Microsoft

GPS Data/AI Strategy FY23

Delivered by CSA Team

Franck Gaillard
Cloud Solution Architect
Data AI
frgail@microsoft.com

Narjes Majdoub
Cloud Solution Architect
Data AI
nmajdoub@microsoft.com

Ali Bouhaddou
Cloud Solution Architect
Data Analytics
albouhad@microsoft.com

Frederic Gisbert
Cloud Solution Architect
Data Analytics
frgisber@microsoft.com

Azure Data & AI technical intensity plan

- From June 2022 to June 2023
- Focus on "**Azure Data & AI**" tech intensity

- Many content, from **L100** Beginner to **L400** Expert level:
  - **Academy** L100
  - **Webinar** L200/L300
  - **Workshop** L300/L400
  - **Certification kickstart** L300/L400
  - **Openhack / Microhack** L400

Data & AI events timeline – H1

Webinar/Academy- L 200/300
Workshop/ Openhack/ Certifications – L 300/400

Accompagnement certifications - DP-500 / DP-100

- Kickstart (17/10)
- Dry Run (14/11)
- Q&A (05/12)

15 Sept	Azure ML pour les Data Scientists
22 Sep	Azure Synapse
29 Sep	Les solutions SQL dans Azure (PaaS, IaaS, SaaS)
06 Oct	Deploiement et sécurisation des workspaces Azure Ml
10 Oct	Workshop - MCPP DP-420
13 Oct	Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share
17 Oct	Academy DP900/AI900
18 Oct	Synapse et Lakehouse Database
20 Oct	MLOps avec Azure ML & Azure CosmosDB
08 Nov	Le NoSQL dans Azure - Migration CosmosDB / OSS
10 Nov	SQL Server 2022 et hybridation native avec Azure SQL Managed Instance
17 Nov	Machine Learning dans Azure Synapse
22 Nov	Lab Nantes - Data & AI
24 Nov	Cosmos DB et IA
29 Nov	Lab Lyon - Data & AI
8 Déc	Azure Cognitive Services - L'IA sans code
12 Déc	L'IA dans les services de données Azure
15 Déc	Gouvernance de données dans Azure avec Microsoft Purview

Data & AI events timeline – H2

Webinar/Academy- L 200/300
Workshop/ Openhack/ Certifications – L 300/400

Accompagnement certifications - **DP-203**

Kickstart (xx)
Dry Run (xx)
Q&A (xx)

12 Jan
19 Jan
26 Jan
09 Feb
16 Feb
09 Mar
16 Mar
23 Mar
06 Apr
13 Apr
04 Mai
16 Mai
24 Mai
01 Juin

MLOps avec Azure Machine Learning

Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.

Déploiement et sécurisartion des workspace Azure Synapse

Azure ML pour les Citizen Data Scientists

Webinar - Self Service Analytics

Responsible AI

Machine Learning dans Azure Synapse

Academy DP900/AI900

Les bases de données Open Source dans le cloud Azure

Le edge/Hybrid Cloud - ML Processing - ML@edge, Percept, SQL@edge

Gouvernance de données dans Azure avec Microsoft Purview

Workshop - Open AI

Les solutions SQL dans Azure (PaaS, IaaS, SaaS)

Data processing dans Azure avec Azure Synapse, Azure Batch, Spark, Notebook, etc.

Le edge/Hybrid Cloud - Data processing - SQL@edge, CEP@edge, Hyperscale, Arc, AKS

Webinar - Self Service Analytics

Liste des évènements de type Webinar 2H

Event Webinar (Les jeudis de la Data & AI) - L200/300	Date	Duration (min)	Link
Azure Machine Learning pour les Data Scientists	15/09/2022	120	https://msevents.microsoft.com/event?id=2454281594
Azure Synapse	22/09/2022	120	https://msevents.microsoft.com/event?id=857781749
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	29/09/2022	120	https://msevents.microsoft.com/event?id=502366997
Déploiement et sécurisation des workspaces Azure Machine learning	06/10/2022	120	https://msevents.microsoft.com/event?id=1505714138
Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share	13/10/2022	120	https://msevents.microsoft.com/event?id=139685175
MLOps avec Azure Machine Learning	20/10/2022	120	https://msevents.microsoft.com/event?id=1245885767
SQL Server 2022 et hybridation native avec Azure SQL Managed Instance	10/11/2022	120	https://msevents.microsoft.com/event?id=145826476
Machine Learning dans Azure Synapse Analytics	17/11/2022	120	https://msevents.microsoft.com/event?id=3637723312
Azure Cosmos DB et IA	24/11/2022	120	https://msevents.microsoft.com/event?id=2646013445
Azure et les Services Cognitifs	08/12/2022	120	https://msevents.microsoft.com/event?id=3772037220
La gouvernance de données dans Azure avec Microsoft Purview	15/12/2022	120	https://msevents.microsoft.com/event?id=1499560981
MLOps avec Azure Machine Learning	12/01/2023	120	https://msevents.microsoft.com/event?id=4115194515
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	19/01/2023	120	https://msevents.microsoft.com/event?id=1537241181
Déploiement et sécurisation des workspace Azure Synapse	26/01/2023	120	https://msevents.microsoft.com/event?id=1806467748
Azure Machine Learning pour les Citizen Data Scientists	09/02/2023	120	En cours
PowerBI - Self Service Analytics	16/02/2023	120	https://msevents.microsoft.com/event?id=1401519679
L'IA responsable avec Azure machine learning	09/03/2023	120	https://msevents.microsoft.com/event?id=2072953112
Machine Learning dans Azure Synapse Analytics	16/03/2023	120	https://msevents.microsoft.com/event?id=3413014857
Les bases de données Open Source dans le cloud Azure	23/03/2023	120	https://msevents.microsoft.com/event?id=2727487131
Hybridation des services de Machine Learning Azure	06/04/2023	120	https://msevents.microsoft.com/event?id=1624914222
La gouvernance de données dans Azure avec Microsoft Purview	13/04/2023	120	https://msevents.microsoft.com/event?id=3909342839
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	04/05/2023	120	https://msevents.microsoft.com/event?id=1162207895
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	16/05/2023	120	https://msevents.microsoft.com/event?id=3517068442
Hybridation des services de données Azure	24/05/2023	120	https://msevents.microsoft.com/event?id=2996507398
Self Service Analytics	01/06/2023	120	En cours

Liste des évènements de type Workshop/Prepa Cert/Academy

Event Workshop L300/400	Date	Duration (min)	Link
Synapse et Lakehouse Database	18/10/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Le NoSQL dans Azure - Migration CosmosDB / OSS	08/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Lab Lyon - Data & AI	22/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMlZZOURET0RSWjcyTERYRkJGTlFFUJaUi4u
Lab Nantes - Data & AI	29/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMlZZOURET0RSWjcyTERYRkJGTlFFUJaUi4u
L'IA dans les services de données Azure	12/12/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Open AI	H2	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u

Event Academy, kickstart certifications, workshop certifications	Date	Duration (min)	Link
MCPP - DP-420	10/10/2022	420	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUMkJISlRKSU1RRFA0OVgzSFdTSTY0RE9WQy4u
Micro Hack CosmosDB	20/10/2022	420	H1 - Inscriptions PTA
Academy DP900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=3250818161
Academy AI900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=2717528090
Kickstart DP-500	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Dry Run DP-500	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Q&A DP-500	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Kickstart DP-100	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Dry Run DP-100	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Q&A DP-100	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMUFYxSkRIMi4u
Kickstart DP-203	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Dry Run DP-203	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Q&A DP-203	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHbR3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u

Microsoft Azure

Exam DP-100
Designing and Implementing an Azure Data Science Solution

Kick Start session

16/10/2022

Microsoft Azure

About the DP-100 certification

The journey to Microsoft Certified:

Azure Data Scientist Associate

Get started at
aka.ms/AzureCerts_DataScientist

Start here
Decide if this is the right certification for you

This certification is a good fit if your responsibilities include:

- Planning and creating a suitable working environment for data science workloads on Azure.
- Running data experiments and training predictive models.
- Managing, optimizing, and deploying machine learning models into production.

Brand new? First, master the basics.

New to AI or AI on Azure?
Choose Azure AI fundamentals training.

Get trained

Exam study guide

- DP-100

Self-paced training

- Microsoft Learn

Instructor-led training

- Course DP-100: Designing and Implementing a Data
- Course DP-090: Implementing a Machine Learning Solution with Microsoft Azure Databricks

Build confidence

Exam preparation

- Exam Readiness Zone

Take a practice exam

- Microsoft Official Practice Test DP-100

Skills measured:

- Manage Azure resources for machine learning
- Run experiments and train models
- Deploy and operationalize machine learning solutions
- Implement responsible machine learning

Get recognized

Pass Exam DP-100 to earn this certification.

Microsoft Certified: **Azure Data Scientist Associate**

Azure data scientists apply data science and machine learning to implement and run machine learning workloads on Azure.

Apply skills

- Microsoft Learn Cloud Games: Data Feeds

Products featured

- Microsoft Azure

Microsoft Azure

Watch the Azure certifications overview video

Learning path for Azure Data Scientist

Azure Data Scientists apply Azure's machine learning techniques to train, evaluate, and deploy models that solve business problems.

aka.ms/AzureDataScienceLearn

Classroom

Designing and Implementing a Data Science Solution on Azure
DP-100T01

Exam DP-100:
Designing and Implementing a Data Science Solution on Azure

Digital skilling: free interactive training content on Microsoft Learn

Azure fundamentals

9H 59M - 12 Modules

1. Principles of cloud computing
2. Introduction to Azure
3. Azure architecture and service guarantees
4. Create an Azure account
5. Manage services with the Azure portal
6. Azure compute options
7. Azure data storage options
8. Azure networking options
9. Security, responsibility and trust in Azure
10. Apply and monitor infrastructure standards with Azure Policy
11. Control and organize Azure resources with Azure Resource Manager
12. Predict costs and optimize spending for Azure

Explore AI solution development with data science services in Azure

1H 40M - 2 Modules

1. Introduction to Data Science in Azure
2. Choose the Data Science service in Azure you need

Get started with Machine Learning with an Azure Data Science Virtual Machine

1H 43M - 3 Modules

1. Introduction to the Azure Data Science Virtual Machine
2. Explore the types of Azure Data Science Virtual Machines
3. Provision and use an Azure Data Science Virtual Machine

Build AI solutions with Azure Machine Learning service

3H 17M - 4 Modules

1. Introduction to Azure Machine Learning service
2. Train a local ML model with Azure Machine Learning service
3. Automate the ML model selection with Azure Machine Learning service
4. Register and deploy ML models with Azure Machine Learning service

Extract knowledge and insights from your data with Azure Databricks

4H 21M - 6 Modules

1. Introduction to Azure Databricks
2. Read and write data by using Azure Databricks
3. Perform exploratory data analysis with Azure Databricks
4. Train, evaluate, and select machine-learning models with Azure Databricks
5. Deep learning with Azure Databricks
6. Perform text analytics with Azure Databricks

Become a Microsoft Certified Professional

Certification helps establish your credentials with current and future employers.

Pass certification exam DP-100 to become a Microsoft Certified: Azure Data Scientist Associate.

Free digital skilling:
Microsoft.com/Learn

Assess your skills at Pluralsight:
Azure.com/Pluralsight

Find a Learning Partner:
aka.ms/LearningPartner

Microsoft Certification:
Microsoft.com/Certification

Microsoft

Publication date: October 2019

About This Course

Learn how to use Azure Machine Learning to operate machine learning workloads in the cloud

- Build on your existing data science and machine learning knowledge
- Leverage cloud services to perform machine learning at scale
- Explore considerations for responsible machine learning

Skills measured

- Manage Azure resources for machine learning (25–30%)
- Run experiments and train models (20–25%)
- Deploy and operationalize machine learning solutions (35–40%)
- Implement responsible machine learning (5–10%)

Exam objectives

Skills Measured	Weights
Manage Azure resources for machine learning	25-30%
Run experiments and train models	20-25%
Deploy and operationalize machine learning solutions	35-40%
Implement responsible machine learning	5-10%

- Percentages indicate the relative weight of each area on the exam
- The higher the percentage, the more questions you are likely to see in that area

Passing score is 700

Course Agenda

Module 1: Getting Started with Azure Machine Learning
Module 2: No-Code Machine Learning
Module 3: Running Experiments and Training Models
Module 4: Working with Data
Module 5: Working with Compute
Module 6: Orchestrating Machine Learning Workflows
Module 7: Deploying and Consuming Models
Module 8: Training Optimal Models
Module 9: Responsible Machine Learning
Module 10: Monitoring Models

Lab Environment

The course emphasizes hands-on learning

You will need:

- A modern web browser (for example, Microsoft Edge)
- The lab instructions for this course: https://aka.ms/mslearn-dp100
- A Microsoft Azure subscription
    - Redeem your Azure Pass code at https://www.microsoftazurepass.com
    - Sign in with a Microsoft account that hasn't been used to redeem an Azure Pass previously

Microsoft Azure

Module 1: Getting Started with Azure Machine Learning

Agenda

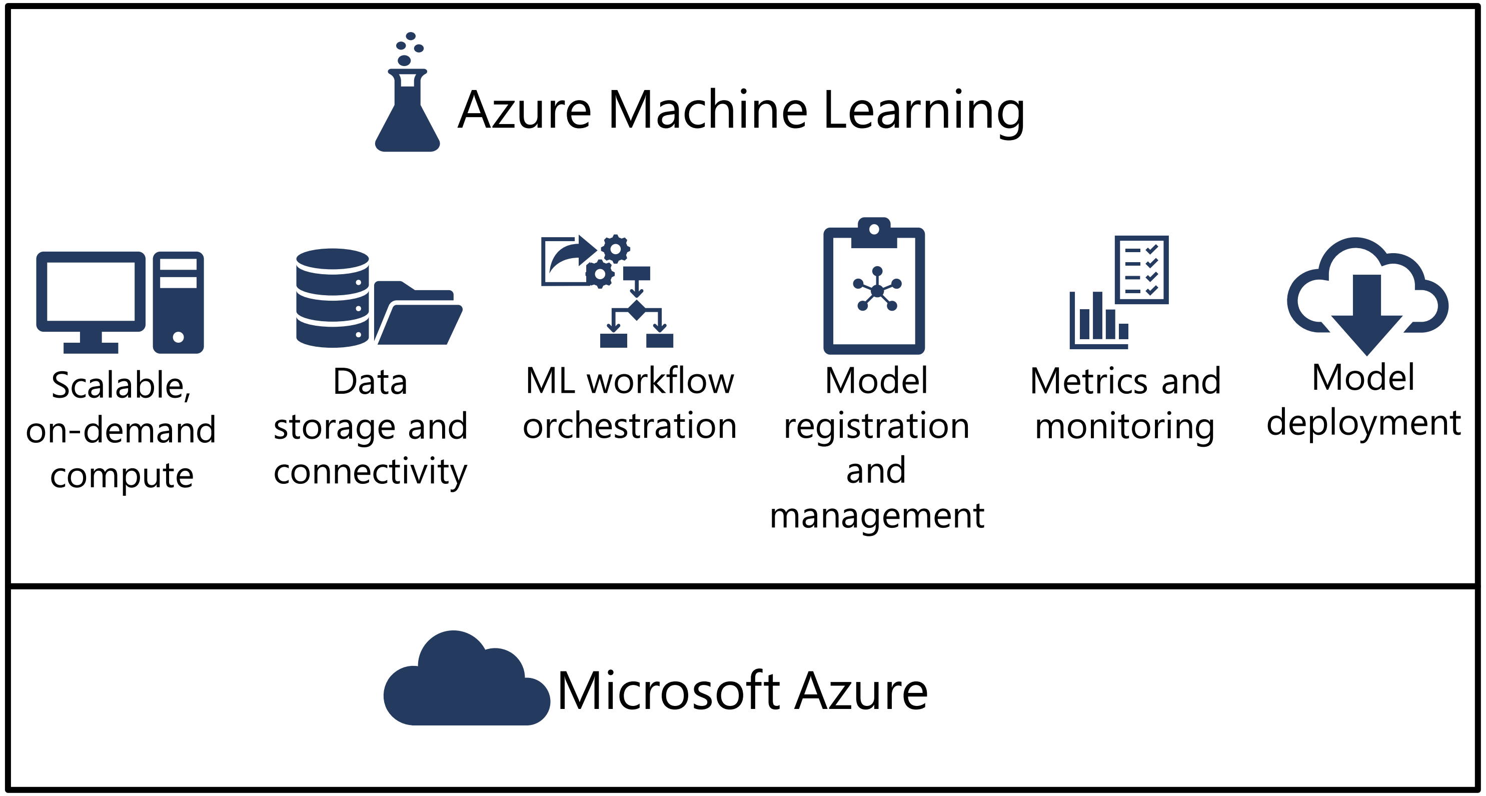
Introduction to Azure Machine Learning

Working with Azure Machine Learning

Introduction to Azure Machine Learning

What is Azure Machine Learning?

