Microsoft

Microsoft Tech Briefings:

Put MLOps into Practice

Microsoft

Speakers

Xiaopeng Li

AI Business Lead in Western Europe, Microsoft

João Pedro Martins (Jota)

AI Rangers Lead for EMEA & Asia, Microsoft

Microsoft

Agenda

- Why MLOps matters
- What is MLOps
- How to put MLOps into practice
- Q&A and Closing

Why MLOps matters

Xiaopeng Li

The pace of AI advancements is increasing

Vision

2016

Object recognition human parity

Speech Recognition

2017

Speech recognition human parity

Reading

2018

Reading comprehension human parity

Translation

2018

Machine translation human parity

Speech Synthesis

2018

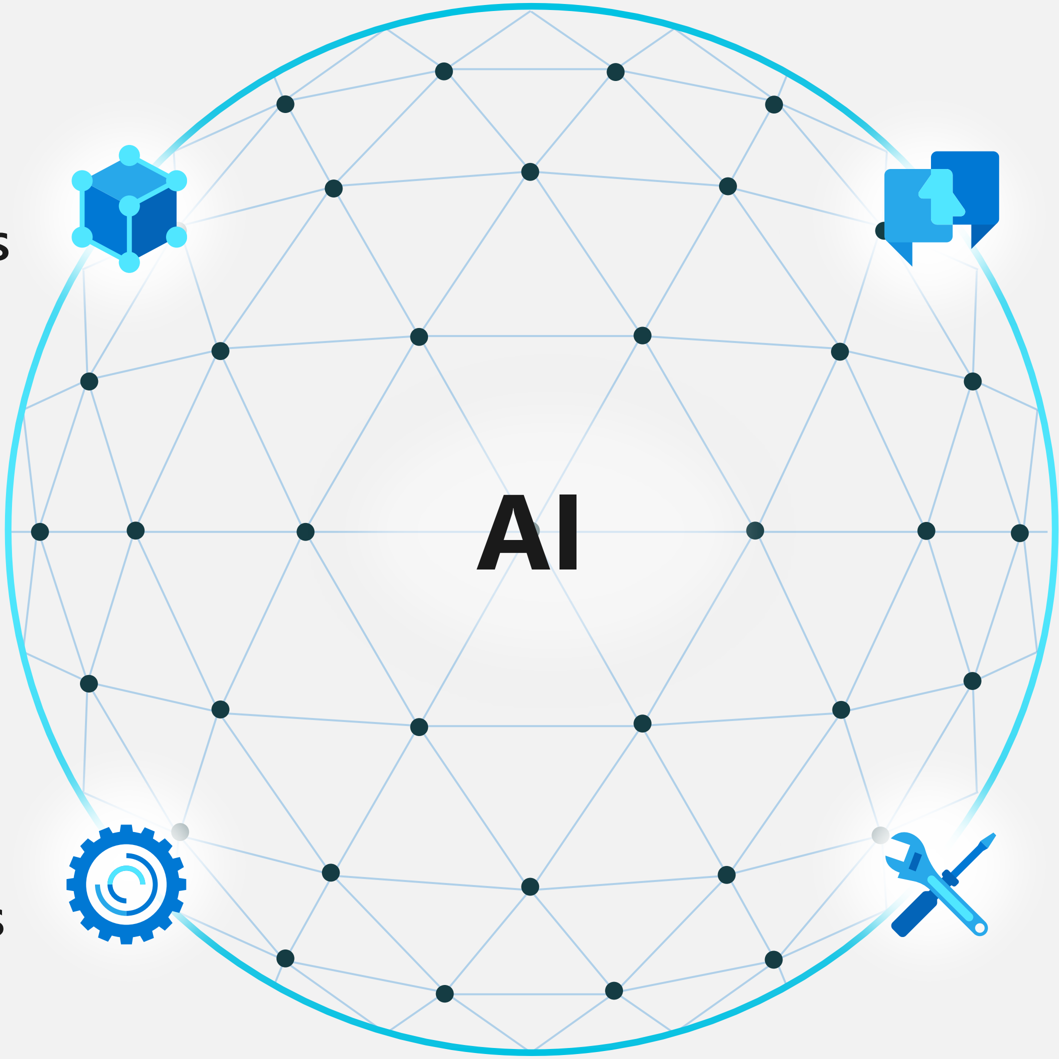
Speech synthesis near-human parity

Language Understanding

2019

General Language Understanding human parity

AI is fueling innovations across the organization



