Microsoft

Designing and implementing a data science solution on Azure

aka.ms/DP100-Course

1

Course objectives

After completing this course, you should be able to:

- Design and create a suitable **working environment** for data science workloads.
- **Explore** data and **train** machine learning models on Azure Machine Learning.
- Run **jobs** and **pipelines** to prepare your models for production.
- **Deploy** and monitor scalable machine learning solutions.

2

Audience profile

Candidates for this exam:

- Leverage **Azure Machine Learning** to manage their assets and resources for data science workloads.
- Interact with the Azure Machine Learning workspace primarily with the **Python SDK** (v2).
- Track and manage machine learning models with **MLflow**.

3

Course schedule – Day 1

Trainers:
Adjust as necessary. Delete this note before publishing.

Early morning	Course introduction
Break	
Late morning	Design a machine learning solution
Lunch	
Early afternoon	Explore Azure Machine Learning workspace resources and assets Explore developer tools for workspace interaction
Break	
Late afternoon	Make data available in Azure Machine Learning

4

Course schedule – Day 2

Early morning	Work with compute resources in Azure Machine Lea
Break	
Late morning	Work with environments in Azure Machine Learning
Lunch	
Early afternoon	Find the best classification model with Automated Machine Learning
Break	
Late afternoon	Track model training in notebooks with Mlflow

Trainers:
Adjust as necessary. Delete this note before publishing.

5

Course schedule – Day 3

Early morning	Run a training script as a command job in Azure Ma
Break	
Late morning	Use MLflow to track training jobs
Lunch	
Early afternoon	Perform hyperparameter tuning with a sweep job
Break	
Late afternoon	Run pipelines in Azure Machine Learning

Trainers:
Adjust as necessary. Delete this note before publishing.

6

Course schedule – Day 4

Trainers:
Adjust as necessary. Delete this note before publishing.

Early morning	Create and explore the Responsible AI dashboard
Break	
Late morning	Log and register models with MLflow
Lunch	
Early afternoon	Deploy a model to a batch endpoint
Break	
Late afternoon	Deploy a model to a managed online endpoint

7

Get the most out of your Microsoft Learn profile

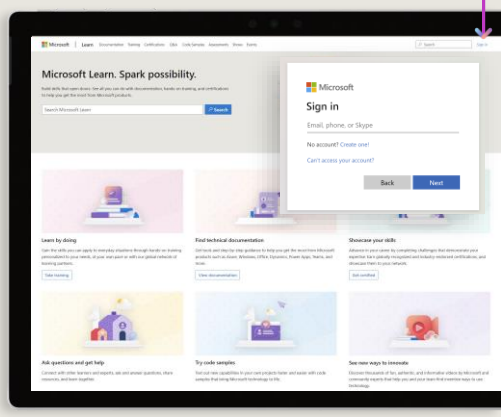
Verify, track, and share your training and certification progress and accomplishments—*all on one platform*

- Claim your achievement code for this course and share you have completed it.
- Access your course material and track progress on your learning activities.
- Share and verify your Microsoft Certifications via email, on social networking platforms, and on your résumé.
- Download and print transcripts and certificates.
- Manage your upcoming activities and certification exam appointments.

www.aka.ms/MyMicrosoftLearnProfile

Create your Microsoft Learn profile at learn.microsoft.com

- Select *Sign in* at the top, right corner of any Microsoft Learn page.
- Follow the Microsoft account authentication process.
- If the account that you have chosen to sign-in with doesn't already have a Microsoft Learn profile, you'll be guided to create one.



8

Access your course material

All course content is available on Microsoft Learn

learn.microsoft.com/training/courses/browse

- We'll go through this content together and as the course progresses, I will advise you on which modules to review.
- You can provide feedback for modules on Microsoft Learn. Find how at the bottom of each page.

Need help? See our troubleshooting guide or provide specific feedback by reporting an issue.