A platform for operating machine learning workloads in the cloud



Azure Machine Learning in Context

Data

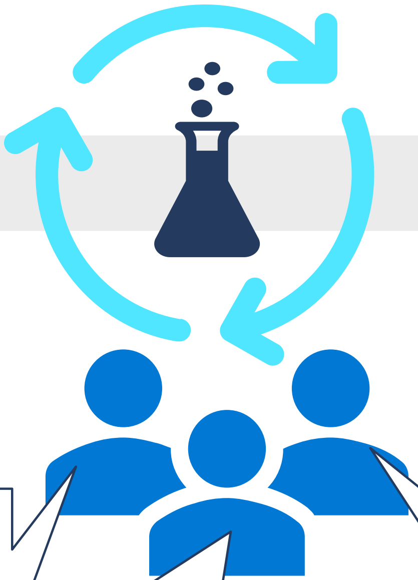
Storage

- Azure Storage
- Azure Data Lake Store
- Azure SQL Database
- Azure Cosmos DB

Transfer and Transform

- Azure Data Factory
- Azure HDInsight
- Azure Databricks

Azure Machine Learning



Data Scientist

- Run data analysis experiments using Python or R
- Train models using frameworks like Scikit-Learn, PyTorch, and TensorFlow

"Citizen" Data Scientist / App Developer

Create models using graphical tools

Software Engineer / Operator

- Deploy models as services
- Orchestrate ML operations
- Integrate ML into DevOps processes

ML-Powered Apps

Application Services

- Azure App Service
- Azure Functions
- Azure IoT Edge

Container Deployment

- Azure Container Instances
- Azure Kubernetes Service

Machine Learning Operationalization (ML Ops)

Webinar MLOps 20/10/22 – 10:00-12:00 : https://lnkd.in/eEEiQ7XU

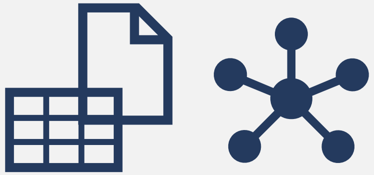
Based on *DevOps* principles, including:

- Infrastructure-as-code and configuration management
- Version control and tracking
- Continuous integration and delivery (CI/CD)
- Continuous monitoring

Experiment history, including metrics, outputs, and metadata

On-demand compute with reusable environments

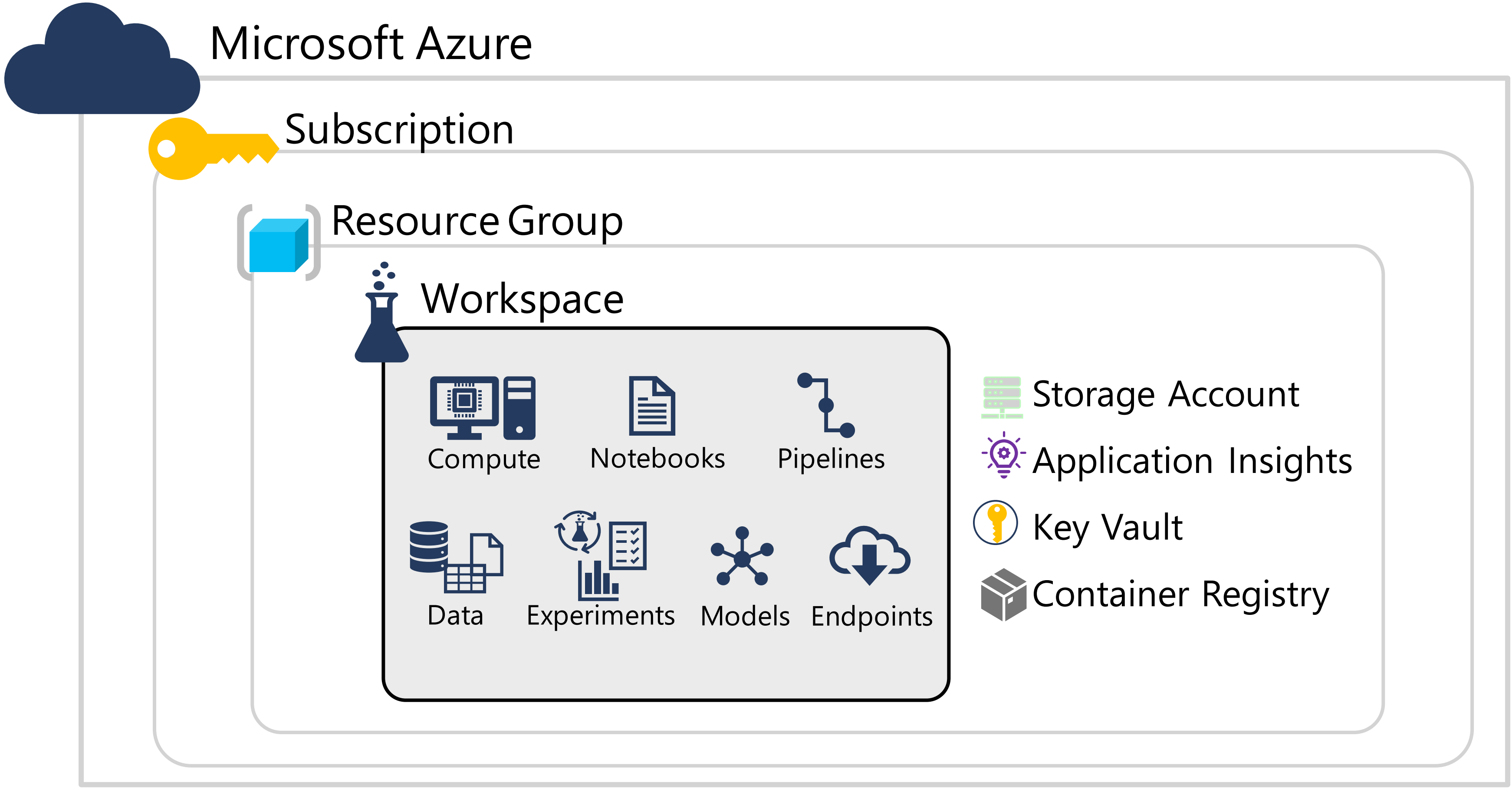
Dataset and model versioning



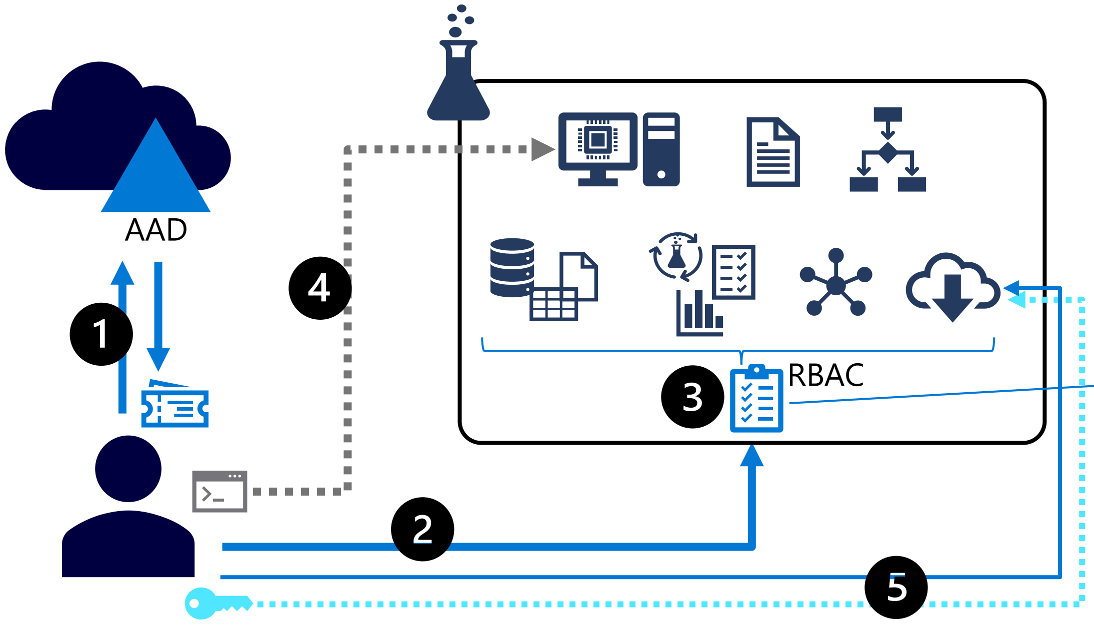
Pipelines, event-driven automation, and CI/CD

Monitoring of deployed models and data drift

Azure Machine Learning Workspaces



Access Control and Permissions



Default RBAC permissions

Permission	Owner	Contributor	Reader
Create workspace	√	√	
Share workspace	√		
Create compute target	√	√	
Attach compute target	√	√	
Attach data stores	√	√	
Run experiment	√	√	
View runs/metrics	√	√	√
Register model	√	√	
Create image	√	√	
Deploy web service	√	√	
View models/images	√	√	√
Call web service	√	√	√

1. User signs into Azure Active Directory (AAD) and obtains token
2. Token grants access to Azure Machine Learning workspace
3. Role-based access control (RBAC) permissions control resource access
4. Compute resources can optionally allow access via SSH
5. Deployed service endpoints can use key or token-based access

Working with Azure Machine Learning

Azure Machine Learning studio

Manage compute and data

Run experiments

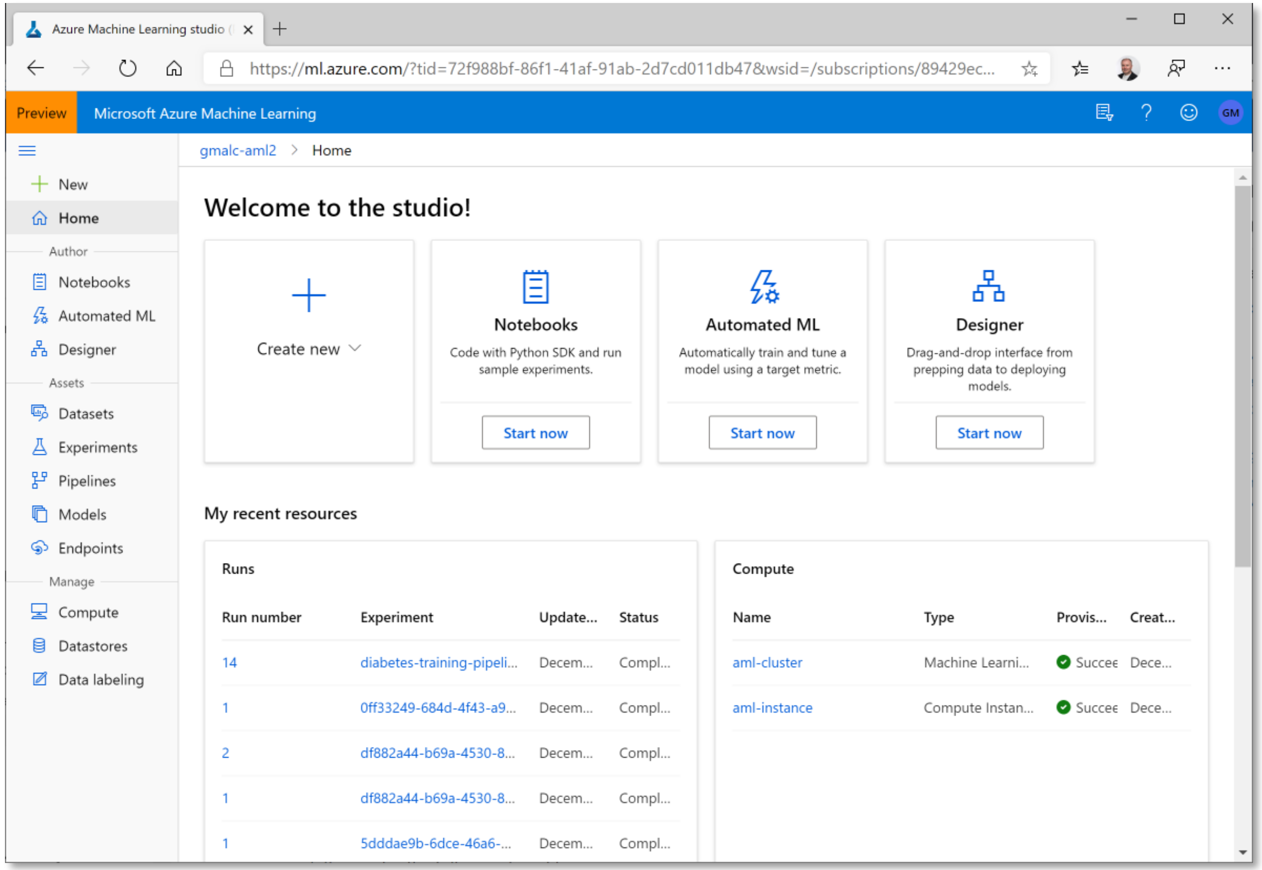
View metrics and logs

Manage and deploy models

Manage service endpoints

Label image data

Use graphical modeling tools:



- *Automated ML* - find the best model for your data
- *Designer* – drag and drop model development

The Azure Machine Learning SDK for Python

Python programming interface for Azure Machine Learning

```
pip install azureml-sdk
```

```
from azureml.core import Workspace

ws = Workspace.from_config()
for compute_name in ws.compute_targets:
    compute = ws.compute_targets[compute_name]
    print(compute.name, ":", compute.type)
```

Azure Machine Learning CLI Extension

Cross-platform command-line interface for Azure Machine Learning

```
az extension add -n azure-cli-ml
```

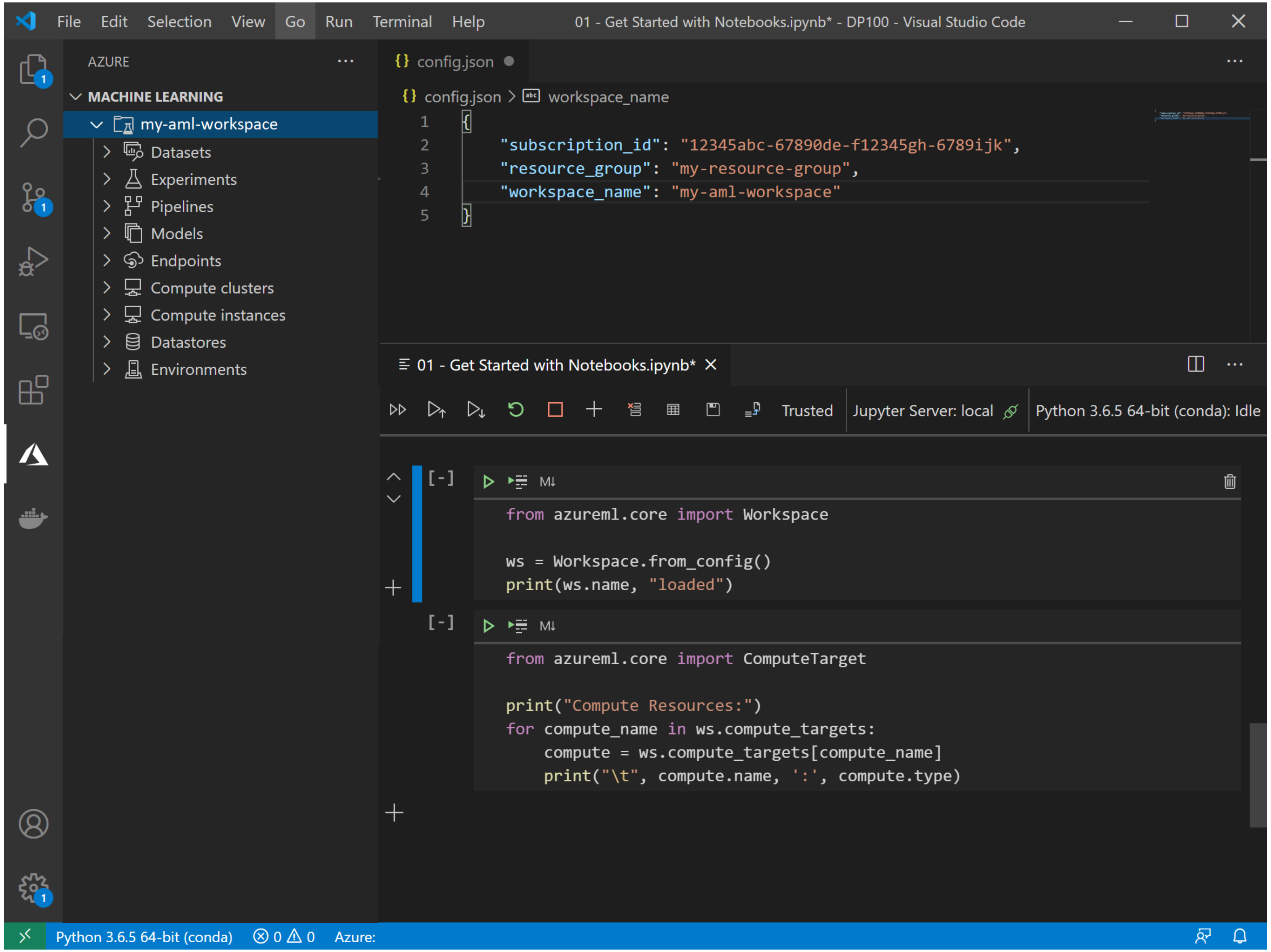
```
az ml computetarget list -g 'my-resource-group' -w 'my-aml-workspace'
```

Visual Studio Code

Cross-platform code editor and integrated development environment

Tools for machine learning provided through *extensions*:

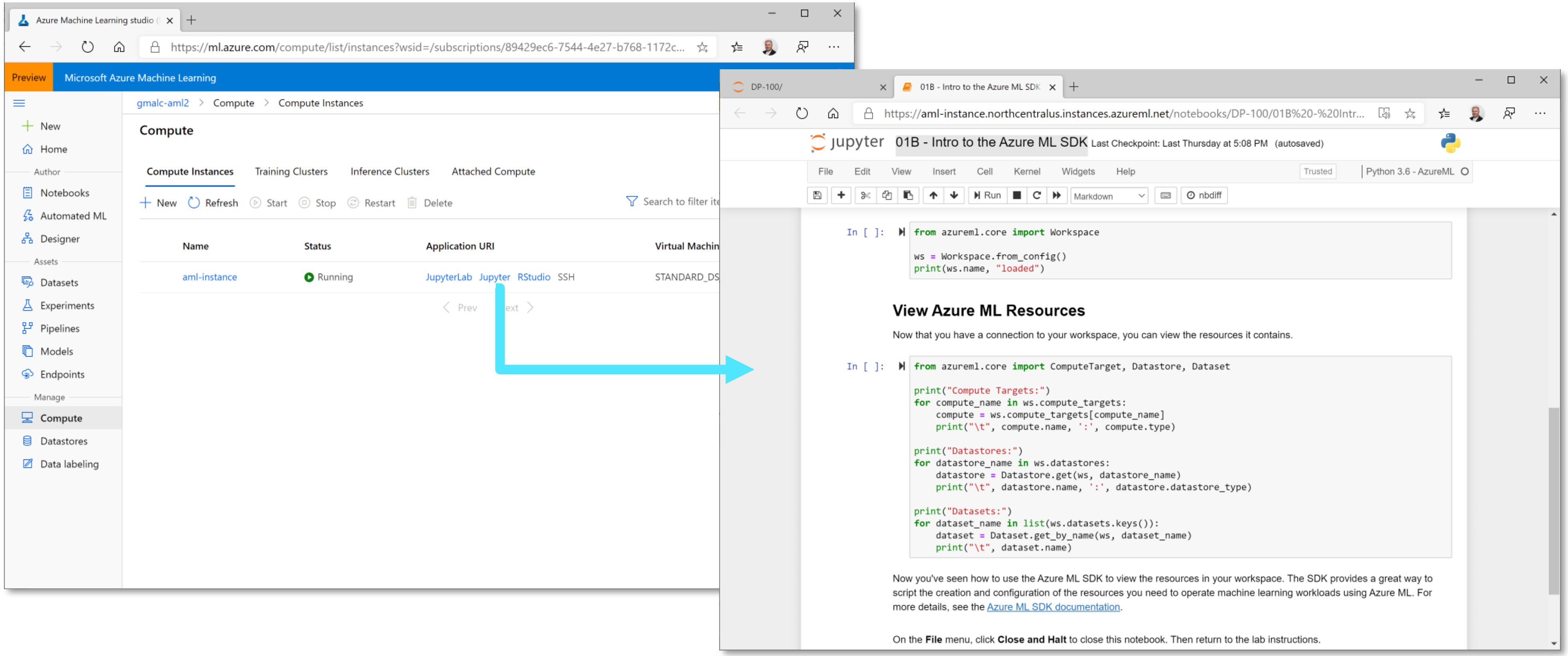
- **Python**: Native Python coding and debugging, and integrated notebook interface
- **Azure Machine Learning**: a graphical interface for working with an Azure Machine Learning workspace



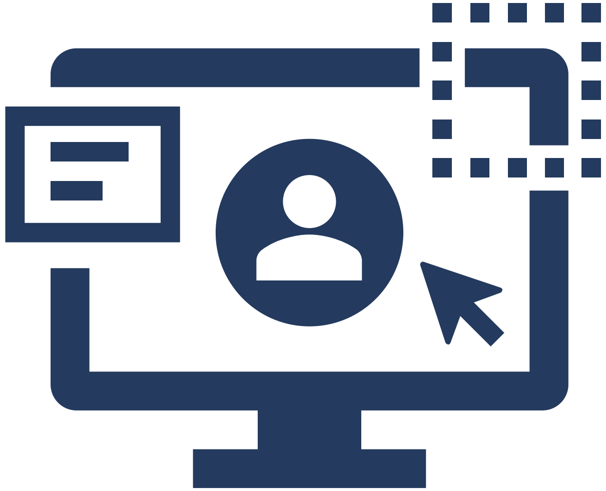
Azure Machine Learning Compute Instances

A cloud-based development workstation right in your workspace

Built-in Jupyter, JupyterLab, and RStudio



Lab: Create an Azure Machine Learning Workspace



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create an Azure Machine Learning workspace** exercise

Knowledge check

Which of the following Azure resources are created alongside an Azure Machine Learning workspace?