Empower employees

Engage customers

Optimize operations

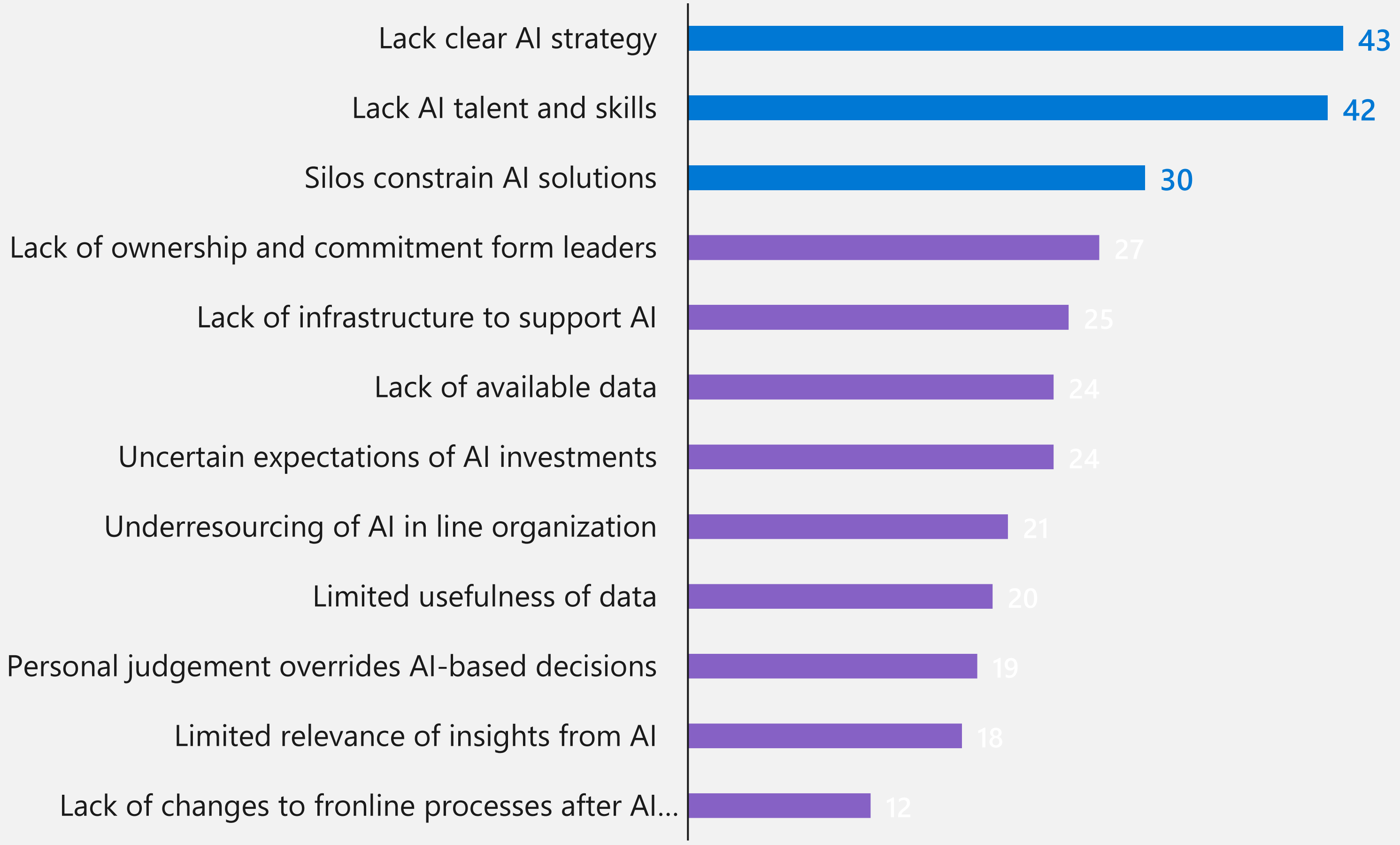
Transform products

Top barriers to adopting and scaling AI

43% Lack of clear strategy for AI

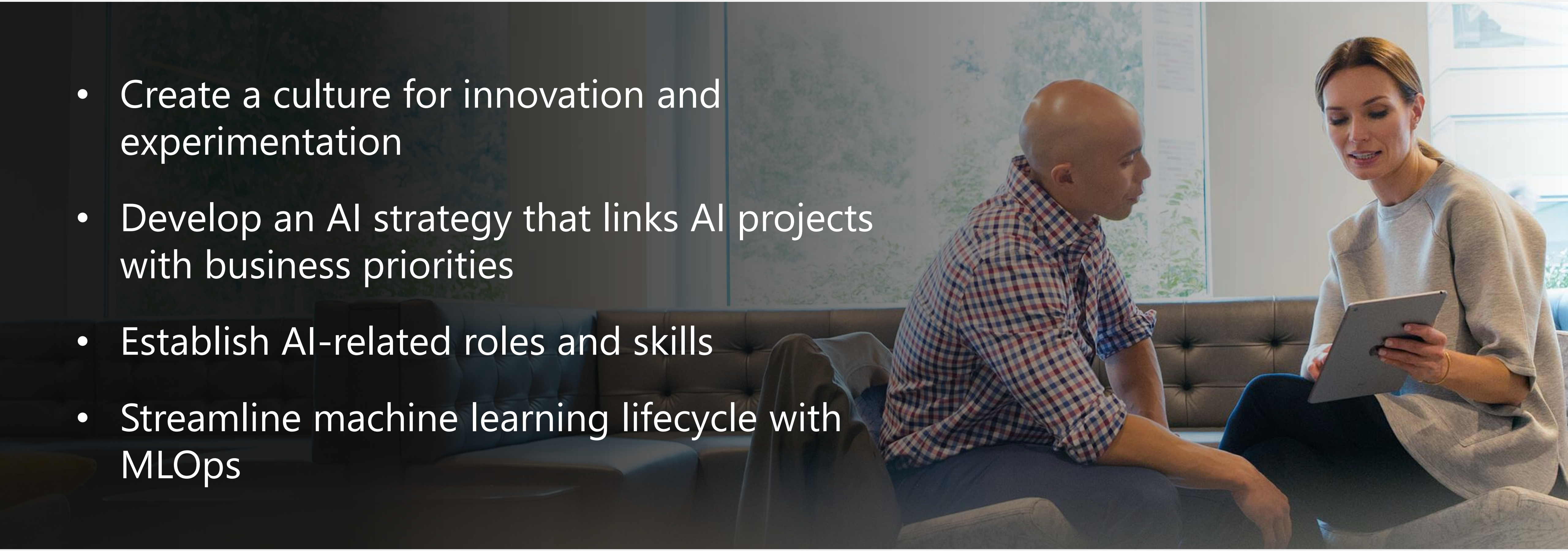
42% Lack of talent with appropriate skill sets for AI work

30% Functional silos constrain end-to-end AI solutions



Source: AI adoption advances, but foundational barriers remain. (2018) from https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain

Best practices for scaling AI throughout your organization



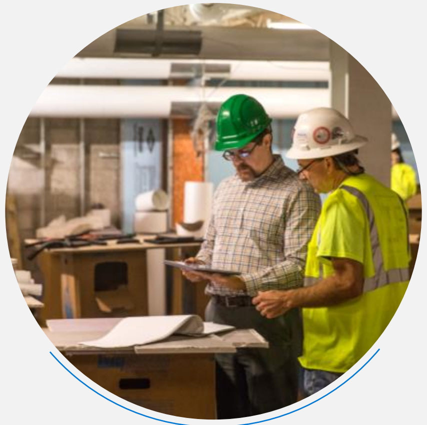
- Create a culture for innovation and experimentation
- Develop an AI strategy that links AI projects with business priorities
- Establish AI-related roles and skills
- Streamline machine learning lifecycle with MLOps