9

This course includes labs:

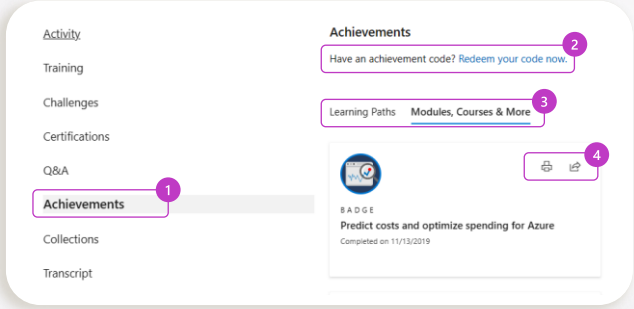
- Detailed lab instructions are included in your lab environment.

aka.ms/DP100-Labs

Celebrate your accomplishments and feel empowered

Get your achievement recognized and earn your badge for completing this course

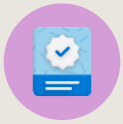
aka.ms/MyMicrosoftLearnProfile



1. Go to **Achievements** on your Microsoft Learn profile
2. Redeem the code provided by your trainer
3. Find your new badge **on Modules, Courses & More**
4. Share your new achievement with your professional network. You can also download and print your certificate.

10

Become Microsoft Certified



92%

of certified IT professionals feel more confident in their abilities after earning certifications[1]

Get recognized by earning industry validation for technical knowledge. Ensure you stay current with the necessary skills and expertise for continued success.

Start at learn.microsoft.com/certifications

Microsoft role-based and specialty certifications require annual renewal.[2]

Learn about certification renewal at aka.ms/RenewYourCert

[1] "2023 Value of IT Certification | Candidate Report," Pearson VUE, 2023 [2] Microsoft fundamentals certifications don't expire

11

Get ready for your Microsoft Certification exam

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

Study area	Percentage
Design and prepare a machine learning solution	20-25%
Explore data and train models	35–40%
Prepare a model for deployment	20–25%
Deploy and retrain a model	10–15%

Exam preparation resources:

- Watch exam prep videos
- Review the exam study guide
- Demo the exam experience with the exam sandbox
- Take a practice assessment

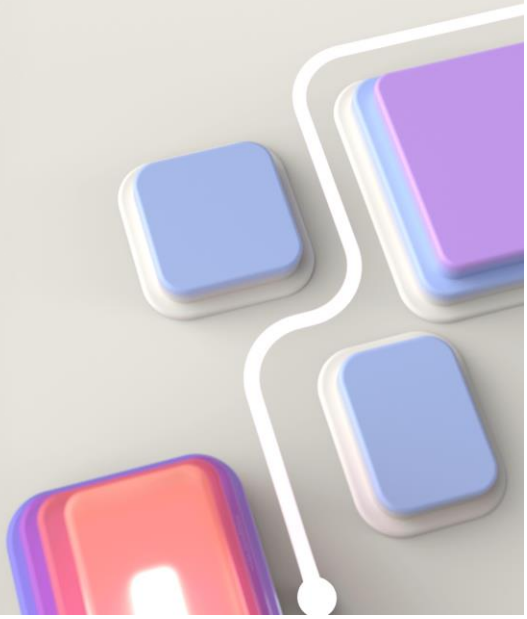
12

Microsoft

Design a machine learning solution

13

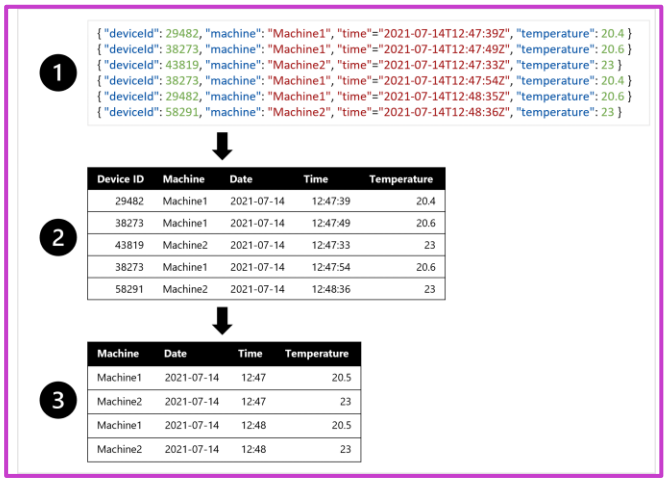
Design a data ingestion solution for machine learning projects



14

Identify your data source and format

1. Identify the **data source**.
2. Identify the **current data format**.
3. Identify the **desired data format**.



15

Choose how to serve data to machine learning workflows

Separate compute from storage:

One of the benefits of the cloud is the ability to scale compute up or down according to your demands.