- [x] Storage Account
- [ ] Databricks workspace
- [x] Key Vault
- [x] Application Insights

Which of the following provides a web interface for managing assets in a workspace?

- [x] Azure Machine Learning studio
- [ ] Azure Cognitive Services
- [ ] Azure Synapse Analytics

Which Visual Studio Code extension enables integrated management of workspace assets?

- [ ] Python
- [x] Azure Machine Learning
- [ ] Jupyter Notebooks

References

Microsoft Learn: Introduction to Azure Machine Learning
https://docs.microsoft.com/learn/modules/intro-to-azure-machine-learning-service

Azure Machine Learning architecture and concepts documentation
https://docs.microsoft.com/azure/machine-learning/concept-azure-machine-learning-architecture

Azure Machine Learning studio documentation
https://docs.microsoft.com/azure/machine-learning/overview-what-is-machine-learning-studio

Azure Machine Learning enterprise security documentation
https://docs.microsoft.com/azure/machine-learning/concept-enterprise-security

Azure Machine Learning Python SDK documentation
https://docs.microsoft.com/python/api/overview/azure/ml/intro

Azure Machine Learning extension for Visual Studio Code documentation
https://docs.microsoft.com/azure/machine-learning/tutorial-setup-vscode-extension

Microsoft Azure

Module 2: No-Code Machine Learning

Agenda

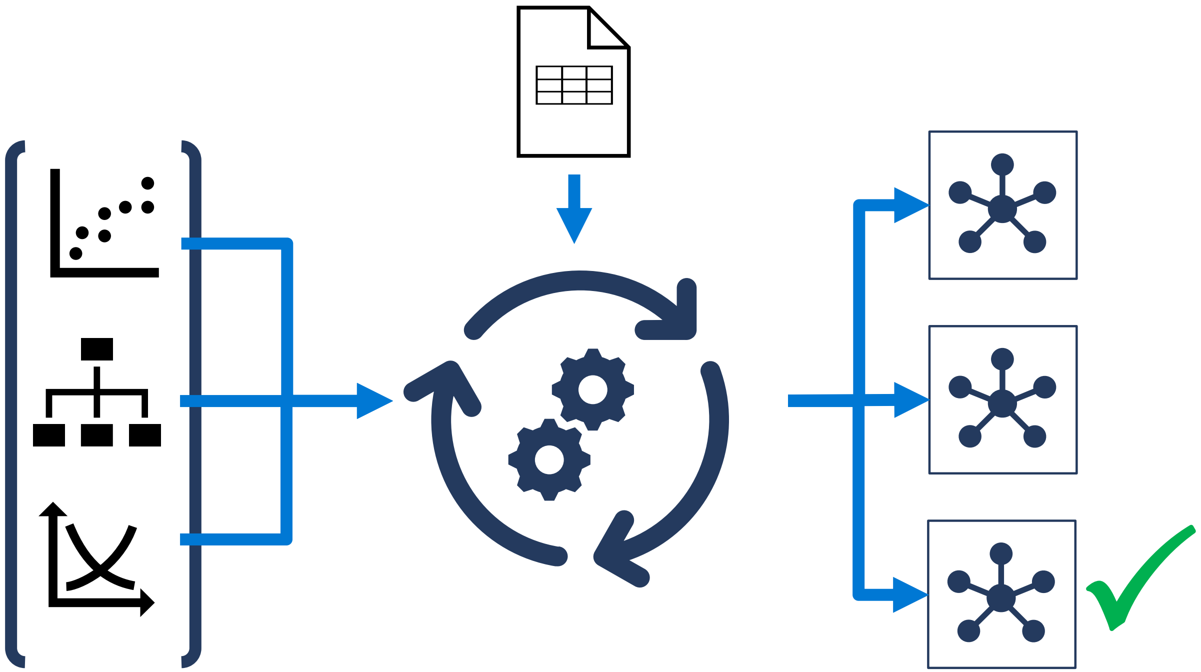
- Automated Machine Learning
- Azure Machine Learning Designer

Automated Machine Learning

What is Automated Machine Learning?

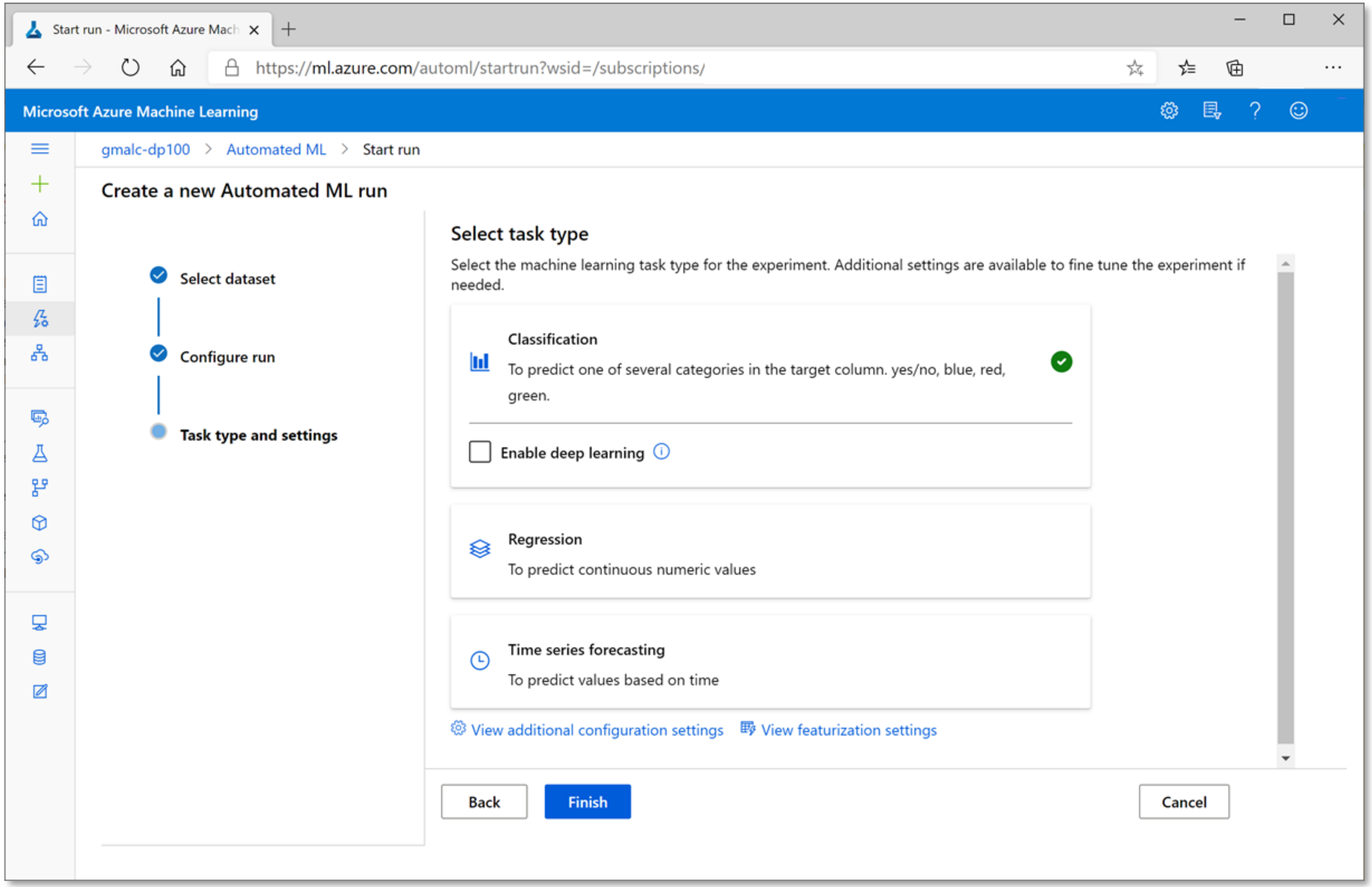
Train multiple models in parallel, varying algorithm and preprocessing

Find the "best" model based on a specific performance metric



Automated ML in Azure Machine Learning Studio

1. **Select dataset**
   - Upload files
   - Import from Web
   - Register data source
2. **Configure run**
   - Experiment name
   - Target label
   - Compute
3. **Task type and settings**
   - Classification
   - Regression
   - Time Series



Configuration and Featurization

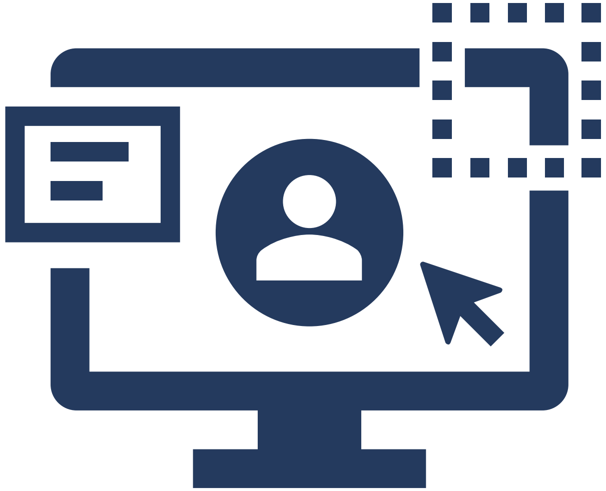
Configuration Options

- Primary metric (used to evaluate the best model)
- Explain best model (generates feature importance)
- Blocked algorithms (restricts training algorithms)
- Exit criterion (enables early-stopping)
- Validation (sets cross-validation technique)
- Concurrency (sets number of parallel iterations)

Featurization

- Normalization / scaling is automatic
- Optional featurization includes:
  - Dropping high-cardinality features
  - Imputing missing values
  - Categorical encoding
  - Derived feature generation
- Data guardrails mitigate unbalanced data

Lab: Use Automated Machine Learning

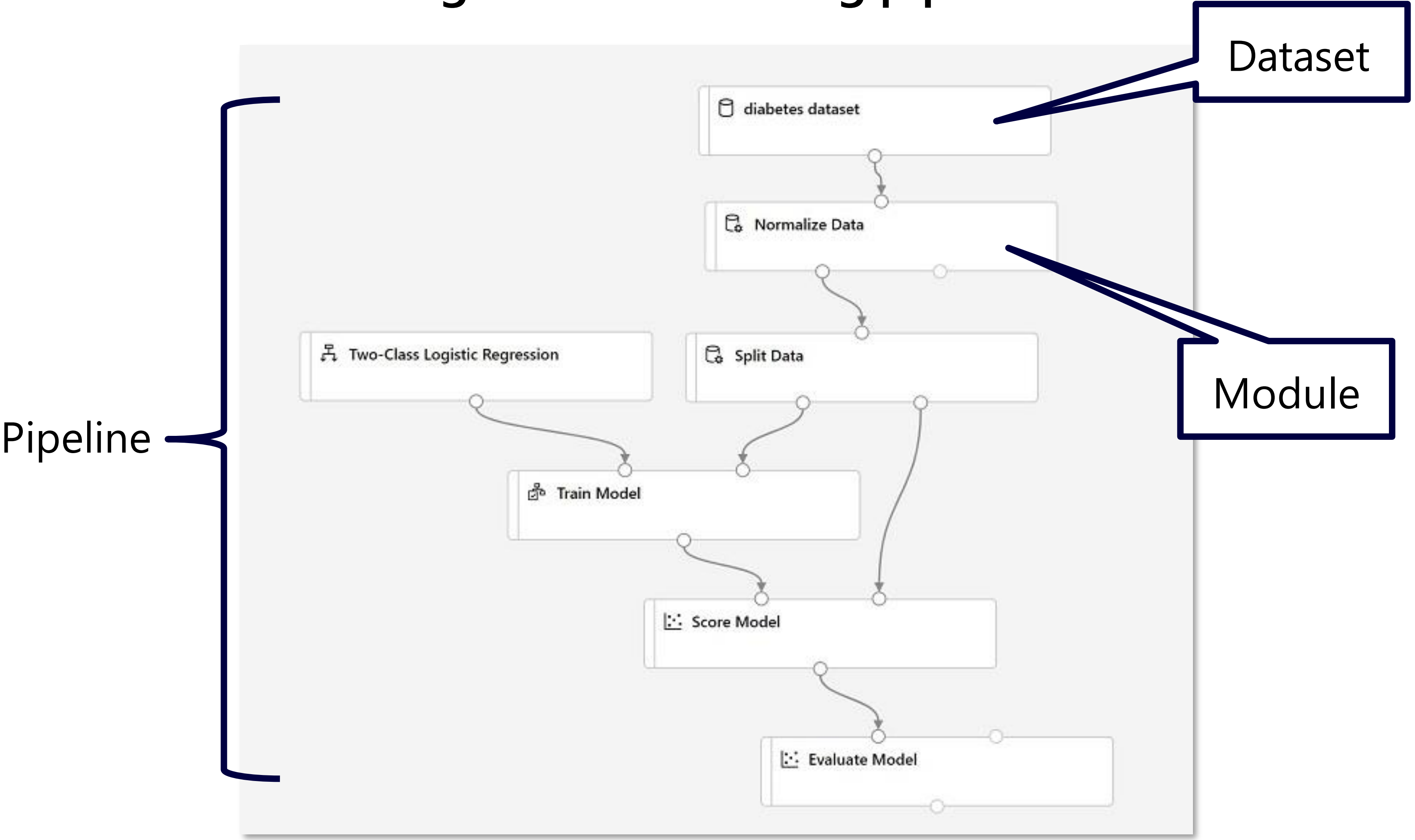


1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Use Automated Machine Learning** exercise

Azure Machine Learning Designer

What is Azure Machine Learning Designer?