Why organizations are adopting MLOps

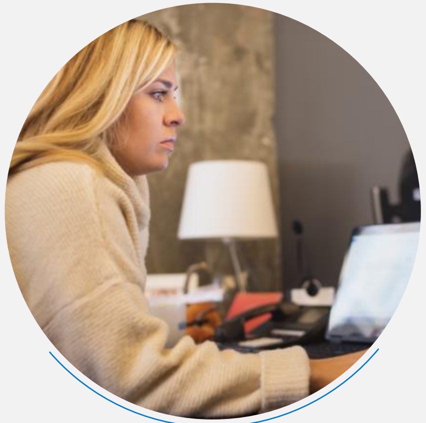
Facilitate better business results

Improve alignment across teams

Enable faster time to market



Assure model quality and auditability



Accelerate experimentation

Create and manage the machine learning lifecycle with MLOps

João Pedro Martins

MLOps = How to bring ML to production

Bring together **people**, **process**, and **platform** to automate ML-infused software delivery & provide continuous value to organizations.

People

- Blend together the work of individual engineers in a repository.
- Each time you commit, your work is automatically built and tested, and bugs are detected faster.
- Code, data, models and training pipelines are shared to accelerate innovation.

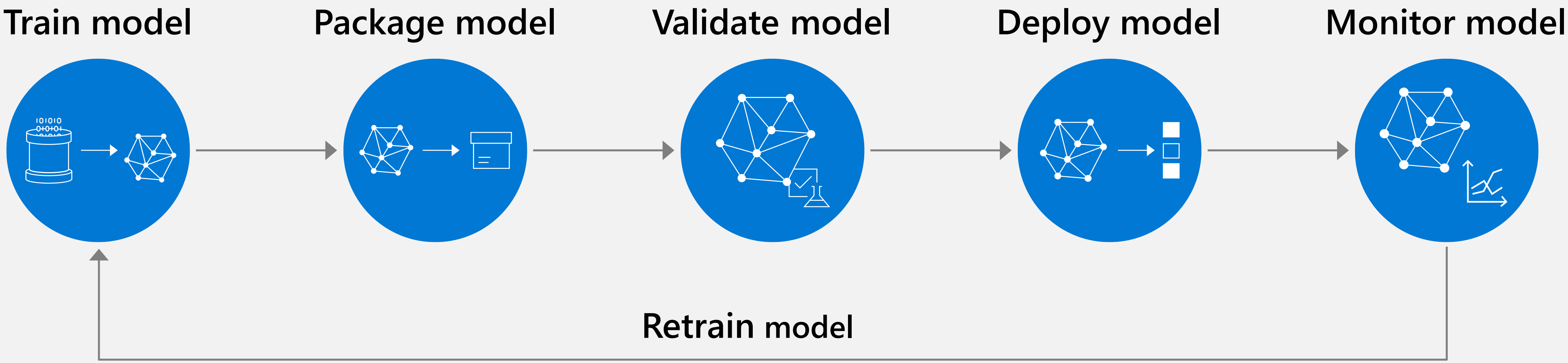
Process

- Provide templates to bootstrap your infrastructure and model development environment, expressed as code.
- Automate the entire process from code commit to production.

Platform

- Safely deliver features to your customers as soon as they're ready.
- Monitor your pipelines, infrastructure and products in production and know when they aren't behaving as expected.