In addition, you can shut down compute when you don't need it and restart it when you want to use it again.

Store data for model training workloads:

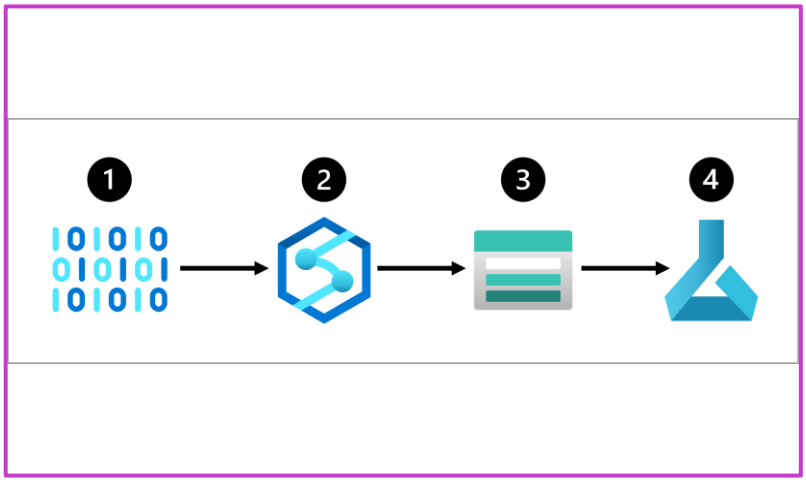
When you use Azure Machine Learning, Azure Databricks, or Azure Synapse Analytics for model training, there are three common options for storing data:

- Azure Blob Storage
- Azure Data Lake Storage (Gen 2)
- Azure SQL Database

16

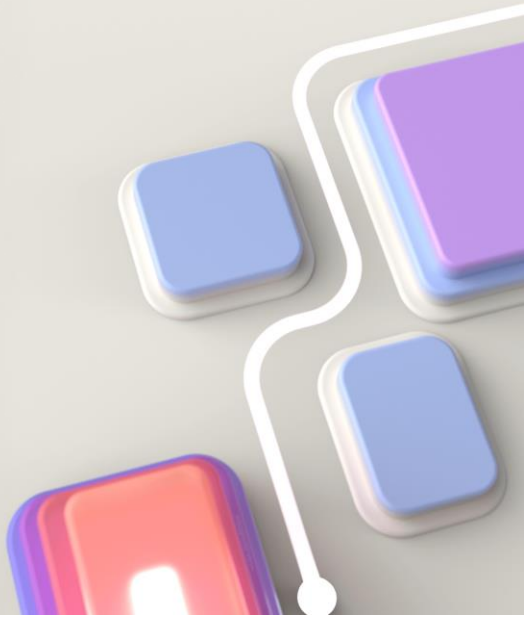
Design a data ingestion solution

To extract data from a source and serve it to your machine learning workspace, create and run a **data ingestion pipeline**.