A visual interface for creating machine learning pipelines

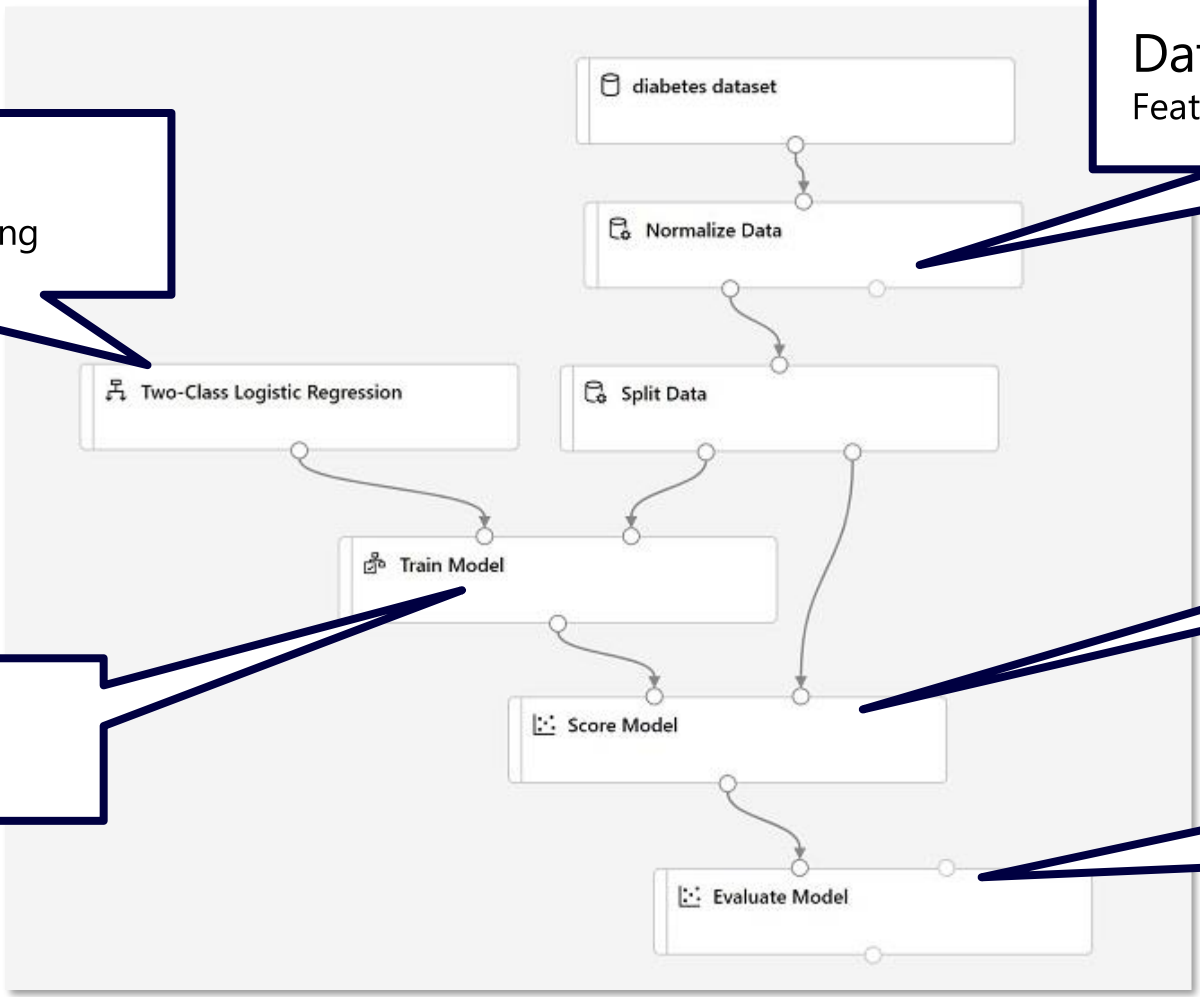


Training Pipelines

Data preparation, model training, scoring, and evaluation

Algorithm
Classification, Regression, Clustering

Data preparation
Feature selection / engineering



Score
Predict from test data

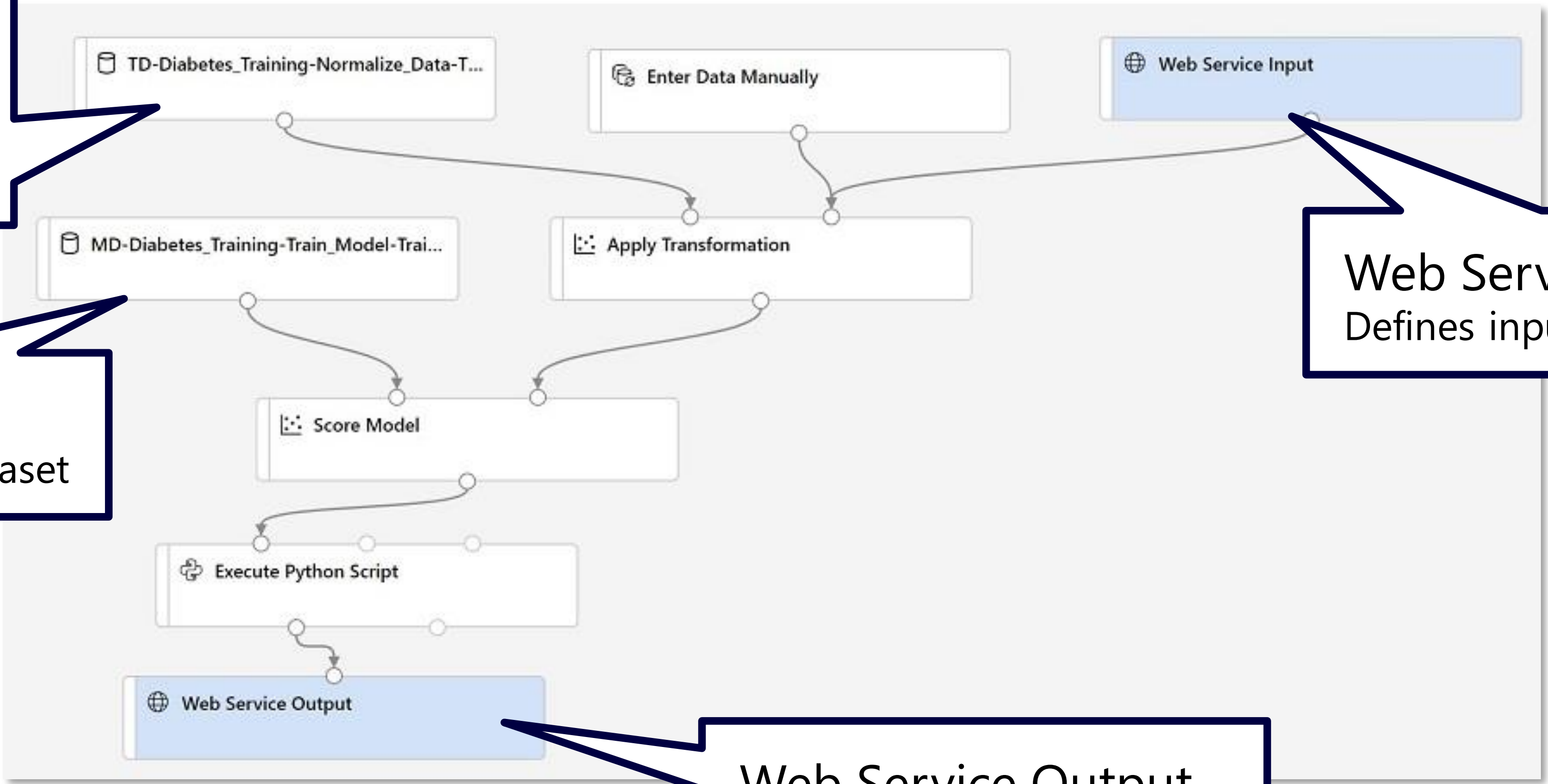
Train
Fit model for specified label

Evaluate
Model-specific metrics

Inference Pipelines

Use the trained model to get predictions from new data

Transformations
Encapsulated in datasets based on training data statistics

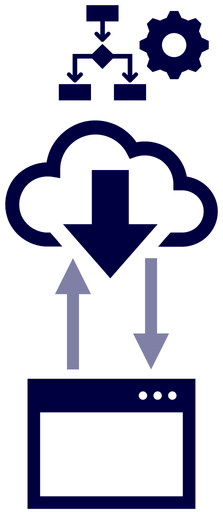


Web Service Input
Defines input schema

Trained Model
Encapsulated in a Dataset

Web Service Output
Defines output schema

Publishing a Service Endpoint

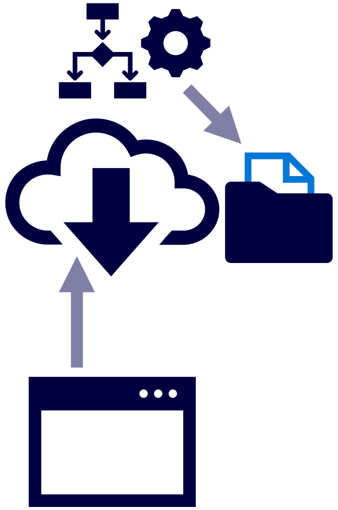


Deploy a Real-Time Pipeline:

Specify deployment target:

- Azure Container Instance
- Azure Kubernetes Services Inference Compute

Submit new data to an HTTP endpoint for immediate results



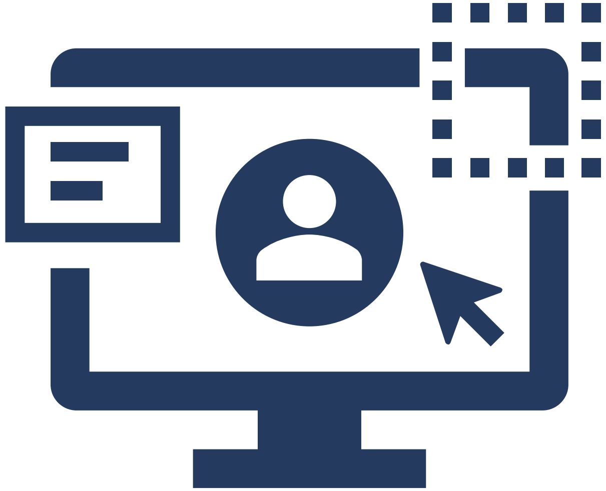
Publish a Batch Pipeline

Runs on Azure Machine Learning Training Compute

Initiate a pipeline experiment run through an HTTP endpoint

Results are saved in the run output

Lab: Use Azure Machine Learning Designer



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Use Azure Machine Learning Designer** exercise

Knowledge check

You want to use automated machine learning with car sales data to train a machine learning model that predicts the price of a car based on its make, model, engine size, and mileage.

What task type should you select?

- [ ] Classification
- [x] Regression
- [ ] Time-series

You are creating a training pipeline using a dataset that has multiple numeric columns. You want to transform the numeric columns so that the values are all on a similar scale.

Which module should you add to the pipeline?

- [ ] Select Columns in a Dataset
- [ ] Clean Missing Data
- [x] Normalize Data

References

Microsoft Learn: Create no-code predictive models with Azure Machine Learning
https://docs.microsoft.com/learn/paths/create-no-code-predictive-models-azure-machine-learning

Automated Machine Learning documentation
https://docs.microsoft.com/azure/machine-learning/concept-automated-ml

Designer documentation
https://docs.microsoft.com/azure/machine-learning/concept-designer

Microsoft Azure

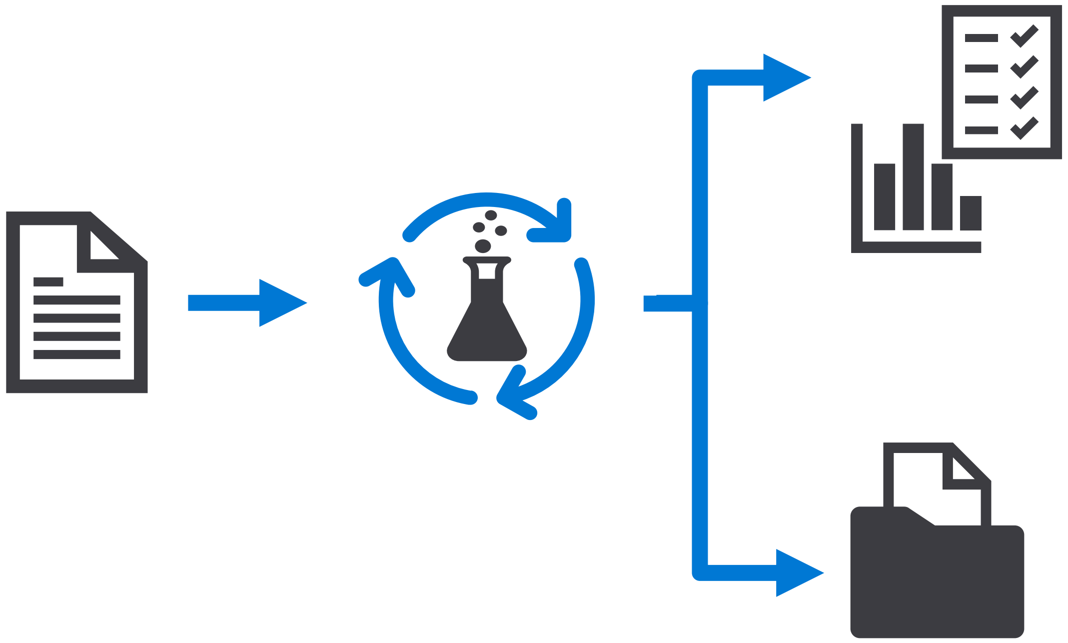
Module 3: Running Experiments and Training Models

Agenda

- Introduction to Experiments
- Training and Registering Models

Introduction to Experiments

What is an Experiment?



An executable process that is run one or more times – often a script

Each run generates metrics and output files

Metadata and events are recorded in log files

Running an Experiment Inline

Create (or get existing) experiment in workspace

Start experiment run

Log metrics in run

Save or upload output files

Complete the run

```
from azureml.core import Workspace, Experiment
import pandas as pd

ws = Workspace.from_config()
experiment = Experiment(workspace=ws, name='my-experiment')
run = experiment.start_logging()
data = pd.read_csv('data.csv')
row_count = (len(data))
run.log('observations', row_count)
data.sample(100).to_csv('sample.csv', index=False, header=True)
run.upload_file(name='outputs/sample.csv', path_or_stream='./sample.csv')
run.complete()
```

Running a Script as an Experiment

Script:

```
from azureml.core import Run

run = Run.get_context()
run.log(…)
run.complete()
```

Get the experiment run context for the current script

Control code (to initiate and monitor experiment run):

```
from azureml.core import Workspace, Experiment, ScriptRunConfig

ws = Workspace.from_config()

script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py')

experiment = Experiment(workspace=ws, name='my-script-experiment')
run = experiment.submit(config=script_config)
```

Define run settings for the experiment script

(can include compute target, conda environment, and more)

Using MLflow

Using MLflow Inline

```
from azureml.core import Experiment
import mlflow

mlflow.set_tracking_uri(ws.get_mlflow_tracking_uri())
experiment = Experiment(workspace=ws, name='mlflow-experiment')
mlflow.set_experiment(experiment.name)
with mlflow.start_run():
    mlflow.log_metric('my_metric', 123)
```

Configure MLflow to log to the Azure Machine Learning workspace

Create an experiment in the workspace

Create and start an MLflow run of the Azure ML experiment

Using MLflow with Scripts

Script:

```
import mlflow

with mlflow.start_run():
    mlflow.log_metric('my_metric', 123)
```

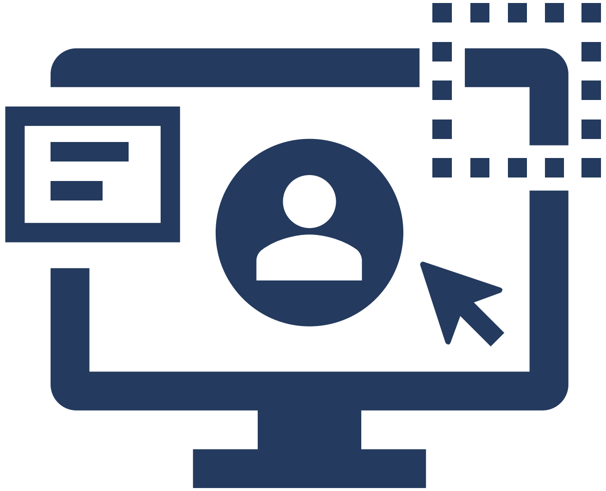
Control code:

```
sc = ScriptRunConfig(source_directory='my_dir',
                     script='script.py'
                     environment=env)

ex = Experiment(workspace=ws, name='mlf-exp')
run = ex.submit(config=sc)
```

An **Environment** that includes the mlflow package

Tracking URI is set to workspace automatically

Lab: Run Experiments



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Run experiments** exercise

Training and Registering Models

Training a Model in a Script

Script:

```
from azureml.core import Run
import joblib
from sklearn.linear_model import LogisticRegression
...
joblib.dump(value=model, filename='outputs/model.pkl')
```

Save trained model in **outputs** folder to record is in experiment run

Control code:

```
from azureml.core import Workspace, Experiment, ScriptRunConfig,
                         Environment, CondaDependencies

env = Environment('training_env')
deps = CondaDependencies.create(pip_packages=['scikit-learn','azureml-defaults'])
env.python.conda_dependencies = deps
script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py',
                                environment=env)
experiment = Experiment(workspace=ws, name='my-script-experiment')
run = experiment.submit(config=script_config)
```

Run script in an environment that includes required ML framework

Scikit-Learn, PyTorch, TensorFlow, ...

Using Script Arguments

Script

```
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('--reg_rate', type=float, dest='reg_rate', default=0.01)
args = parser.parse_args()

model = LogisticRegression(C=1/args.reg_rate).fit(X_train, y_train)
```

Parse script arguments

Use argument values in script

Control code:

```
script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py',
                                arguments = ['--reg_rate', 0.1],
                                environment=env)
```

Specify named arguments in ScriptRunConfig

Registering a Model

Register from training run:

```
run.register_model(model_name='classification_model',
                   model_path='outputs/model.pkl',
                   description='A classification model')
```

Model saved in run **outputs**

Register from local file(s)

```
from azureml.core import Model

model = Model.register(model_name='classification_model',
                       model_path='local_dir/model.pkl',
                       description='A classification model'
                       workspace = ws)
```

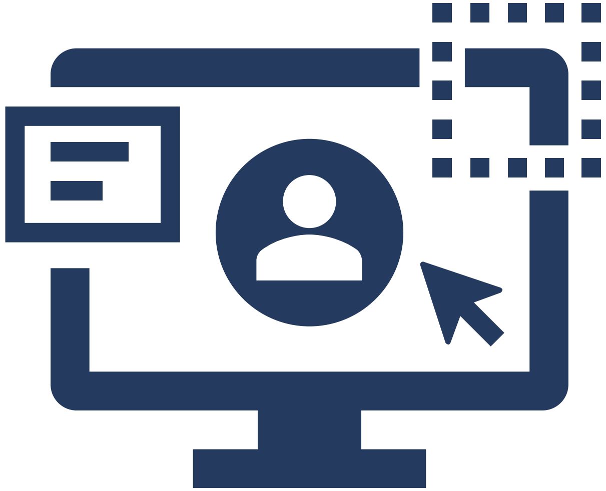
Local model (can be file or folder)

Retrieve registered models

```
for model in Model.list(ws):
    print(model.name, 'version:', model.version)
```

Models are automatically versioned based on name

Lab: Train Models



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Train models** exercise

Knowledge check

You are using the Azure Machine Learning Python SDK to write code for an experiment. You must log metrics from each run of the experiment and be able to retrieve them easily from each run.

What should you do?

- ❑ Add **print** statements to the experiment code to print the metrics.
- ☑ Use the **log*** methods of the **Run** class to record named metrics
- ❑ Save the experiment data in the **outputs** folder

You want to use a script-based experiment to train a PyTorch model, setting the batch size and learning rate hyperparameters to different values each time the experiment runs.

What should you do?

- ❑ Create multiple script files – one for each batch size and learning rate combination you want to use.
- ❑ Set the **batch_size** and **learning_rate** properties of the **ScriptRunConfig** before running the experiment.
- ☑ Add arguments for batch size and learning rate to the script, and set them in the **arguments** property of the **ScriptRunConfig**

References

Microsoft Learn: Introduction to Azure Machine Learning
https://docs.microsoft.com/learn/modules/intro-to-azure-machine-learning-service

Microsoft Learn: Train a machine learning model with Azure Machine Learning
https://docs.microsoft.com/learn/modules/train-local-model-with-azure-mls

Azure Machine Learning training run documentation
https://docs.microsoft.com/azure/machine-learning/how-to-set-up-training-targets

Microsoft Azure

Module 4: Working with Data

Agenda

Working with Datastores

Working with Datasets

Working with Datastores

What are Datastores?

Abstractions for cloud data sources

- Azure Storage
- Azure Data Lake
- Azure SQL Database
- Azure Databricks File System
- Others

Built-in Datastores

- workspaceblobstore (default)
- workspacefilestore
- azureml_globaldatasets*

* Added when open datasets are used

Working with Datastores

Add a datastore in Azure Machine Learning studio

or

Use the Azure Machine Learning SDK:

```
from azureml.core import Workspace, Datastore

ws = Workspace.from_config()

blob_ds = Datastore.register_azure_blob_container(workspace=ws,
                                                  datastore_name='blob_data',
                                                  container_name='data_container',
                                                  account_name='az_store_acct',
                                                  account_key='123456abcde789…')

ds = Datastore.get(ws, datastore_name='blob_data')

ds.upload(src_dir='/files', target_path='/data/files')
ds.download(target_path='downloads', prefix='/data')
```

Register a new datastore of a specific type

Get registered datastore by name

Add or retrieve data

Considerations for Datastores

- ✓ Configure blob storage performance type and replication for your needs
- ✓ *Parquet* file format generally performs better than *CSV*
- ✓ You can manage the default datastore using the SDK

```
ws.set_default_datastore(my_datastore)
...
ds = ws.get_default_datastore()
```

Working with Datasets

What are Datasets?

Versioned data objects for experiments

Usually based on datastore contents

Two types:

- *Tabular* datasets: Easy conversion to Pandas dataframe format for structured data files
- *File* datasets: Collection of file references for structured or unstructured data

Creating and Registering Datasets

Add a dataset in Azure Machine Learning studio

or

Use the Dataset object in the SDK

```
from azureml.core import Dataset

csv_paths = [(blob_ds, 'data/files/current_data.csv'),(blob_ds, 'data/files/archive/*.csv')]
tab_ds = Dataset.Tabular.from_delimited_files(path=csv_paths)
tab_ds = tab_ds.register(workspace=ws, name='csv_table')

csv_ds = ws.datasets['csv_table']
```

Create tabular dataset

Register in workspace

Retrieve (in this case from workspace **datasets** collection)

```
from azureml.core import Dataset

file_ds = Dataset.File.from_files(path=(blob_ds, 'data/files/images/*.jpg'))
file_ds = file_ds.register(workspace=ws, name='img_files')

img_ds = Dataset.get_by_name(ws, 'img_files')
```

Create file dataset

Register in workspace

Retrieve (in this case from **Dataset** class by name)

Working with Tabular Datasets

Pass a dataset as a script argument

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
                     script='script.py',
                     arguments=['--ds', tab_ds],
                     environment=env)
```

Required to work with datasets in script

Pass dataset object as script argument

Script:

```
from azureml.core import Run, Dataset

parser.add_argument('--ds', type=str, dest='ds_id')
args = parser.parse_args()

run = Run.get_context()
ws = run.experiment.workspace
dataset = Dataset.get_by_id(ws, id=args.ds_id)
data = dataset.to_pandas_dataframe()
```

Argument contains dataset ID

Get dataset by ID

Convert to dataframe

or

Pass a dataset as a named input

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
        script='script.py',
        arguments=['--ds', tab_ds.as_named_input('my_ds')],
        environment=env)
```

Required to work with datasets in script

Pass dataset as named input

Script:

```
from azureml.core import Run

parser.add_argument('--ds', type=str, dest='ds_id')
args = parser.parse_args()

run = Run.get_context()
dataset = run.input_datasets['my_ds']
data = dataset.to_pandas_dataframe()
```

Argument still required!