Typical high-level Machine Learning (ML) lifecycle



Train model

Package model

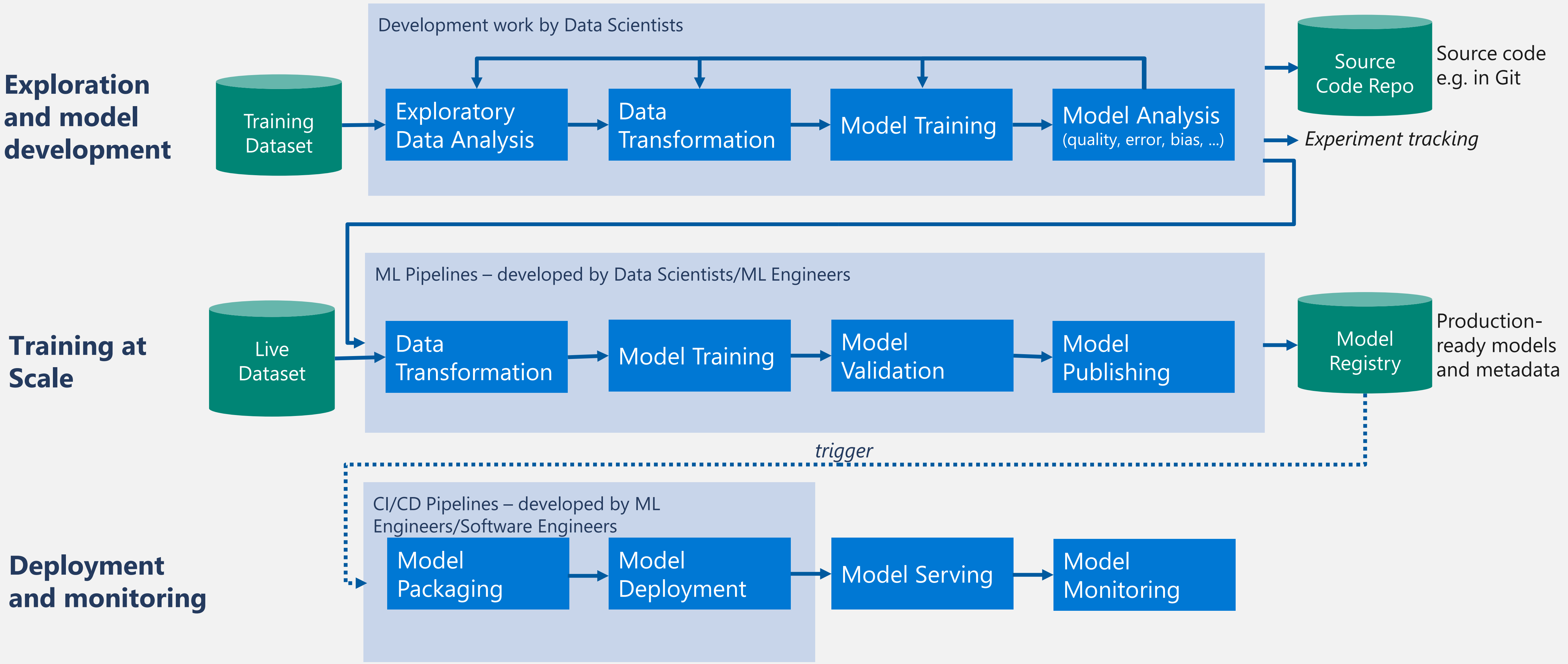
Validate model

Deploy model

Monitor model

Retrain model

Breaking down the Machine Learning (ML) lifecycle



Exploration and model development

Training Dataset

Development work by Data Scientists

Exploratory Data Analysis

Data Transformation

Model Training

Model Analysis
(quality, error, bias, …)

Source Code Repo

Source code e.g. in Git

Experiment tracking

Training at Scale

Live Dataset

ML Pipelines – developed by Data Scientists/ML Engineers

Data Transformation

Model Training

Model Validation

Model Publishing

Model Registry

Production-ready models and metadata

trigger

Deployment and monitoring

CI/CD Pipelines – developed by ML Engineers/Software Engineers

Model Packaging

Model Deployment

Model Serving

Model Monitoring

Microsoft Azure AI

Why can MLOps be a challenge?

Roles/Skills

Data Scientists

ML Engineers

Software Developers

Infra & Security teams

Tools

Notebooks/R Studio/VS Code/Visual Studio

Machine Learning services

GitHub/GitHub Actions/Azure DevOps

Kubernetes/Container-Hosting

Environment deployment templates

Artifacts & Versioning

Source Code

Data (schema + snapshots)

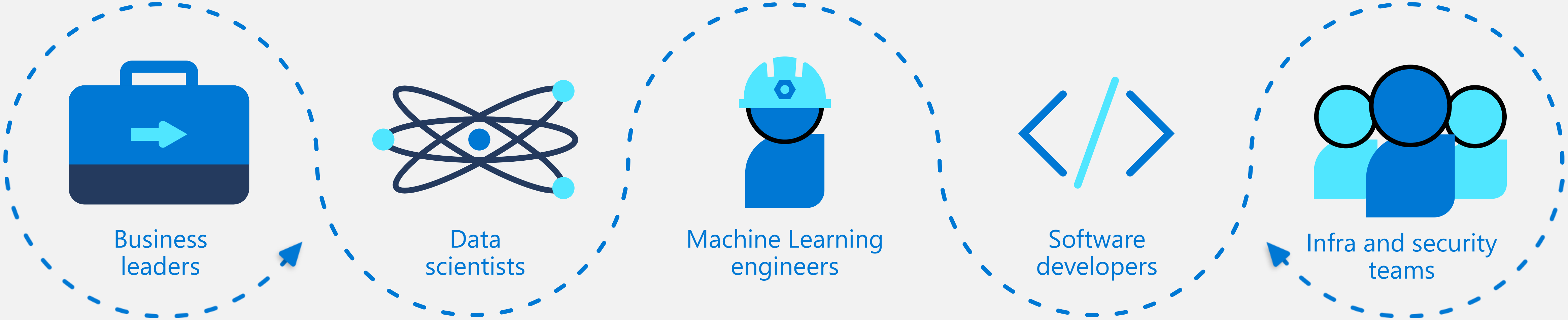
Model Registry

Development/Production Disconnect

What is developed is not what is deployed.

Production models need monitoring and retraining.

