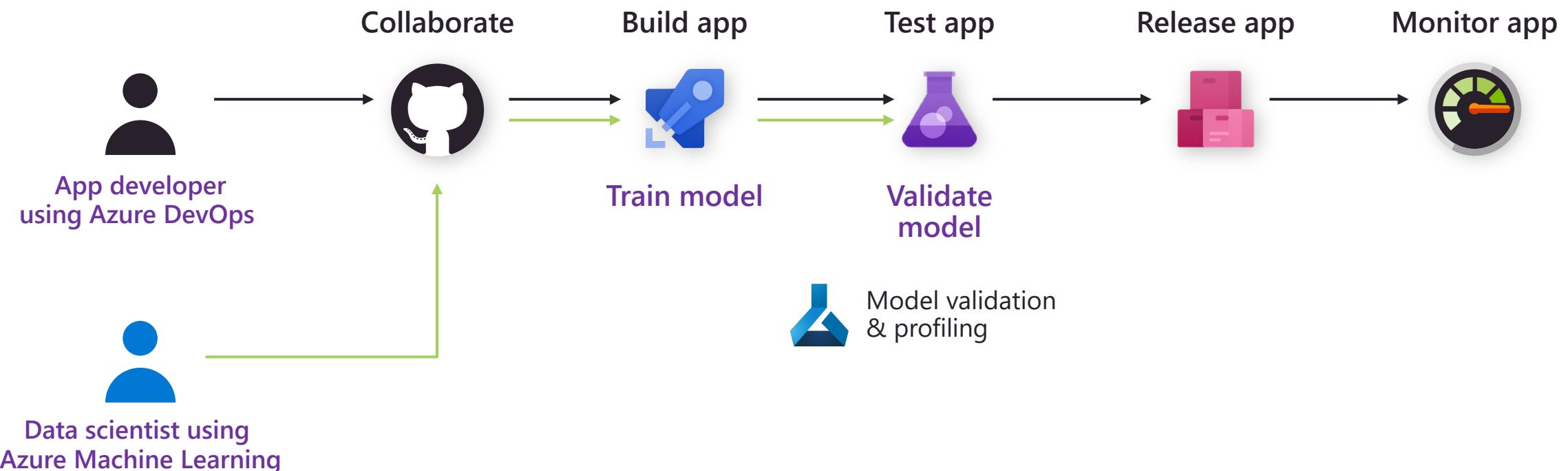
Model reproducibility

Model validation

Model deployment

Model retraining

# MLOps with Azure Machine Learning



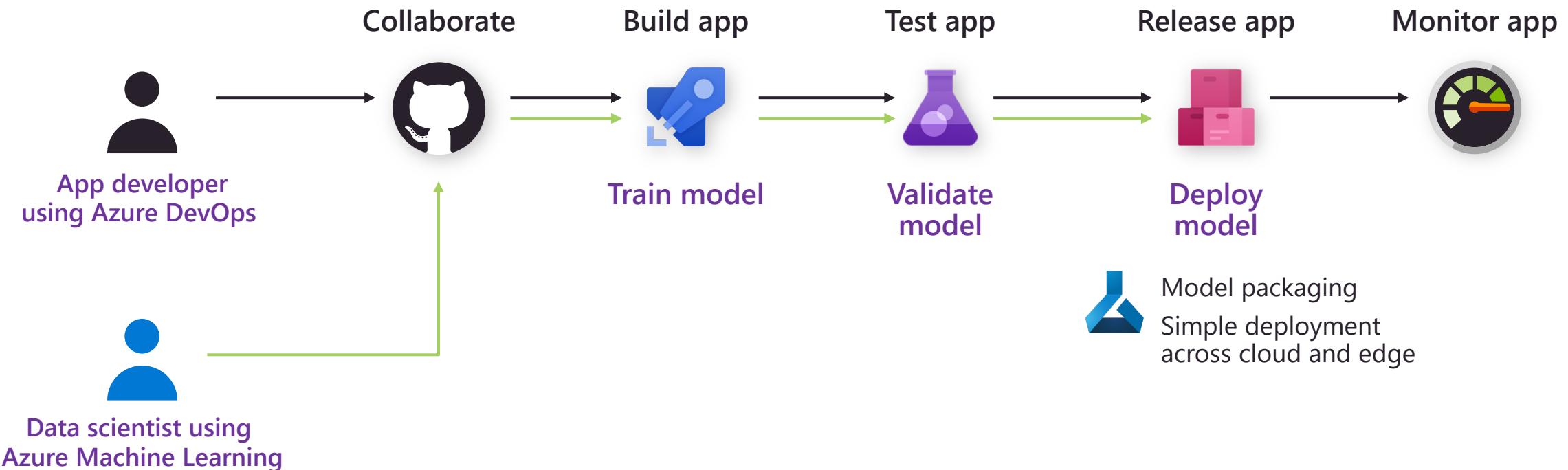
Model reproducibility

Model validation

Model deployment

Model retraining

# MLOps with Azure Machine Learning



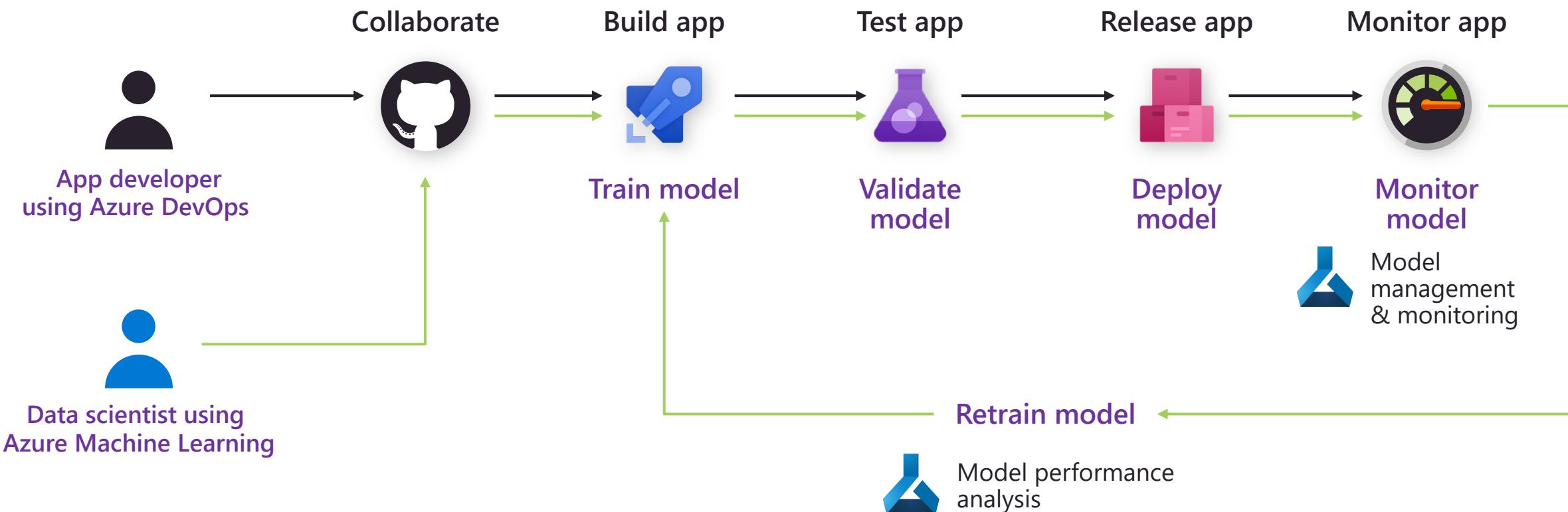
Model reproducibility

Model validation

Model deployment

Model retraining

# MLOps with Azure Machine Learning



Model reproducibility

Model validation

Model deployment

Model retraining

# Leveraging Azure DevOps : CI (training pipeline)

## Build Model

- | Linting (code quality analysis)
- | Unit Tests
- | Code coverage
- | Build & publish ML pipeline in ML Workspace
- | Publish artifact (model)

## Train model

- | Trigger ML pipeline
- | Verify metrics of the model (cancel if not better)

## Create pipeline artifact

- | Create artifact (model.json) with the model information



## Machine Learning

Microsoft DevLabs | 3,999 installs | ★★★★★ (1) | Free

Submit experiments from a DevOps Pipeline, track code from Azure Repos or GitHub, trigger release pipelines when an ML model is registered, and automate ML deployments using Azure Pipelines.