Retrieve named dataset from input_datasets

Convert to dataframe

Working with File Datasets

Pass a dataset as a script argument

ScriptRunConfig:

Required to work with datasets in script

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
              script='script.py',
              arguments=['--ds', file_ds.as_download()],
              environment=env)
```

Pass dataset object as *download* or *mount*

Script:

Argument contains data reference

```
from azureml.core import Run
import glob

parser.add_argument('--ds', type=str, dest='ds_ref')
args = parser.parse_args()
run = Run.get_context()

imgs = glob.glob(ds_ref + "/*.jpg")
```

Get file paths from data reference

or

Pass a dataset as a named input

ScriptRunConfig:

```
env = Environment('my_env')
packages = CondaDependencies.create(
     pip_packages=[...,'azureml-dataprep[pandas]'])
env.python.conda_dependencies = packages

sc = ScriptRunConfig(source_directory='my_dir',
              script='script.py',
              arguments=['--ds',
                  file_ds.as_named_input('my_ds').as_download()],
              environment=env)
```

Pass dataset as named input

Script:

Argument still required!

```
from azureml.core import Run
import glob

parser.add_argument('--ds', type=str, dest='ds_ref')
args = parser.parse_args()
run = Run.get_context()

dataset = run.input_datasets['my_ds']
imgs= glob.glob(dataset + "/*.jpg")
```

Retrieve named dataset from input_datasets

Get file paths from data reference

Dataset Versioning

Create a new version of an existing dataset

```
# add .png files to dataset definition
img_paths = [(blob_ds, 'data/files/images/*.jpg'),(blob_ds, 'data/files/images/*.png')]
file_ds = Dataset.File.from_files(path=img_paths)
file_ds = file_ds.register(workspace=ws, name='img_files', create_new_version=True)
```

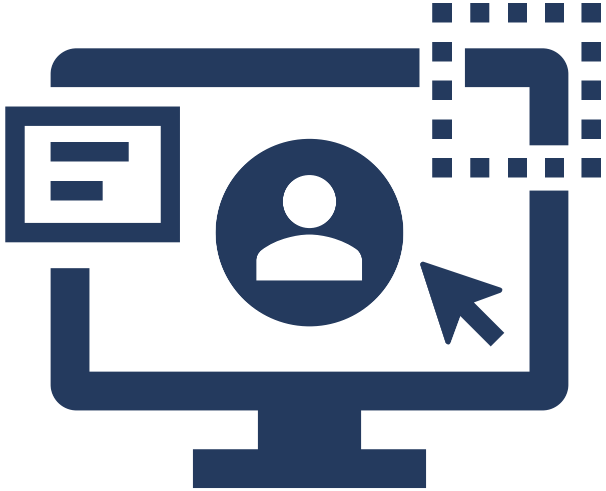
Auto-increments version if a dataset of the same name exists

Specify a version to retrieve

```
ds = Dataset.get_by_name(workspace=ws, name='img_files', version=2)
```

Version number

Lab: Work with Data



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Work with data** exercise

Knowledge check

You have a reference to a Workspace named *ws*.
Which code retrieves the default datastore for the workspace?

- [ ] `default_ds = Datastore.get(ws, 'default')`
- [ ] `default_ds = ws.Datastores[0]`
- [x] `default_ds = ws.get_default_datastore()`

A datastore contains a CSV file of structured data that you want to use as a Pandas dataframe.
Which kind of dataset should you create to make it easy to do this?

- [ ] A file dataset
- [x] A tabular dataset

You want a script to stream data directly from a file dataset. Which mode should you use?

- [x] `as_mount()`
- [ ] `as_download()`
- [ ] `as_upload()`

References

Microsoft Learn: Work with Data in Azure Machine Learning
https://docs.microsoft.com/learn/modules/work-with-data-in-aml/

Azure Machine Learning data documentation
https://docs.microsoft.com/azure/machine-learning/concept-data

Module 5: Working with Compute

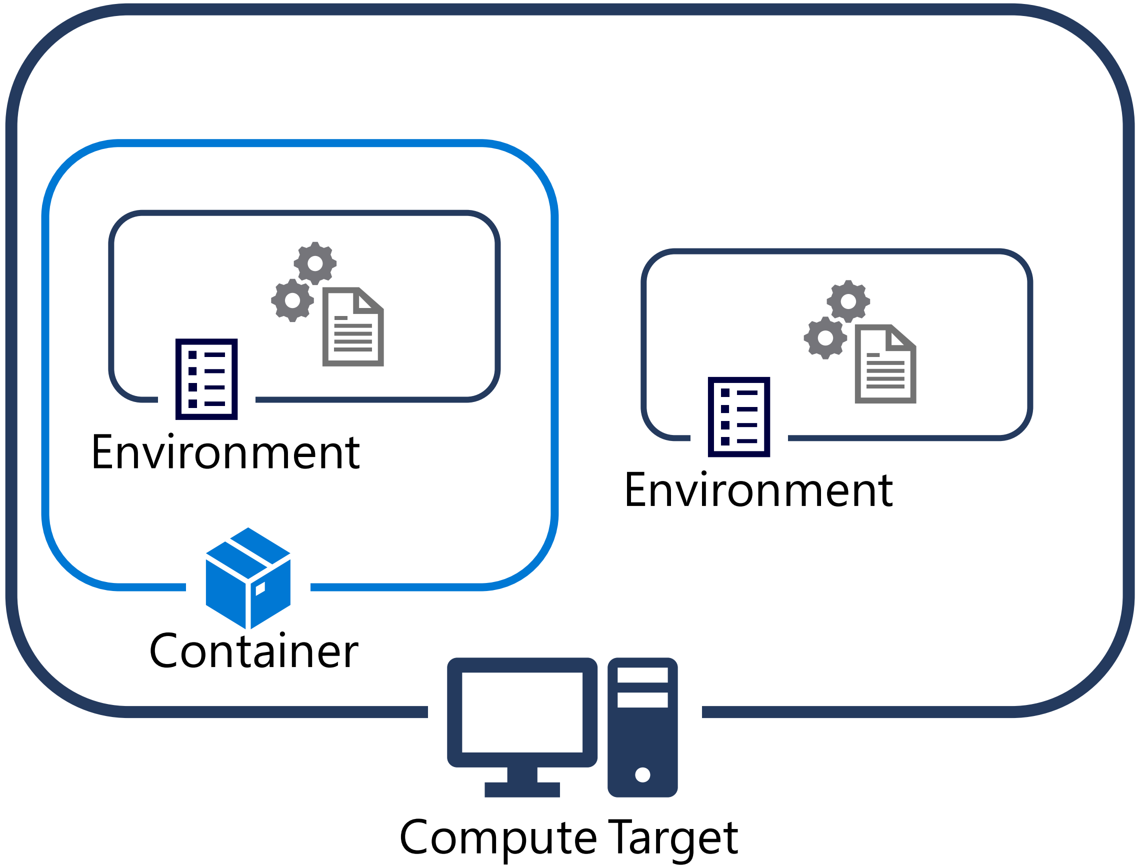
Agenda

- Environments
- Compute Targets

Environments

Run Contexts for Experiments

- Python scripts run in a virtual *environment* that defines the Python version and installed packages
- The environment is usually (but not always) in a *container*
- The container (or environment) is hosted on a *compute target*
    - The default in most cases is the *local* compute (where the control code is run)



Explicitly Creating Environments

Create from specification file

```
env = Environment.from_conda_specification(name='training_environment',
                                           file_path='./conda.yml')
```

File in standard YAML format for Conda environments

Create from existing conda environment

```
env = Environment.from_existing_conda_environment(name='training_environment',
                                                   conda_environment_name='py36')
```

Existing conda environment on local compute

Create with specified packages

Conda package installation is generally more efficient, so use it when possible

```
env = Environment('training_environment')
deps = CondaDependencies.create(conda_packages=['scikit-learn','pandas','pip'],
                                pip_packages=['azureml-defaults'])
env.python.conda_dependencies = deps
```

Most experiments require azureml-defaults

Use conda to install pip if you plan to also install pip packages

Configuring Environment Containers

Use the *docker* section of the environment

```
env.docker.enabled = True
deps = CondaDependencies.create(conda_packages=['scikit-learn','pandas','pip'],
                                pip_packages=['azureml-defaults']
env.python.conda_dependencies = deps
```

Create environment in a container (default)

```
env.docker.base_image='my-base-image'
env.docker.base_image_registry='myregistry.azurecr.io/myimage'
```

Override the default base image with your own prebuilt container image…

```
env.docker.base_image = None
env.docker.base_dockerfile = './Dockerfile'
```

…or create one from a dockerfile

Override managed Python configuration

```
env.python.user_managed_dependencies=True
env.python.interpreter_path = '/opt/miniconda/bin/python'
```

If your image already includes Python and packages, manage dependencies yourself

Registering and Reusing Environments

Register an environment in the workspace

```
env.register(workspace=ws)
```

Saves a definition pf the environment in the workspace for later use

View Registered Environments

```
env_names = Environment.list(workspace=ws)
for env_name in env_names:
    print('Name:',env_name)
```

Azure Machine Learning provides a set of useful *curated* environments with names that begin "AzureML..."

Retrieve and use an environment

Enables you to reuse the environment on any compute target

```
training_env = Environment.get(workspace=ws, name='training_environment')

script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py',
                                environment=training_env)
```

Environment will be created if not already on compute target

Compute Targets

Compute Options for Experiment Runs

Local Compute

- Compute where the control code for the experiment is running
- Often a development workstation or Azure Machine Learning compute instance

Compute Cluster

- Cloud-based cluster managed in an Azure Machine Learning workspace
- Starts, stops, and scales on-demand

Attached Compute

- Azure compute resource outside of a workspace
- For example:
  - Virtual Machine
  - Azure Databricks
  - Azure HDInsight

Creating a Compute Cluster

Create in Azure Machine Learning studio

or

Use the SDK

```
from azureml.core.compute import ComputeTarget, AmlCompute

compute_name = 'aml-cluster'
compute_config = AmlCompute.provisioning_configuration(vm_size='STANDARD_DS11_V2',
                                                       max_nodes=4,
                                                       vm_priority='lowpriority')
aml_compute = ComputeTarget.create(ws, compute_name, compute_config)
aml_compute.wait_for_completion(show_output=True)
```

Specify a suitable Azure VM image
(consider cores, memory, disk, GPU)

Cluster will scale up to this size as required

Low-priority or dedicated
(low-priority can be pre-empted, causing runs to restart; dedicated is more expensive)

Additional options for virtual network and managed identity for access to other Azure resources

Attaching Azure Databricks Compute

Create in Azure Machine Learning studio

or

Use the SDK

```
from azureml.core import Workspace
from azureml.core.compute import ComputeTarget, DatabricksCompute

compute_name = 'db_cluster'

db_workspace_name = 'db_workspace'
db_resource_group = 'db_resource_group'
db_access_token = '1234-abc-5678-defg-90...'
db_config = DatabricksCompute.attach_configuration(resource_group=db_resource_group,
                                                   workspace_name=db_workspace_name,
                                                   access_token=db_access_token)

databricks_compute = ComputeTarget.attach(ws, compute_name, db_config)
databricks_compute.wait_for_completion(True)
```

An existing Azure Databricks workspace in the same Azure subscription as the workspace

Generate a token in the Azure Databricks workspace and specify it here

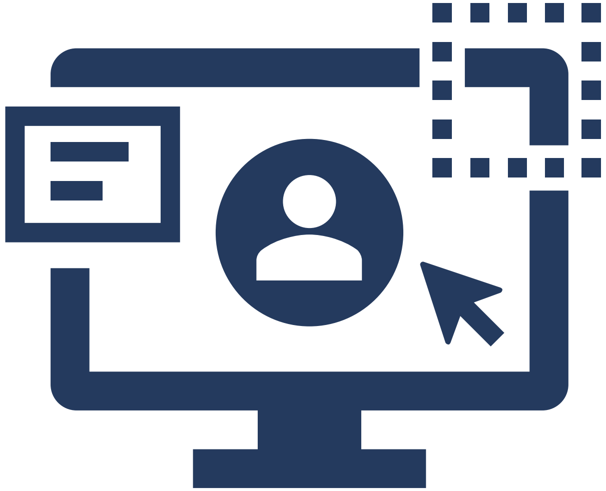
Using Compute Targets

Specify the compute target for an experiment

```
script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py',
                                environment=env,
                                compute_target=compute_name)
```

Specify the compute target name or object

Lab: Work with Compute



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Work with compute** exercise

Knowledge check

You need to create an environment from a Conda configuration (.yml) file.
Which method of the *Environment* class should you use?

- [ ] create
- [x] create_from_conda_specification
- [ ] create_from_existing_conda_environment

You need to run a training script on compute that scales on-demand from 0 to 3 GPU-based nodes.
Which kind of compute target should you create?

- [ ] Compute Instance
- [x] Compute Cluster
- [ ] Inference Cluster

Which ScriptRunConfig parameter causes the script to run on a compute cluster named *train-cluster*?

- [ ] arguments=['--AmlCluster', 'train-cluster']
- [ ] environment='train-cluster'
- [x] compute_target='train-cluster'

References

Microsoft Learn: Work with Compute in Azure Machine Learning
https://docs.microsoft.com/learn/modules/use-compute-contexts-in-aml/

Azure Machine Learning environments documentation
https://docs.microsoft.com/azure/machine-learning/concept-environments

Azure Machine Learning compute targets documentation
https://docs.microsoft.com/azure/machine-learning/concept-compute-target

Microsoft Learn: Perform data science with Azure Databricks
https://docs.microsoft.com/learn/paths/perform-data-science-azure-databricks/

Microsoft Azure

Module 6: Orchestrating Machine Learning Workflows

Agenda

Introduction to Pipelines

Publishing and Running Pipelines

Introduction to Pipelines

What is a Pipeline?

A workflow of machine learning tasks

- Each task is a step
- Steps may be arranged sequentially or in parallel
- Steps can be allocated to specific compute targets

An executable process

- Can be run as an experiment
- Can be published as a REST-based service

The foundation for automating ML operationalization tasks

- Automate data preparation, model training, and deployment
- Trigger based on events or schedules

Pipeline Steps

Common Step Types:

Step Class	Description
PythonScriptStep	Run a Python script
DataTransferStep	Copy data between data stores
DatabricksStep	Run a Databricks notebook, script, or JAR
AdlaStep	Run an Azure Data Lake Analytics U-SQL script
ParallelRunStep	Run a Python script as a distributed task on multiple compute nodes

```
step1 = PythonScriptStep(name='prepare_data', ...)
step2 = PythonScriptStep(name='train_model', ...)
training_pipeline = Pipeline(workspace=ws, steps=[step1,step2])
pipeline_experiment = Experiment(workspace=ws, name='training-pipeline')
pipeline_run = experiment.submit(pipeline_experiment)
```

Passing Data Between Steps

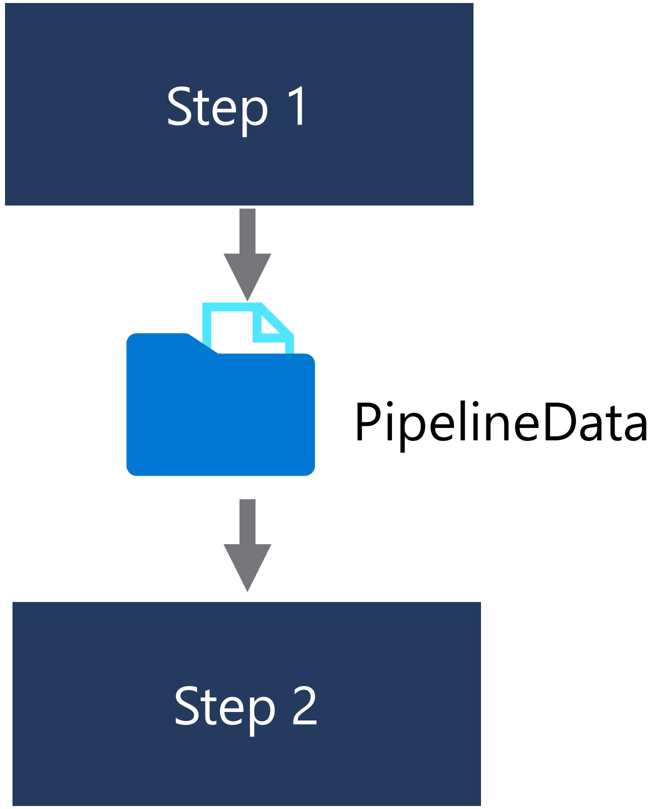
Use a PipelineData object:

- Defines a data reference for an intermediary data store
- Pass as script argument *and* step input/output
- Creates flow dependency between steps

```
data_store = ws.get_default_datastore()
prepped = PipelineData('prepped_data',
                        datastore=data_store)

step1 = PythonScriptStep(name='prepare data',
           arguments=['--out_folder', prepped],
           outputs=[prepped], ...)

step2 = PythonScriptStep(name='train model',
           arguments=['--in_folder', prepped],
           inputs=[prepped], ...)
```

PipelineData output ...

...PipelineData input



Pipeline Step Reuse

Reuse output without re-running the step

Control this behavior with the **allow_reuse** parameter

```
step1 = PythonScriptStep(name='prepare data', arguments = ['--folder', prepped],
                         outputs=[prepped], allow_reuse=True, ...)
```

Reuse cached step output if unchanged

Force all steps to re-run:

Use the **regenerate_outputs** parameter when submitting the experiment

```
pipeline_run = experiment.submit(pipeline_experiment, regenerate_outputs=True)
```

Override step reuse

Publishing and Running Pipelines

Pipeline Endpoints

Publish a pipeline to create a REST endpoint

```
published_pipeline = pipeline_run.publish(name='training_pipeline',
                                          description='Model training pipeline',
                                          version='1.0')
```

Post a JSON request to initiate a pipeline

- Requires an authorization header
- Returns a run ID

```
import requests
response = requests.post(rest_endpoint,
                         headers=auth_header,
                         json={"ExperimentName": "run training pipeline"})
run_id = response.json()["Id"]
```

Pipeline Parameters

Parameterize a pipeline before publishing

Increases flexibility by allowing variable input

```
reg_param = PipelineParameter(name='reg_rate', default_value=0.01)
...
step2 = PythonScriptStep(name='train model',
                      estimator_entry_script_arguments=['--reg', reg_param], ...)
...
published_pipeline = pipeline_run.publish(name='model training pipeline',
                                          description='trains a model with reg parameter',
                                          version='2.0')
```

Pass parameters in the JSON request