Four key aspects of scale AI across the organization



Business leaders

Data scientists

Machine Learning engineers

Software developers

Infra and security teams

Model reproducibility

Model validation

Model deployment

Model retraining

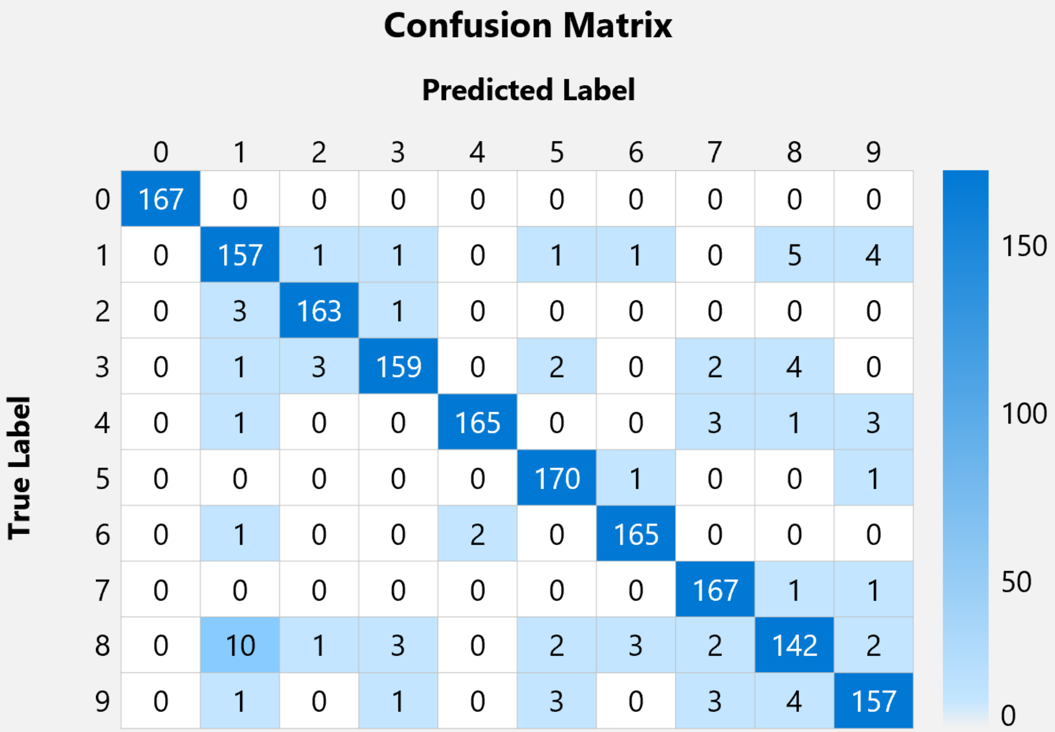
Model reproducibility

Centrally manage assets
Models | Code (&environments) | Data

Create machine learning pipelines

Model validation

An ML model with high accuracy



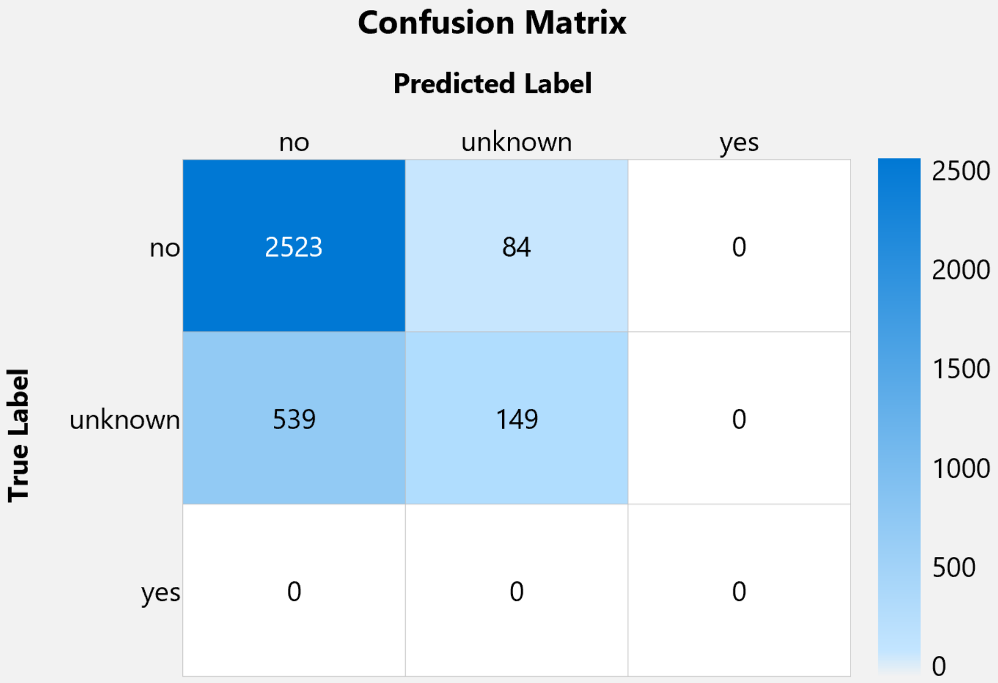
Confusion Matrix

Predicted Label

True Label

	0	1	2	3	4	5	6	7	8	9
0	167	0	0	0	0	0	0	0	0	0
1	0	157	1	1	0	1	1	0	5	4
2	0	3	163	1	0	0	0	0	0	0
3	0	1	3	159	0	2	0	2	4	0
4	0	1	0	0	165	0	0	3	1	3
5	0	0	0	0	0	170	1	0	0	1
6	0	1	0	0	2	0	165	0	0	0
7	0	0	0	0	0	0	0	167	1	1
8	0	10	1	3	0	2	3	2	142	2
9	0	1	0	1	0	3	0	3	4	157