Get it free

# Leveraging Azure DevOps : CD (model deploy)

- | Deploy model to ACI
- | Deploy model to AKS
- | Deploy model to AppService



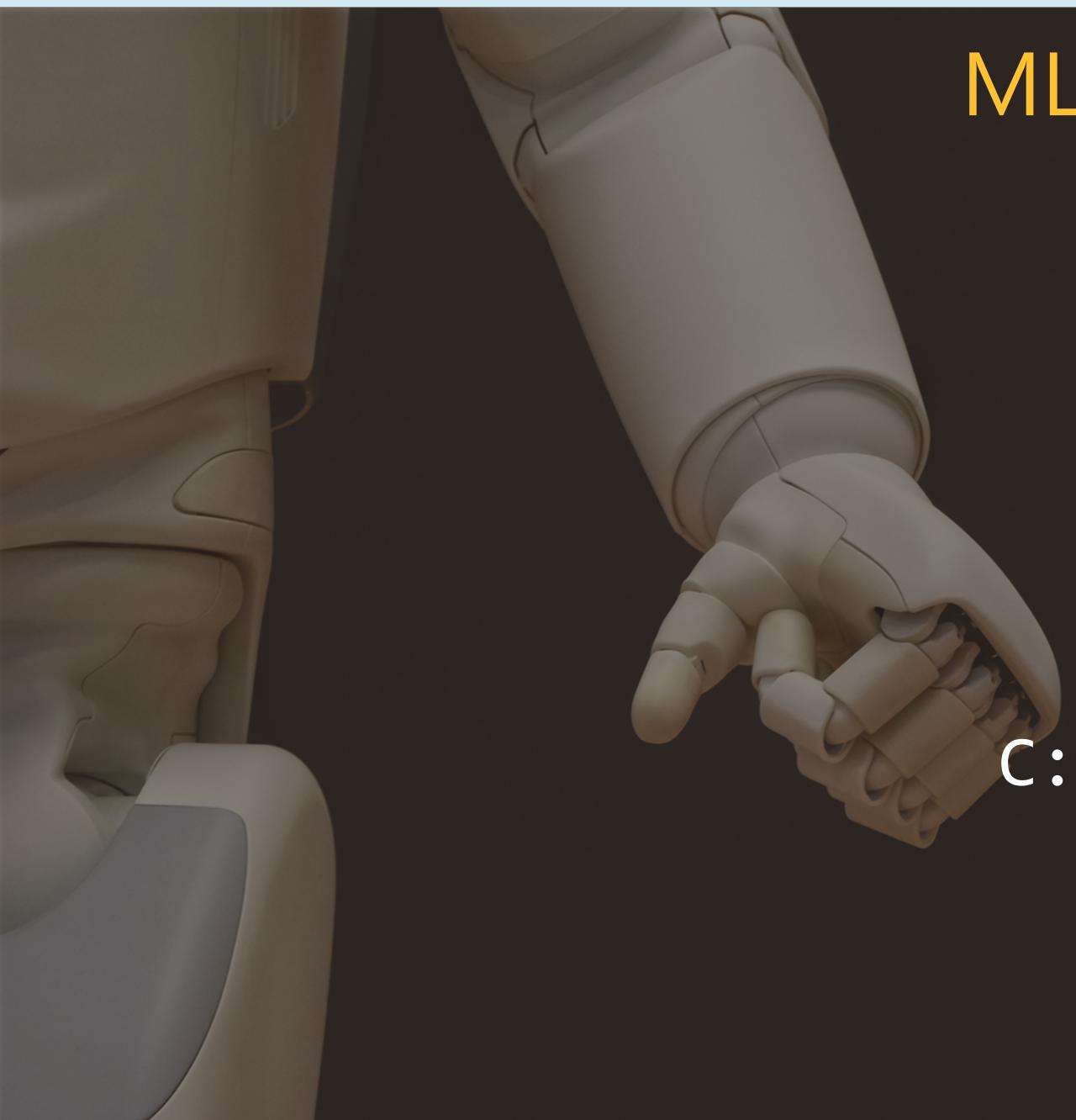
**Machine Learning**  
Microsoft DevLabs | 3,999 installs | ★★★★★ (1) | Free

Submit experiments from a DevOps Pipeline, track code from Azure Repos or GitHub, trigger release pipelines when an ML model is registered, and automate ML deployments using Azure Pipelines.

[Get it free](#)

# MLOps with Azure DevOps from repo to service

C:\Users\samva>demo02.exe\_



# Takeaways

AzureML drives collaboration & traceability of your ML projects

- | Track your experiments (code, results, hyper params)
- | Combine flexibility, ease of use and enterprise quality
- | Treat your machine learning models like code, and leverage the MLOps approach

<https://www.arcus-azure.net>

<https://github.com/SamVanhoutte/azureml-mlops>

<https://github.com/microsoft/MLOpsPython>

Thank you. Let's connect!