```
response = requests.post(rest_endpoint,
                         headers=auth_header,
                         json={"ExperimentName": "run training pipeline",
                               "ParameterAssignments": {"reg_rate": 0.1}})
```

Scheduling Pipelines

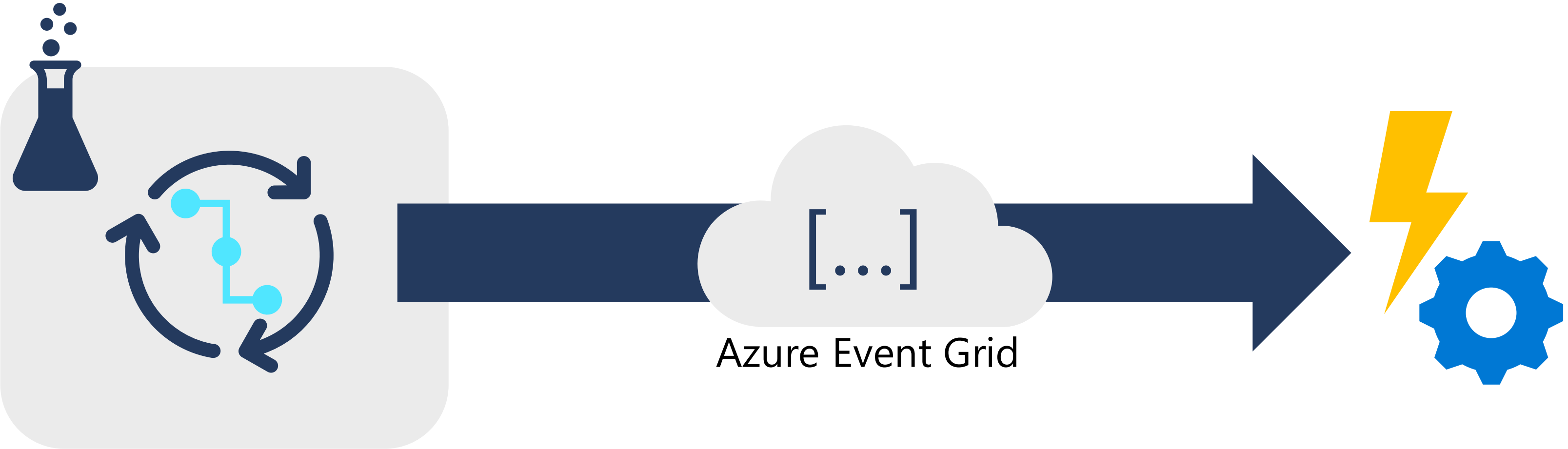
Schedule pipeline runs based on time

```
daily = ScheduleRecurrence(frequency='Day', interval=1)
pipeline_schedule = Schedule.create(ws, name='Daily Training',
                                        description='trains model every day',
                                        pipeline_id=published_pipeline_id,
                                        experiment_name='Training-Pipeline',
                                        recurrence=daily)
```

Trigger pipeline runs when data changes

```
training_datastore = Datastore(workspace=ws, name='blob_data')
pipeline_schedule = Schedule.create(ws, name='Reactive Training',
                                        description='trains model on data change',
                                        pipeline_id=published_pipeline_id,
                                        experiment_name='Training-Pipeline',
                                        datastore=training_datastore,
                                        path_on_datastore='data/training')
```

Event-Driven Workflows



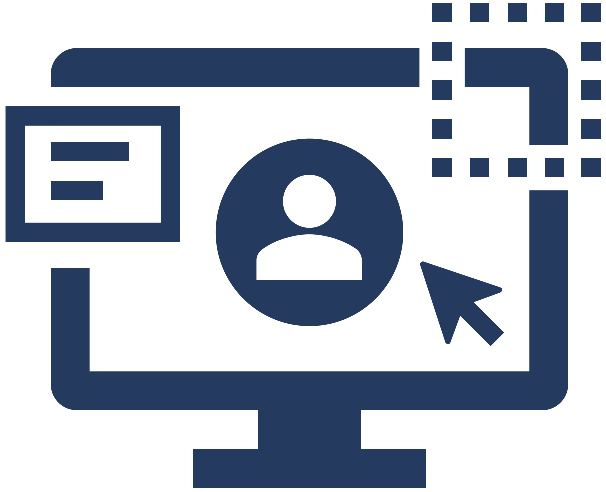
Define events for:

- Run completion
- Run failure
- Model registration
- Model deployment
- Data drift detection

Trigger automated actions:

- Azure Functions
- Azure Logic Apps
- Azure Event Hubs
- Azure Data Factory pipelines
- Generic webhooks

Lab: Create a Pipeline



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create a pipeline** exercise

Knowledge check

What type of object should you use to pass data between pipeline steps?

- [ ] Datastore
- [ ] Dataset
- [x] PipelineData

You plan to use the *Schedule.create* method to create a schedule for a published pipeline. What kind of object must you create first to configure how frequently the pipeline runs?

- [x] ScheduleRecurrance
- [ ] Datastore
- [ ] PipelineParameter

References

Microsoft Learn: Orchestrate machine learning with pipelines
https://docs.microsoft.com/learn/modules/create-pipelines-in-aml/

Azure Machine Learning pipelines documentation
https://docs.microsoft.com/azure/machine-learning/how-to-create-your-first-pipeline

Azure Machine Learning ML Ops documentation
https://docs.microsoft.com/azure/machine-learning/concept-model-management-and-deployment

Azure Machine Learning events documentation
https://docs.microsoft.com/azure/machine-learning/how-to-use-event-grid

Microsoft Azure

Module 7: Deploying and Consuming Models

Agenda

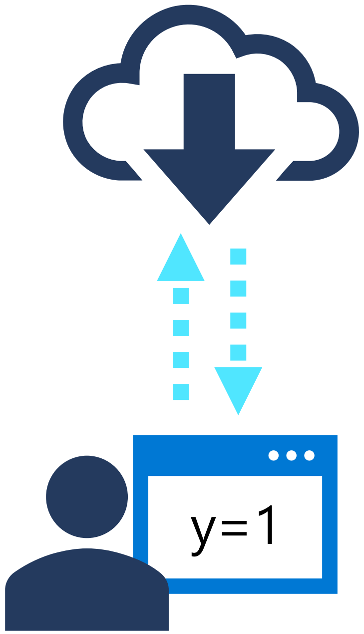
- Real-time Inferencing
- Batch Inferencing

Real-time Inferencing

What is Real-Time Inferencing?

Immediate prediction from new data

Usually deployed as a web service endpoint



Deploying a Real-Time Inferencing Service

1. **Register a trained model**
2. **Define an Inference Configuration**
   - Create a scoring script (implement **init()** and **run()** functions to load the model and return predictions)
   - Create an environment (use a Conda configuration file)
3. **Define a Deployment Configuration**
   - Create a Compute Target (for example: local, Azure Container Instance, AKS cluster)
4. **Deploy the model as a service**

```
service = Model.deploy(ws, 'my_service', [model], inference_config, deploy_config)
```

Consuming a Real-time Inferencing Service

Use the SDK

```
import json

x_new = [[0.1,2.3,4.1,2.0],[0.2,1.8,3.9,2.1]] # Array of feature vectors
json_data = json.dumps({"data": x_new})
response = service.run(input_data = json_data)
predictions = json.loads(response)
```

Use the REST Endpoint

```
import json
import requests

x_new = [[0.1,2.3,4.1,2.0],[0.2,1.8,3.9,2.1]] # Array of feature vectors
json_data = json.dumps({"data": x_new})
request_headers = { 'Content-Type':'application/json' }
response = requests.post(url=endpoint, data=json_data, headers=request_headers)
predictions = json.loads(response.json())
```

Troubleshooting a Real-Time Inferencing Service

Check the service state

```
print(service.state)
```

Review service logs

```
print(service.get_logs())
```

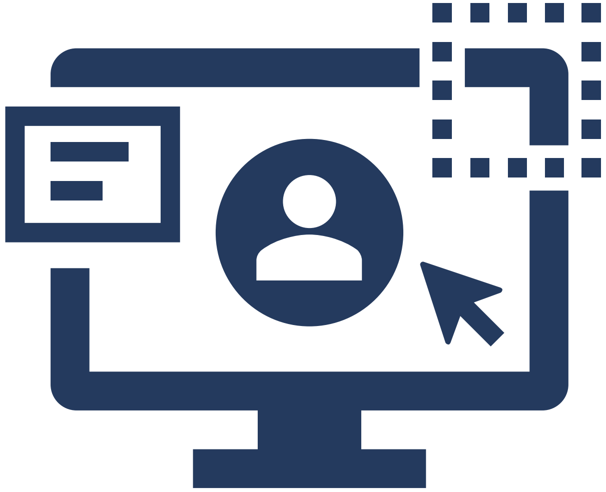
Deploy to a local container

```
deployment_config = LocalWebservice.deploy_configuration(port=8890)
service = Model.deploy(ws, 'test-svc', [model], inference_config, deployment_config)
```

Modify entry script to debug, and then reload to test

```
service.reload()
service.run(input_data=test_sample)
```

Lab: Create a Real-time Inference Service



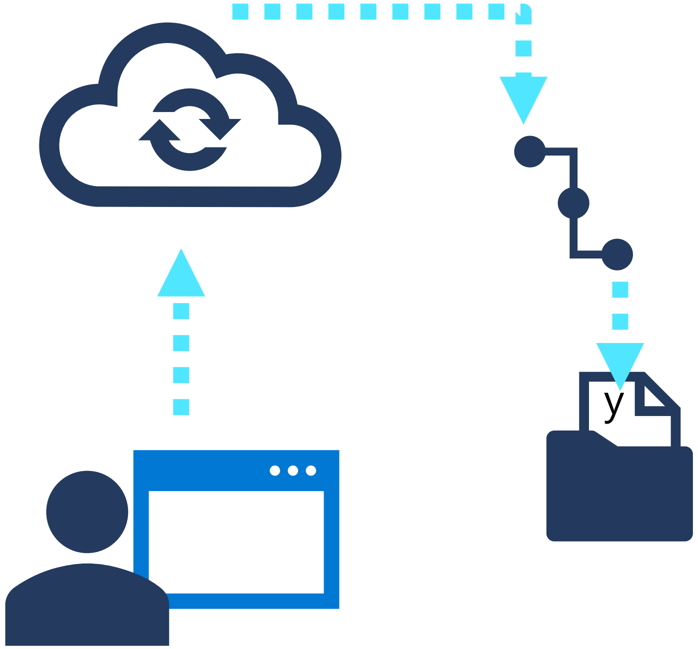
1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create a real-time inference service** exercise

Batch Inferencing

What is Batch Inferencing?

Asynchronous prediction from batched data

Implemented as a pipeline



Creating a Batch Inferencing Pipeline

1. **Register the model**
2. **Create a scoring script**
    - Implement **init()** and **run(mini_batch)** functions to load the model and return predictions for each mini-batch
3. **Create a pipeline with a ParallelRunStep to run the script**
    - Define a **File** dataset input for the batch data
    - Define a **PipelineData** reference for the output folder
    - Configure with an **output_action** of "append_row" so all results are collated in *parallel_run_step.txt*.
4. **Retrieve batch predictions from the output**

Publishing a Batch Inferencing Service

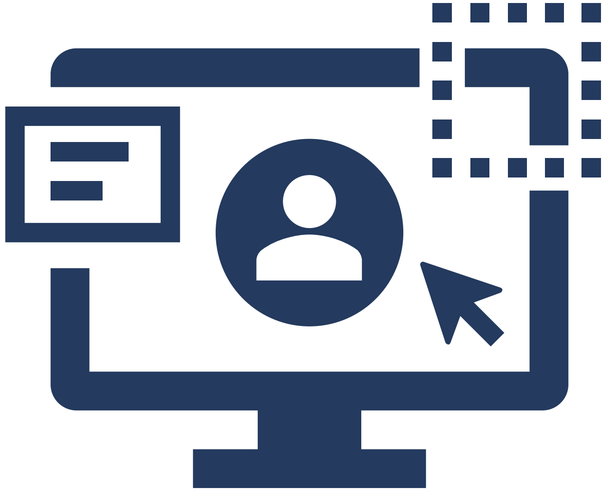
Publish the batch pipeline as a REST service

Use the pipeline endpoint to initiate batch inferencing

```
published_pipeline = pipeline_run.publish_pipeline(name='Batch_Prediction_Pipeline',
                                                    description='Batch pipeline',
                                                    version='1.0')
rest_endpoint = published_pipeline.endpoint
```

```
import requests

response = requests.post(rest_endpoint,
                         headers=auth_header,
                         json={"ExperimentName": "Batch_Prediction"})

run_id = response.json()["Id"]
```

Lab: Create a Batch Inference Service



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Create a batch inference service** exercise

Microsoft Azure

Module 8: Training Optimal Models

Agenda

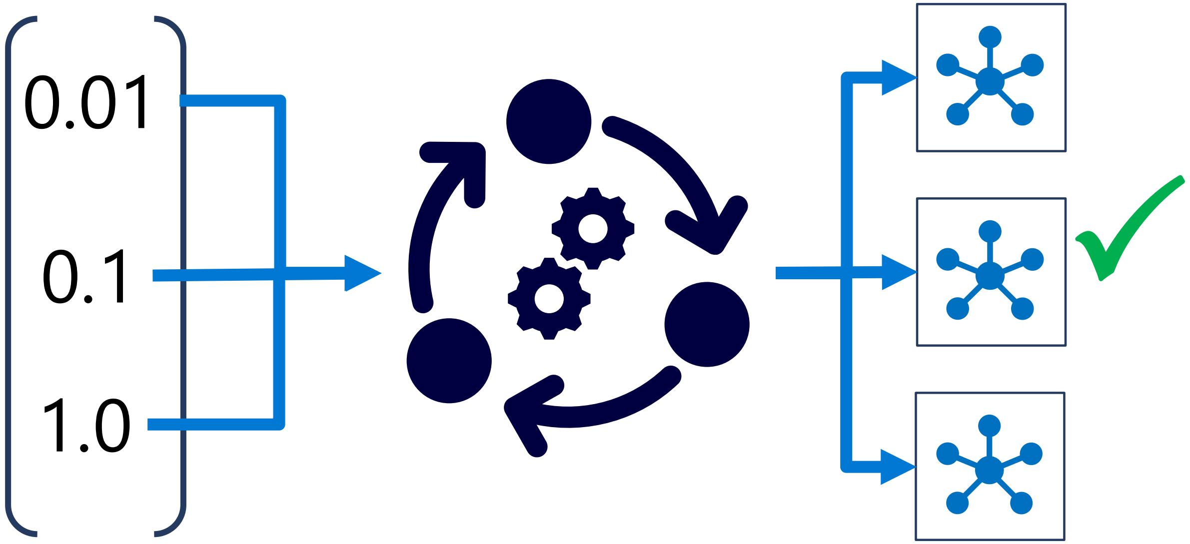
- Hyperparameter Tuning
- Automated Machine Learning

Hyperparameter Tuning

What is Hyperparameter Tuning?

Train multiple models, using the same algorithm but varying hyperparameter values

Find the "best" model based on a specific performance metric



Hyperparameter Search Space

Discrete Hyperparameters

Choice (any list or range)

From a discrete distribution (qnormal, quniform, qlognormal, qloguniform)

Continuous Hyperparameters

From a continuous distribution (normal, uniform, lognormal, loguniform)

```
param_space = {
                 '--batch_size': choice(16, 32, 64),
                 '--learning_rate': normal(10, 3)
              }
```

Hyperparameter Sampling

Grid Sampling

Tries every combination of discrete hyperparameter values

Can only be used when all hyperparameters are discrete

Random Sampling

Randomly selects hyperparameter values

Can be used with discrete and continuous hyperparameter combinations

Bayesian Sampling

Selects hyperparameter values based on performance of previous selection

Can only be used with **choice**, **uniform**, and **quniform** hyperparameters

```
from azureml.train.hyperdrive import RandomParameterSampling

param_sampling = RandomParameterSampling(param_space)
```

Early Termination Policy

Evaluate primary metric at intervals and compare to previous runs

Bandit Policy:

Stop if metric underperforms the best run so far by a specified margin

Median Stopping:

Stop if metric is worse than median of running averages

Truncation Selection:

Stop if metric is in the worst X% of all runs at the same interval

```
from azureml.train.hyperdrive import TruncationSelectionPolicy

stop_policy = TruncationSelectionPolicy(evaluation_interval=1,
                                        truncation_percentage=20,
                                        delay_evaluation=5)
```

Tuning Hyperparameters with Hyperdrive

Experiment script

```
parser.add_argument('--reg', type=float, dest='reg_rate')
...
run.log('Accuracy', model_accuracy)
```

Hyperparameters in sampling collection are passed as arguments

Log performance metric for evaluation

Hyperdrive run configuration

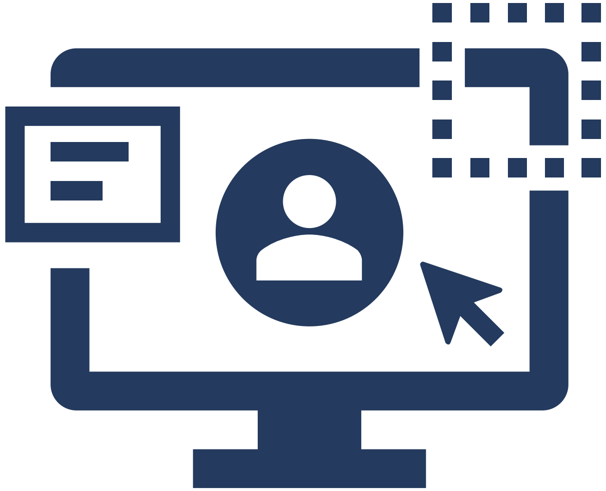
```
hyperdrive = HyperDriveConfig(run_config=script_config,
                              hyperparameter_sampling=param_sampling,
                              policy=stop_policy,
                              primary_metric_name='Accuracy',
                              primary_metric_goal=PrimaryMetricGoal.MAXIMIZE,
                              max_total_runs=6,
                              max_concurrent_runs=4)