An ML model with high accuracy and high bias in model predictions



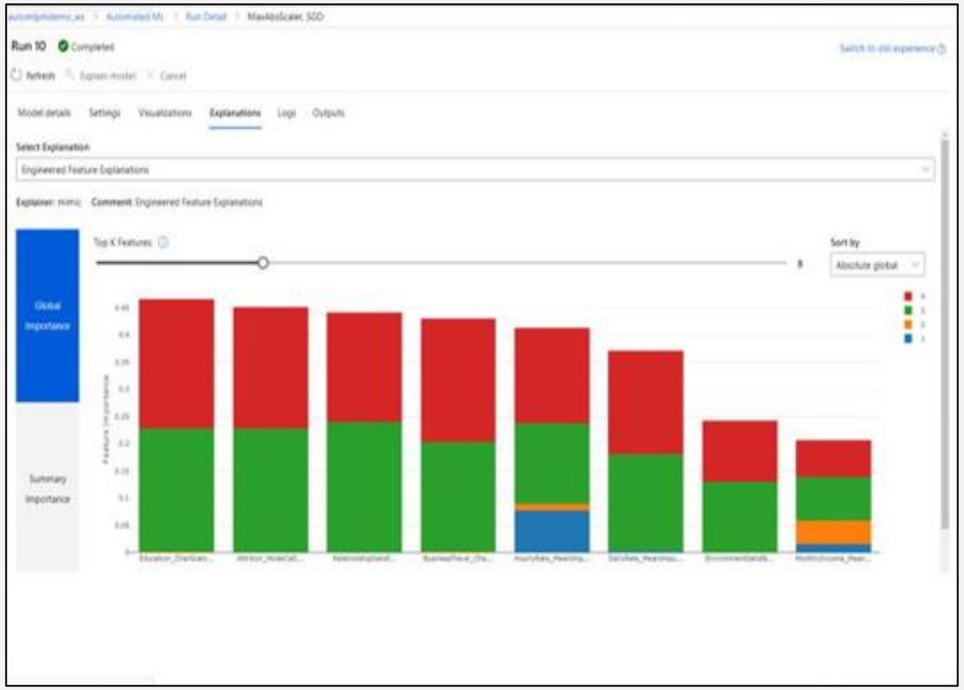
Confusion Matrix

Predicted Label

True Label

	no	unknown	yes
no	2523	84	0
unknown	539	149	0
yes	0	0	0

Model validation – interpretability/fairness

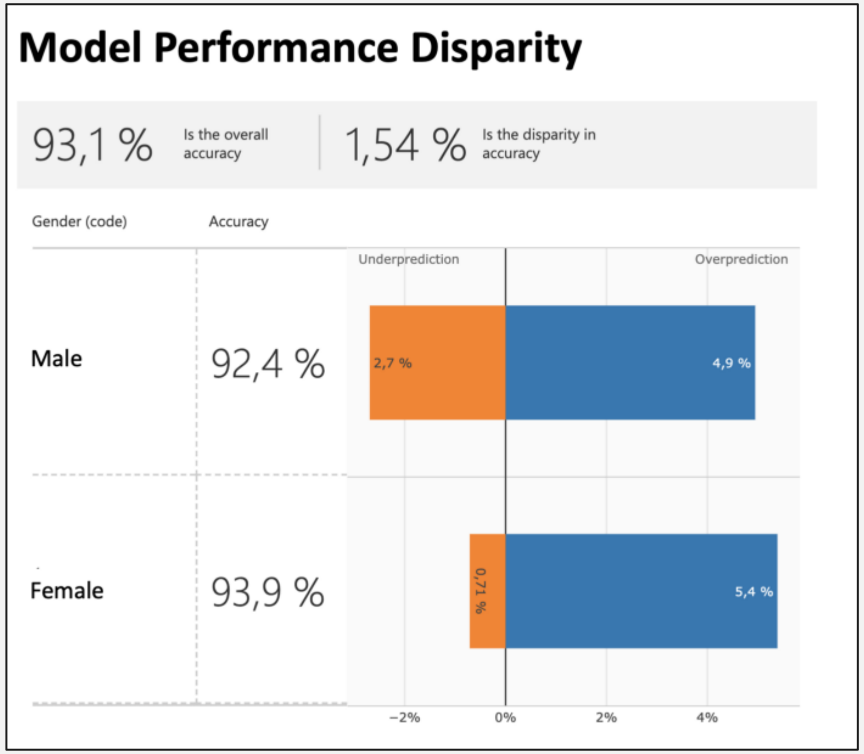


Select Explanation: Engineered Feature Explanations

Explainer: mimic Comment: Engineered Feature Explanations

Sort by: Absolute global

Model Interpretability



Fairness assessment & unfairness mitigation

https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability
https://github.com/microsoft/responsible-ai-toolbox :: https://github.com/slundberg/shap :: https://github.com/marcotcr/lime :: https://github.com/interpretml/interpret

Model deployment

1

Register the model

2

Prepare to deploy

Create and Publish a Container Image with all the dependencies

3

Deploy the model

Typically to container-hosting environments and requiring specialized SW Engineering/Infra knowledge

4

Test and monitor

Model reproducibility | Model validation | **Model deployment** | Model retraining

Monitoring of models and data

Automatic monitoring with established **performance thresholds on models** and/or **statistical tests on data**, that trigger alerts

Manual spot check, where someone tests the model or explores the data and does their own analysis

Model monitoring and retraining

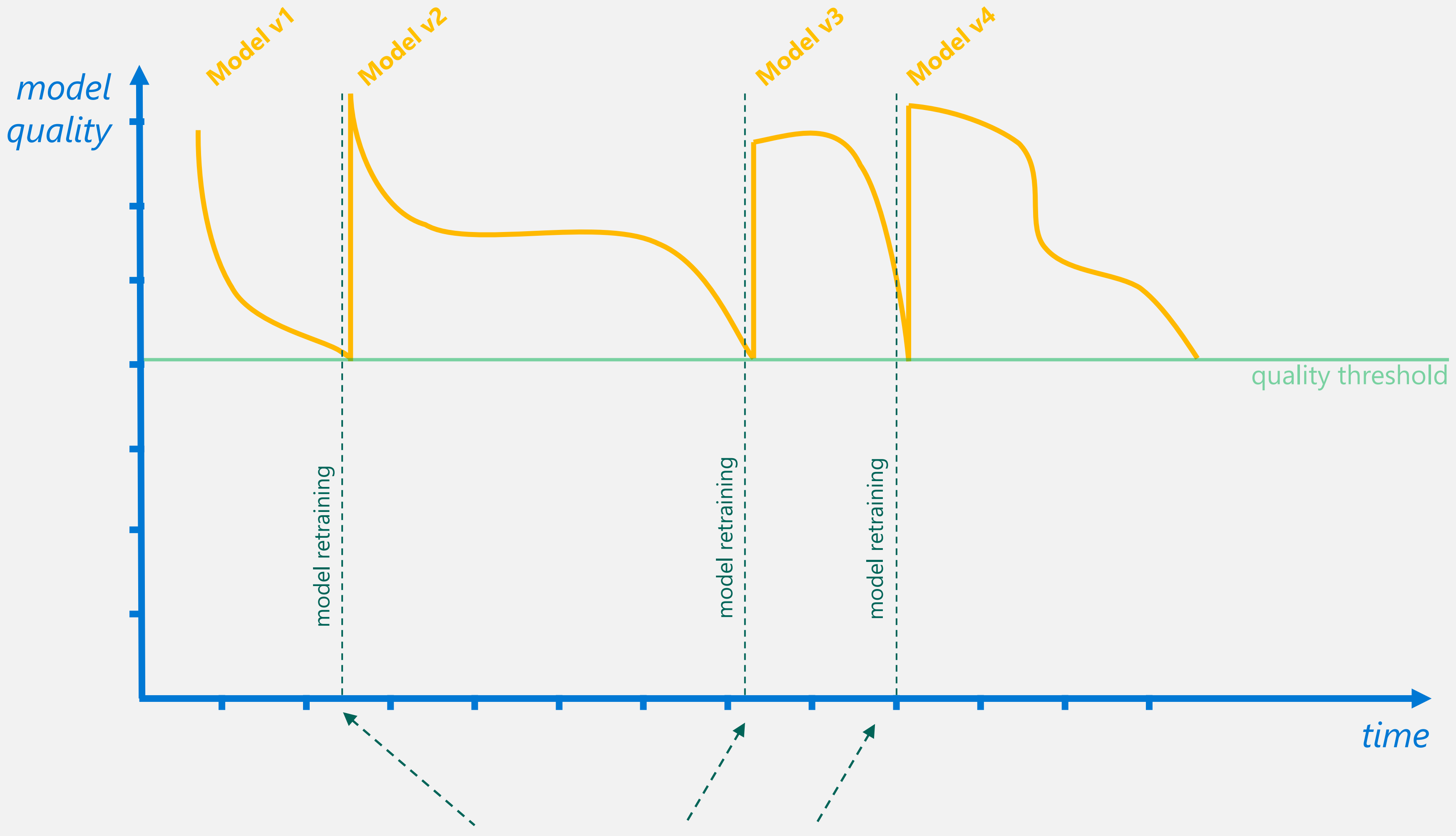
Why do ML models need to be monitored and retrained?