hyperdrive_run = experiment.submit(config=hyperdrive)
```

ScriptRunConfig for training script

Params added to script arguments

Name must match logged metric

Lab: Tune Hyperparameters

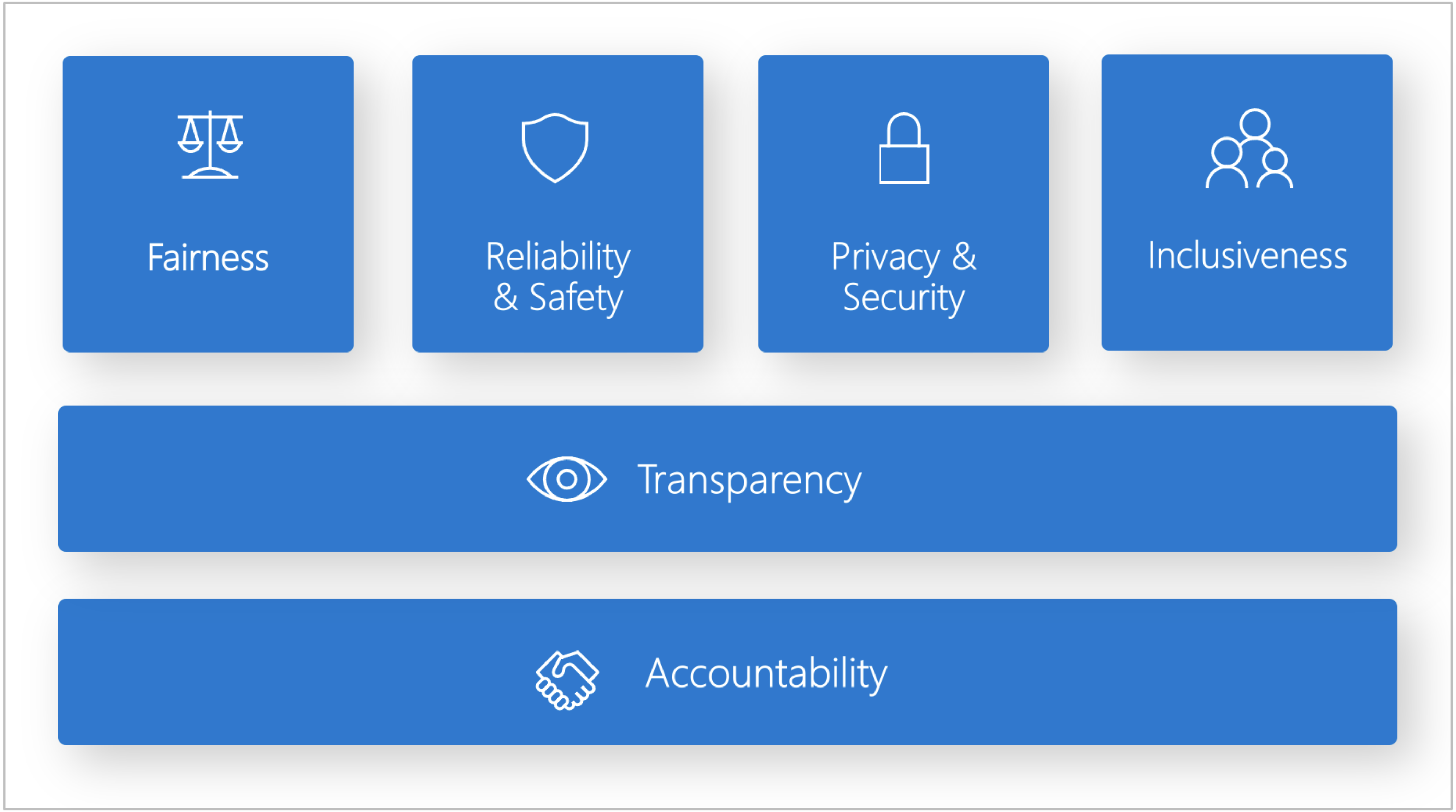


1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Tune hyperparameters** exercise

Microsoft Azure

Module 9: Responsible Machine Learning

Responsible AI standard



Agenda

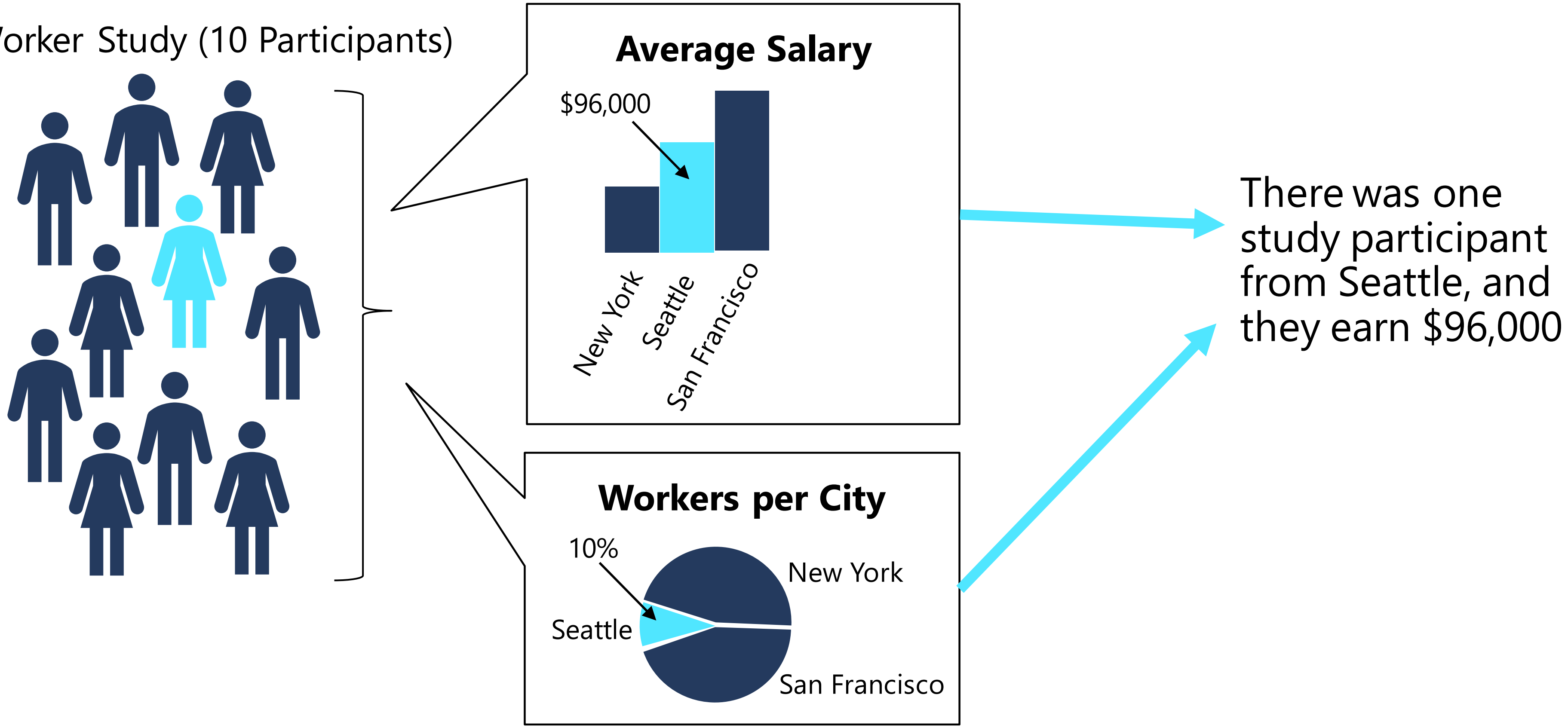
- Differential Privacy
- Model Interpretability
- Fairness

Differential Privacy

The Data Privacy Problem

Studies are ethically and legally required to protect personal information

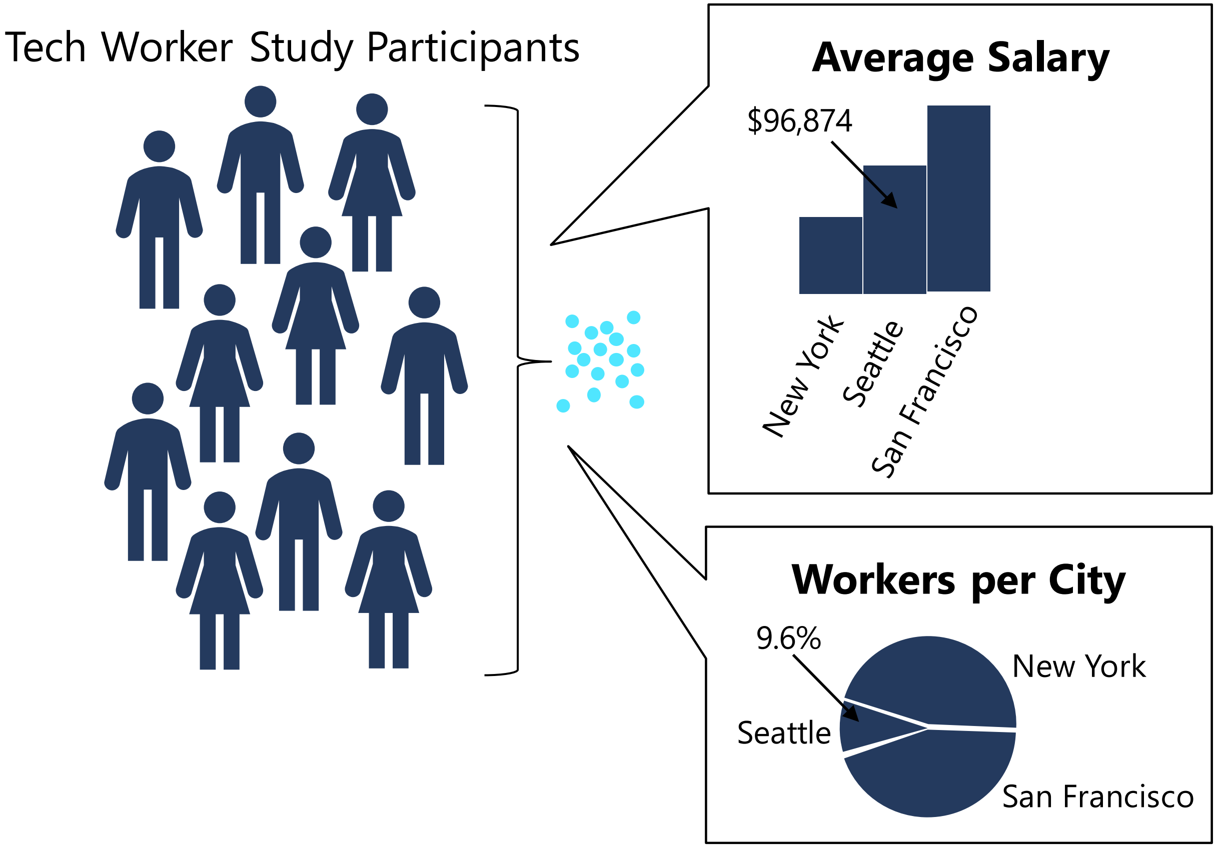
Repeated analyses of aggregated results can reveal details about individuals



What is Differential Privacy?

The analysis function adds random "noise" to the data

Results are statistically consistent, non-deterministic approximations



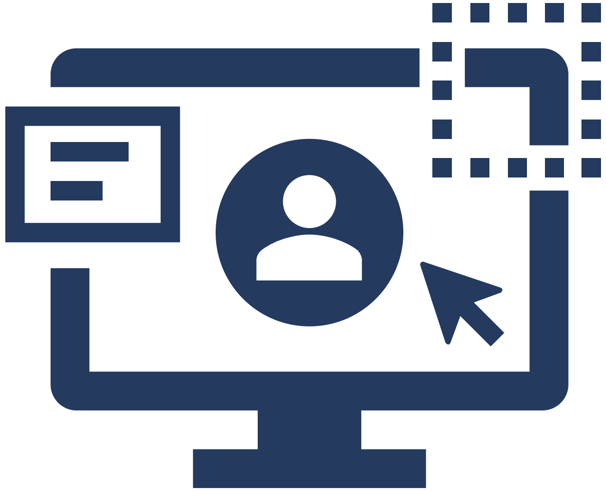
- Each analysis produces slightly different results due to random noise
- Results are statistically consistent with true data distribution allowing for random deviation based on probability
- Individual contributions to the aggregated values are not identifiable

Epsilon - The Privacy Loss Parameter

- **To minimize risk of personal identification, an individual could *opt out* of a study**
  - To be effective for all individuals, they would all need to opt out - so the study would be useless
- **Differential privacy adds noise so the maximum impact of an individual on the outcome of an aggregative analysis is at most *epsilon* (ϵ)**
  - The incremental privacy risk between opting out vs participation for any individual is governed by ϵ
  - Lower ϵ values result in greater privacy but lower accuracy
  - Higher ϵ values result in greater accuracy with higher risk of individual identification

Privacy $\longleftarrow$ ϵ $\longrightarrow$ Accuracy

Lab: Explore Differential Privacy



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Explore differential privacy** exercise

Model Interpretability

Model Interpretability in Azure Machine Learning

Statistical explanation of feature importance

Quantifies the influence of each feature on prediction

Important to identify bias or unintended correlation in the model

Based on the Open Source *Interpret-Community* package

Includes explainers based on common model interpretation algorithms like:

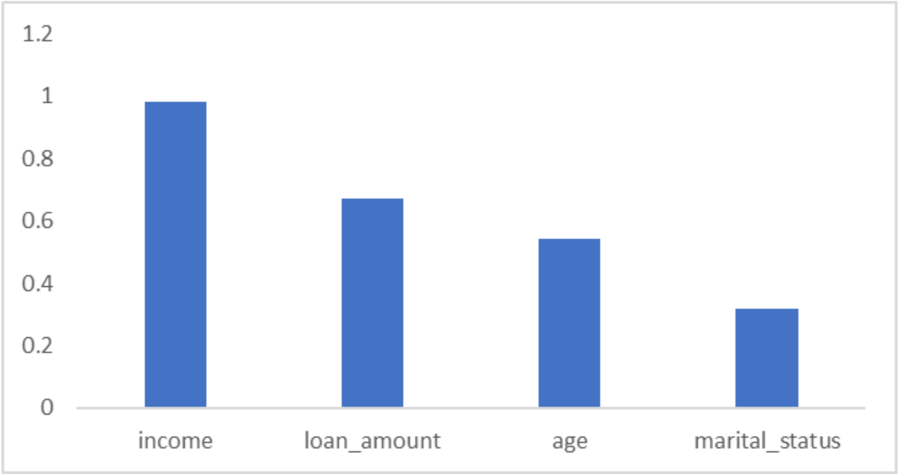
- Shapely Additive Explanations (SHAP)
- Local Interpretable Model-Agnostic Explanations (LIME)

Global and Local Feature Importance

Global Feature Importance

Overall feature importance for all test data

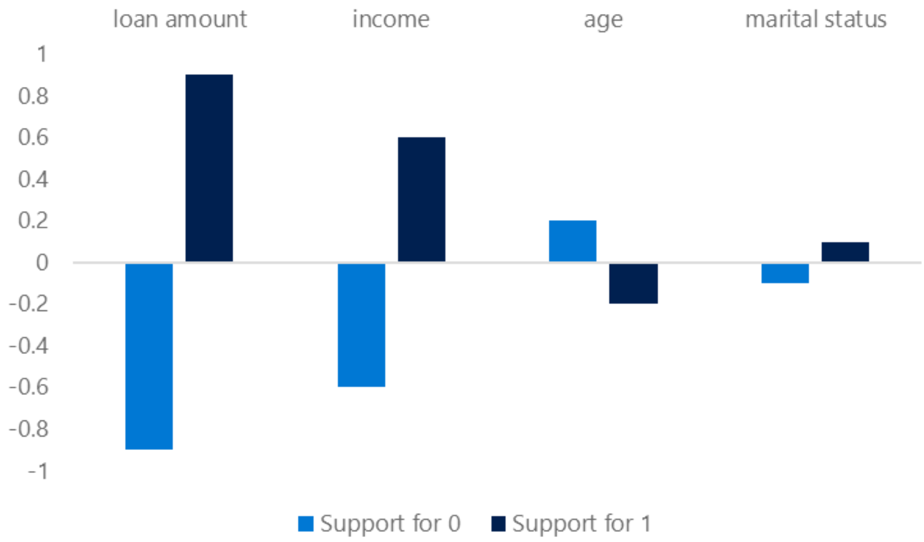
Indicates the relative influence of each feature on the predicted label



Local Feature Importance

Feature importance for an individual prediction

In classification, this shows the relative support for each possible class per feature



Explainers

Use the azureml-interpret package

Create an explainer:

MimicExplainer – global surrogate model that approximates your model

TabularExplainer – Invokes direct SHAP explainer based on model architecture

PFIExplainer – Permutation Feature Importance based on feature shuffling

Get global or local feature explanations

```
from interpret.ext.blackbox import TabularExplainer

tab_explainer = TabularExplainer(model, X_train, features=features, classes=labels)
global_explanation = tab_explainer.explain_global(X_train)
```

Adding Explanations to Training Experiments

In the training script, import the ExplanationClient class

Generate explanations and upload them to the run

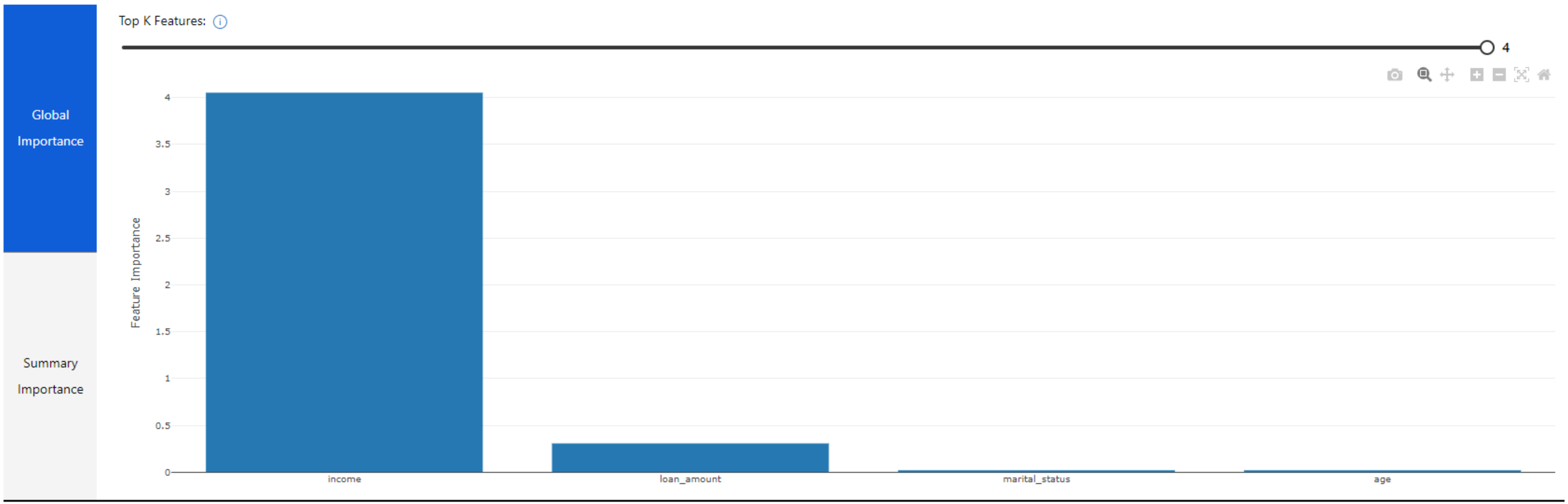
```
explain_client = ExplanationClient.from_run(run)
explainer = MimicExplainer(model, X_train, LinearExplainableModel,
                           features=features, classes=labels)
explanation = explainer.explain_global(X_test)
explain_client.upload_model_explanation(explanation, comment='Model Explanation')
```

Use ExplanationClient to download explanations

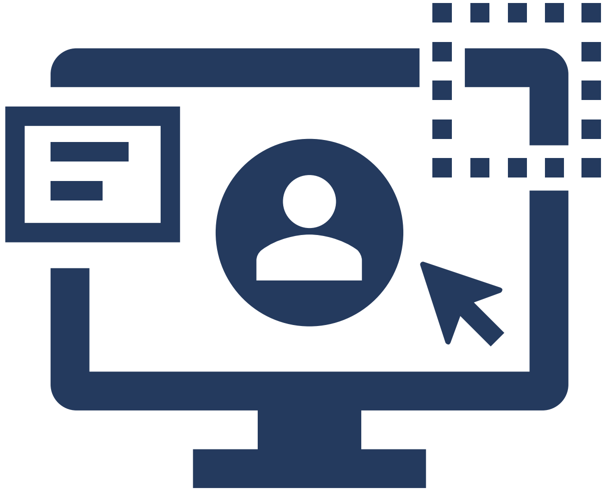
```
from azureml.interpret.explanation_client import ExplanationClient

client = ExplanationClient.from_run_id(workspace=ws,
                                       experiment_name=experiment.experiment_name,
                                       run_id=run.id)
explanation = client.download_model_explanation()
```

Visualizing Model Explanations

View the Explanations tab for the run in Azure Machine Learning studio



Lab: Interpret Models



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Interpret models** exercise

Fairness

What is Fairness?

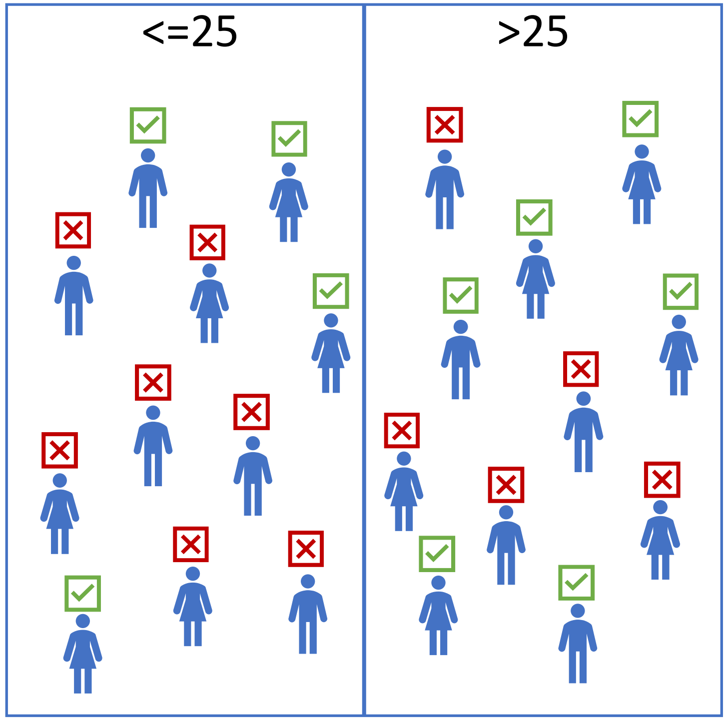
Absence of negative impact on groups based on:

- Ethnicity
- Gender
- Age
- Physical disability
- other sensitive features

Evaluating Model Fairness

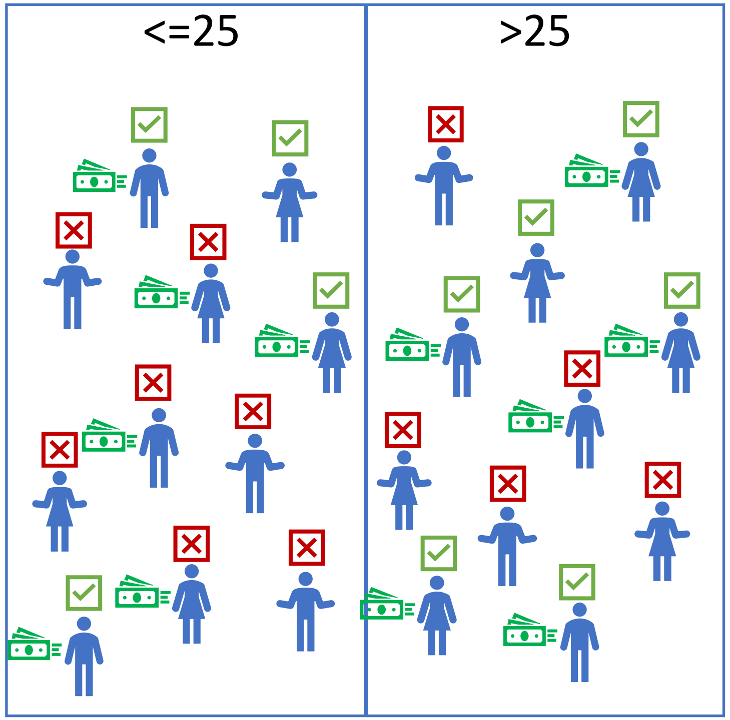
Example: Loan repayment binary classification for two age groups

Selection Rate Disparity



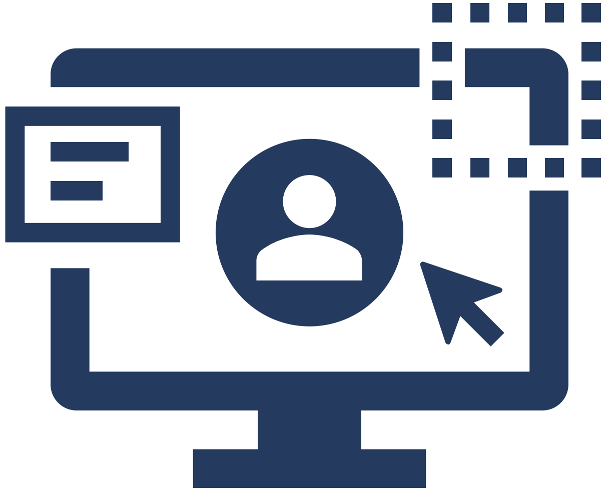
Overall *selection rate* = 10/22 (45%)
25 & under *selection rate* = 4/11 (36%)
Over 25 *selection rate* = 6/11 (54%)
Disparity = 18%

Prediction Performance Disparity



Overall *recall* = 8/12 (67%)
25 & under *recall* = 3/6 (50%)
Over 25 *recall* = 5/6 (83%)
Disparity = 33%

Lab: Detect and Mitigate Unfairness



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Detect and mitigate unfairness** exercise

Microsoft Azure

Module 10: Monitoring Models

Agenda

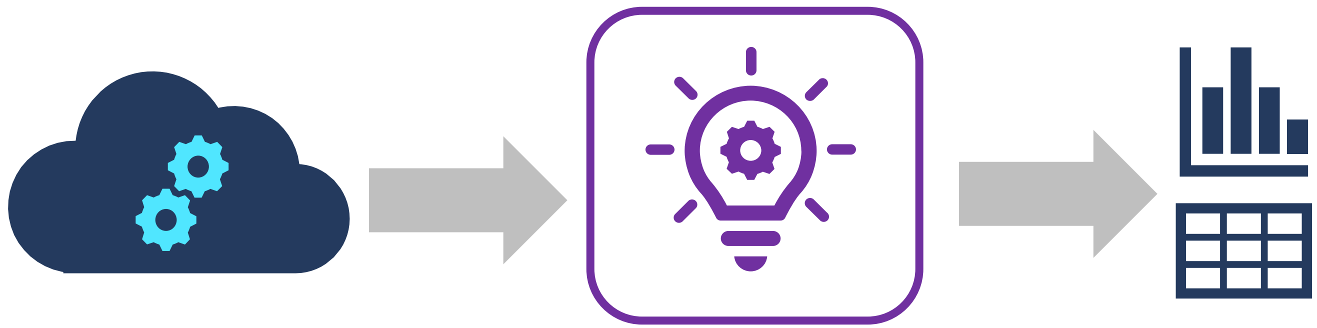
- Monitoring Models with Application Insights
- Monitoring Data Drift

Monitoring Models with Application Insights

What is Application Insights?

An Application Performance Management service in Azure

Enables capture, storage, and analysis of telemetry data



Enabling Application Insights

Determine the Application Insights resource for your workspace

```
ws.get_details()['applicationInsights']
```

Enable in a new service deployment configuration using the SDK:

```
deploy_config = Webservice.deploy_configuration(enable_app_insights=True)
```

Enable for existing deployed services:

Configure AKS deployment in Azure Machine Learning studio

Update deployed service using the SDK

```
service.update(enable_app_insights=True)
```

Capturing and Viewing Application Insights Data

Print log data in the scoring script

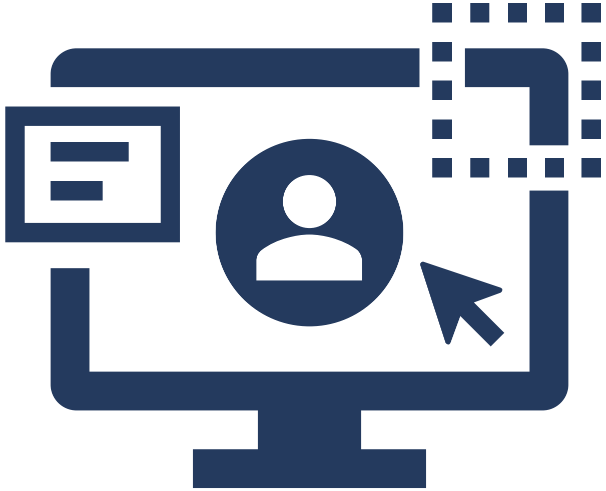
```
def init():
    model = joblib.load(Model.get_model_path('my_model'))
def run(raw_data):
    data = json.loads(raw_data)['data']
    predictions = model.predict(data)
    log_txt = 'Data:' + str(data) + ' - Predictions:' + str(predictions)
    print(log_txt)
```

Query Logs in Application Insights

```
traces
|where  message == "STDOUT" and customDimensions.["Service Name"] = "my-svc"
| project  timestamp, customDimensions.Content
```

timestamp	customDimensions_Content
01/02/2020, 9:11:57.846 PM	Data:[[1, 2, 2.5, 3.1], [0, 1, 1,7, 2.1]] - Predictions:[0 1]

Lab: Monitor a Model



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Monitor a model** exercise

Monitoring Data Drift

What is Data Drift?

Changing data trends that can affect the accuracy of trained models

Creating a Data Drift Monitor

Monitor by Comparing Datasets

Baseline dataset (original training data)

Target dataset for comparison over time (requires timestamp column)

Backfill to populate a data drift profile from target dataset

```
monitor = DataDriftDetector.create_from_datasets(ws, 'dataset-drift-detector',
                                                  baseline_data_set, target_data_set, ...)

backfill = monitor.backfill(dt.datetime.now() - dt.timedelta(days=30), dt.datetime.now())
```

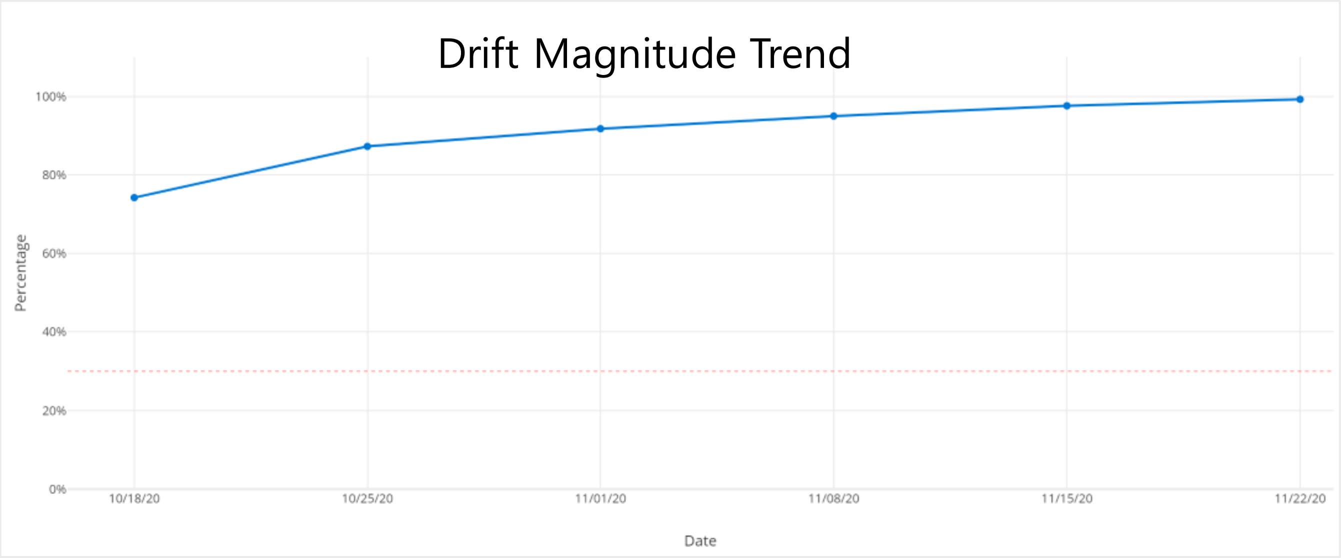
Data Drift Schedules and Alerts

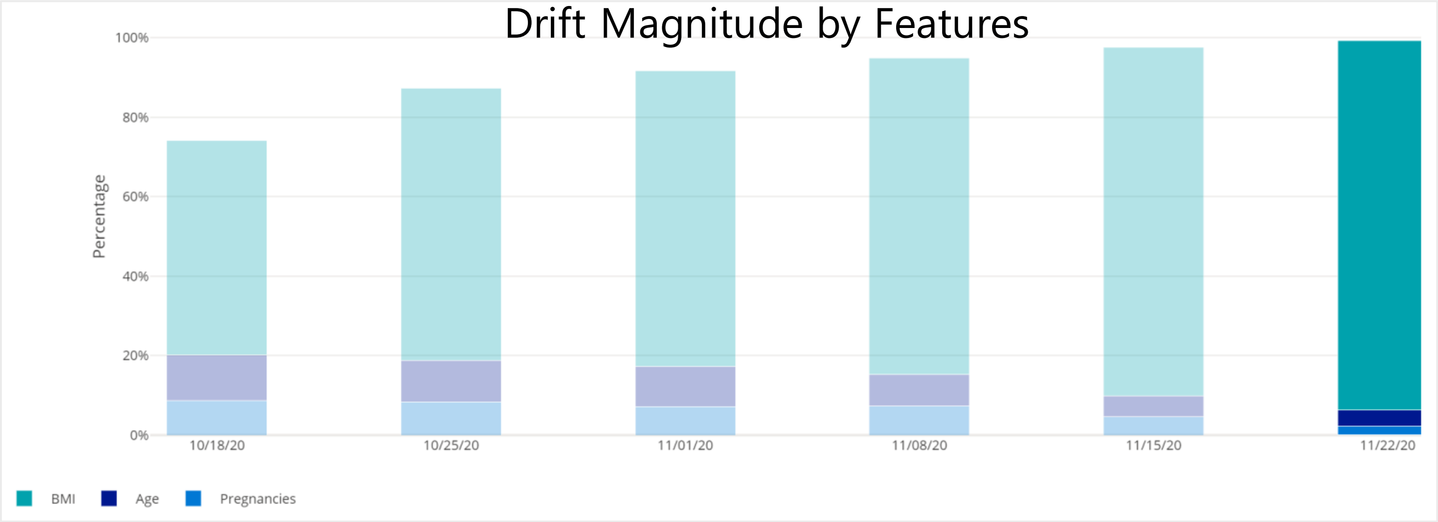
On creation, specify:

- Frequency
- Drift threshold for alerting
- Alert configuration
- Schedule start (for model data drift monitors)
- Data latency (for dataset data drift monitors)

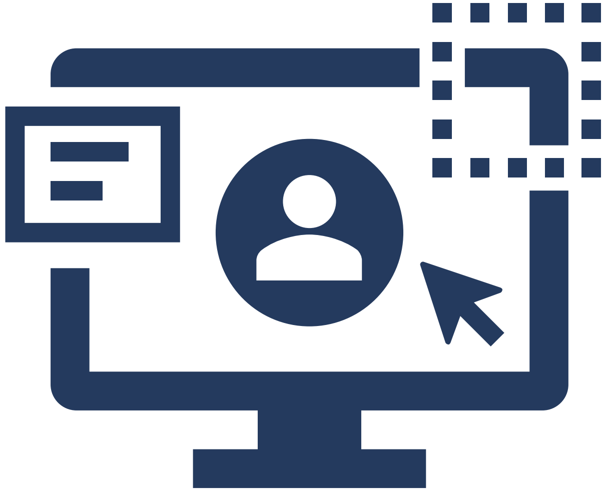
```
alert_email AlertConfiguration('data_scientists@contoso.com')
monitor = DataDriftDetector.create_from_datasets(ws, 'dataset-drift-detector',
                                                 baseline_data_set, target_data_set,
                                                 compute_target=cpu_cluster,
                                                 frequency='Week', latency=2,
                                                 drift_threshold=.3,
                                                 alert_configuration=alert_email)
```

Reviewing Data Drift





Lab: Monitor Data Drift



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Monitor data drift** exercise

Study Resources

Study resources	Links to learning and documentation
Get trained	Choose from self-paced learning paths and modules or take an instructor led course
Find documentation	Azure Databricks Azure Machine Learning
Ask a question	Microsoft Q&A | Microsoft Docs
Get community support	AI - Machine Learning - Microsoft Tech Community AI - Machine Learning Blog - Microsoft Tech Community
Follow Microsoft Learn	Microsoft Learn - Microsoft Tech Community
Find a video	Microsoft Learn Shows

Q&A

Enregistrez vous dès maintenant au prochain Webinars Data AI

Event Webinar (Les jeudis de la Data & AI) - L200/300	Date	Duration (min)	Link
Azure Synapse	22/09/2022	120	https://msevents.microsoft.com/event?id=857781749
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	29/09/2022	120	https://msevents.microsoft.com/event?id=502366997
Déploiement et sécurisation des workspaces Azure Machine learning	06/10/2022	120	https://msevents.microsoft.com/event?id=1505714138
Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share	13/10/2022	120	https://msevents.microsoft.com/event?id=139685175
MLOps avec Azure Machine Learning	20/10/2022	120	https://msevents.microsoft.com/event?id=1245885767
SQL Server 2022 et hybridation native avec Azure SQL Managed Instance	10/11/2022	120	https://msevents.microsoft.com/event?id=145826476
Machine Learning dans Azure Synapse Analytics	17/11/2022	120	https://msevents.microsoft.com/event?id=3637723312
Azure Cosmos DB et IA	24/11/2022	120	https://msevents.microsoft.com/event?id=2646013445
Azure et les Services Cognitifs	08/12/2022	120	https://msevents.microsoft.com/event?id=3772037220
La gouvernance de données dans Azure avec Microsoft Purview	15/12/2022	120	https://msevents.microsoft.com/event?id=1499560981
MLOps avec Azure Machine Learning	12/01/2023	120	https://msevents.microsoft.com/event?id=4115194515
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	19/01/2023	120	https://msevents.microsoft.com/event?id=1537241181
Déploiement et sécurisation des workspace Azure Synapse	26/01/2023	120	https://msevents.microsoft.com/event?id=1806467748
Azure Machine Learning pour les Citizen Data Scientists	09/02/2023	120	En cours
PowerBI - Self Service Analytics	16/02/2023	120	https://msevents.microsoft.com/event?id=1401519679
L'IA responsable avec Azure machine learning	09/03/2023	120	https://msevents.microsoft.com/event?id=2072953112
Machine Learning dans Azure Synapse Analytics	16/03/2023	120	https://msevents.microsoft.com/event?id=3413014857
Les bases de données Open Source dans le cloud Azure	23/03/2023	120	https://msevents.microsoft.com/event?id=2727487131
Hybridation des services de Machine Learning Azure	06/04/2023	120	https://msevents.microsoft.com/event?id=1624914222
La gouvernance de données dans Azure avec Microsoft Purview	13/04/2023	120	https://msevents.microsoft.com/event?id=3909342839
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	04/05/2023	120	https://msevents.microsoft.com/event?id=1162207895
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	16/05/2023	120	https://msevents.microsoft.com/event?id=3517068442
Hybridation des services de données Azure	24/05/2023	120	https://msevents.microsoft.com/event?id=2996507398
Self Service Analytics	01/06/2023	120	En cours

Annexes