- Live data is different from training data
- Data characteristics change
- Data quality issues with live data
- Address fairness or bias issues
- The model "drifts" or declines over time
- The world changes

Best practices

- Schedule regular retraining to take advantage of "fresher" training data
- Retrain when there is a decline in performance
- Automate the detection of performance decline
- Use a CI/CD pipeline and A/B testing
- Keep a human "in the loop"

Model decay and retraining



(triggered by model drift detection)

How to put MLOps into practice

João Pedro Martins

MLOps Maturity Stages

Maturity Level	Training Process	Release Process	Integration into app
Level 1 – No MLOps	Untracked, file is provided for hand-off	Manual, hand-off	Manual, heavily Data Science (DS) driven
Level 2- Training Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Manual release, clean hand-off process, managed by Software Engineering team	Manual, heavily DS driven, basic integration tests added
Level 3 – Release Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Automated, CI/CD pipeline set up, everything is version controlled	Semi-automated, unit and integration tests added, still needs human signoff
Level 4 – Training & Release Operationalized Together	Tracked, run results and model artifacts are captured in a repeatable way, **retraining set up** based on metrics from app	Automated, CI/CD pipeline set up, everything is version controlled, A/B testing has been added	Semi-automated, unit and integration tests added, **may** need human signoff

Set up a team with the required skills

- **Data Science** – capable of doing data exploration/transformation and training models.

- **Machine Learning Engineering –** produce production-ready code from data science outputs (e.g., ML pipelines)

- **Software Engineering** – develop CI/D pipelines

- **Security** – Data is valuable and often confidential. From training to serving, infrastructures must be secure and minimize risks like data exfiltration

- **Infrastructure** – Deploying new models may imply deploying new cloud services in particular networking environments

Identify and prioritize the requirements and KPIs

Identify and prioritize the set of **requirements** for successive versions of a MLOps setup, e.g.:

- Security – e.g., can data scientists use production data to train?
- Model Retraining – e.g., on a schedule/manual/event
- Online vs Batch Inferencing
- Target model authors – professional or citizen data scientist?
- Fairness / Bias analysis – Is there any risk of harm? Are protected classes considered?
- Data Volumes for training
- A/B testing before deployment
- Data drift detection – based on data used for inference
- Model drift detection – based on model predictions

Identify the main **KPIs** for the platform, e.g.:

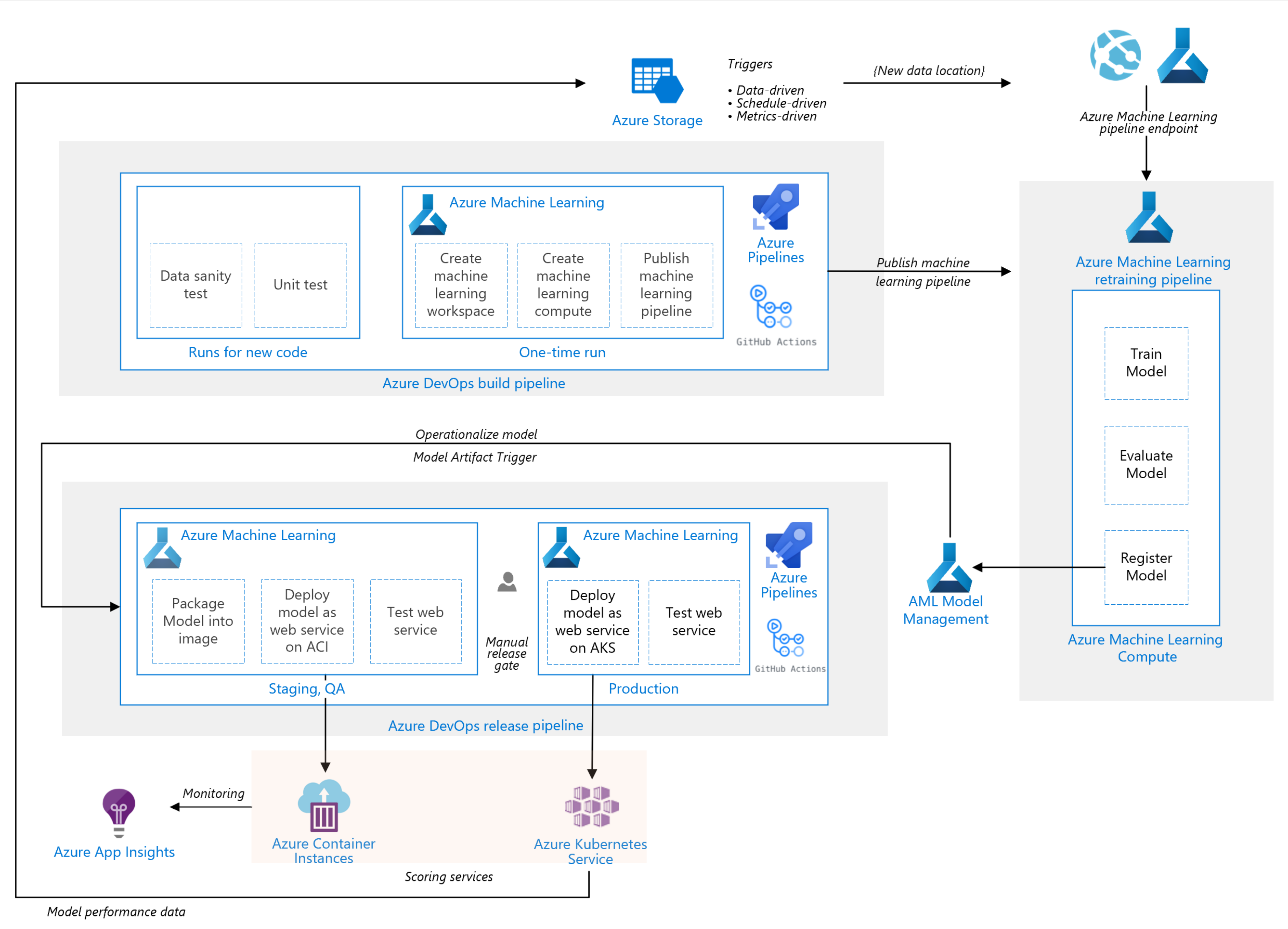
- Time from model trained to deployment
- Time from model request to data access

Technical Architecture Design

Azure Machine Learning
Azure DevOps/GitHub Actions
Azure Synapse
Azure Kubernetes Service
Azure Container Instances
Azure AppService
Azure AppInsights
Azure Data Lake Storage
PowerBI

Also:

AML SDK v1.0/preview 2.0
Managed Endpoints preview



Azure Storage

Triggers

- *Data-driven*
- *Schedule-driven*
- *Metrics-driven*

{New data location}

Azure Machine Learning pipeline endpoint

Data sanity test

Unit test

Runs for new code

Azure Machine Learning

Create machine learning workspace

Create machine learning compute

Publish machine learning pipeline

One-time run

Azure Pipelines

GitHub Actions

Azure DevOps build pipeline

Publish machine learning pipeline

Azure Machine Learning retraining pipeline

Train Model

Evaluate Model

Register Model

Azure Machine Learning Compute

AML Model Management

Operationalize model

Model Artifact Trigger

Azure Machine Learning

Package Model into image

Deploy model as web service on ACI

Test web service

Staging, QA

Manual release gate

Azure Machine Learning

Deploy model as web service on AKS

Test web service

Production

Azure Pipelines

GitHub Actions

Azure DevOps release pipeline

Monitoring

Azure App Insights

Azure Container Instances

Azure Kubernetes Service

Scoring services

Model performance data

Putting it together – from Level 1 to increased maturity

Secure Business backing and requirements

Set up a team with the required skills
Identify and prioritize technical requirements
Design a Secure Technical Architecture including inbound data flows
Implement the Technical Architecture
MVP with an existing model that is production-ready

Iterate to increase maturity level / add requirements

Customer Examples

João Pedro Martins

Case Study: TransLink

TransLink, the transit agency for Vancouver, Canada, wanted to provide more accurate time estimates for bus departures.

TransLink partnered with Microsoft and T4G to **build 18,000 AI models** that together automatically predicted accurate departure times by considering factors like traffic, bad weather, and other schedule disruptions.

TransLink succeeded in creating and managing this high volume of sophisticated models because they adopted MLOps strategies to:

- Automate model training and deployment processes through pipelines
- Create an approval process for automated model training results
- Integrate a data drift system into build-and-release pipelines so that retraining is triggered automatically if data drift is detected

The solution improved the accuracy of predicted departure times by 74% and reduced average customer wait times by 50%.

SAS

Customer:
Scandinavian Airlines

Industry:
Travel and Transportation

Size:
10,000+ employees

Country:
Sweden

Products and Services:
Microsoft Azure
Azure Data Factory
Azure Databricks
Azure DevOps
Azure Kubernetes Service (AKS)
Azure Machine Learning

[Read full story here]

Microsoft

> "We use Azure Machine Learning to solve real business problems without worrying about building and managing infrastructure or creating new tools—we can focus directly on gaining value from the technology."
>
> —Daniel Engberg, Head of Data Analytics and Artificial Intelligence, Scandinavian Airlines

Situation:

After moving to Microsoft Azure, Scandinavian Airlines (SAS) wanted to use AI and machine learning to address a variety of business challenges, including fresh food optimization.

Solution:

SAS developers were impressed with Azure Machine Learning capabilities, including model interpretability and automated machine learning. So the company narrowed 150 potential use cases down to 5 and started putting them into production.

Impact:

With Azure Machine Learning, SAS has created sophisticated models that cut down on fresh food waste by 45%, accurately forecast sales and full flights, and predict customer willingness to upgrade their flight class, all of which help SAS take better care of its customers.

Microsoft

Q&A with

Xiaopeng Li

AI Business Lead in Western Europe, Microsoft

João Pedro Martins (Jota)

AI Rangers Lead for EMEA & Asia, Microsoft

Microsoft

Call to Actions

Provision your first Azure Machine Learning Workspace

https://docs.microsoft.com/en-us/azure/machine-learning/quickstart-create-resources

Read on the MLOps in Azure Concepts
https://docs.microsoft.com/en-us/azure/machine-learning/concept-model-management-and-deployment

Try an end-to-end MLOps sample in Azure
https://github.com/Microsoft/MLOpsPython

Azure Architecture Center Industry Solutions – MLOps in Manufacturing

https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-technical-paper

Learn about AI/ML in Azure with Microsoft Engineering – The AI Show

https://docs.microsoft.com/en-us/shows/ai-show/

Microsoft

Thank you!