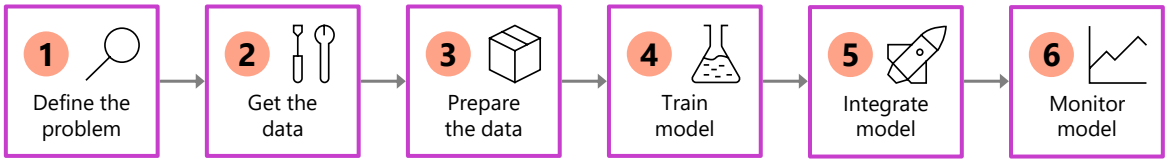
17

Design a machine learning model training solution



18

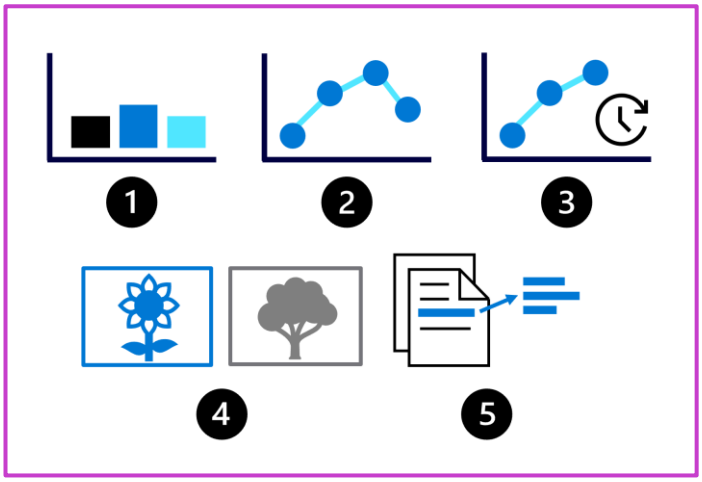
Understand the machine learning process



19

Identify machine learning tasks

1. Classification
2. Regression
3. Time-series forecasting
4. Computer vision
5. Natural language processing (NLP)



20

Choose a service to train a machine learning model

Azure AI Services

- Customize or consume prebuilt models.
- Save time and effort.

Azure Synapse Analytics

- One platform for all data engineering and data science projects at scale.
- Offers an easy-to-use UI, notebooks, and scripts.

Azure Databricks

- Use notebooks for data engineering and data science at scale.
- Offers distributed compute (PySpark).

Azure Machine Learning

- Manage machine learning models from development, to testing, to production.
- Use Python in notebooks and scripts.

21

Decide between compute options

CPU or GPU

CPU will be sufficient and cheaper to use for smaller tabular datasets

Whenever working with unstructured data like images or text, GPUs will be more powerful and effective.

General purpose or memory optimized

General purpose: Have a balanced CPU-to-memory ratio. Ideal for testing and development with smaller datasets.

Memory optimized: Have a high memory-to-CPU ratio. Great for in-memory analytics.

Spark

Services like Azure Synapse Analytics and Azure Databricks offer Spark compute.

A Spark cluster consists of a driver node and worker nodes.

Monitor the compute utilization

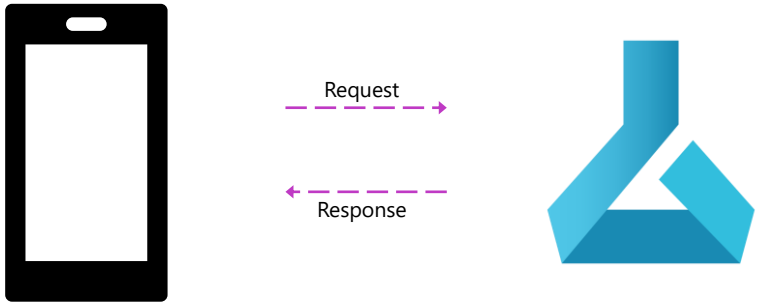
Configuring your compute resources for training a machine learning a model is an iterative process.

22

Design a model deployment solution

23

Understand how model will be consumed



Deploy a model to an endpoint

When you train a model, the goal is often to integrate the model into an application.

To easily integrate a model into an application, you can use endpoints.

24

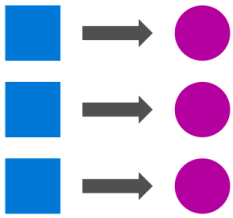
Decide on real-time or batch deployment

When you deploy a model, you have two options:

Get real-time predictions:

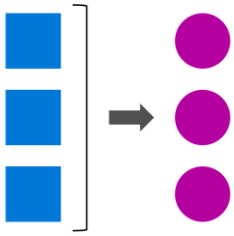
If you want the model to score any new data as it comes in, you need predictions in real-time.

Real-time predictions are often needed when a model is used by an application such as a mobile app or a website.



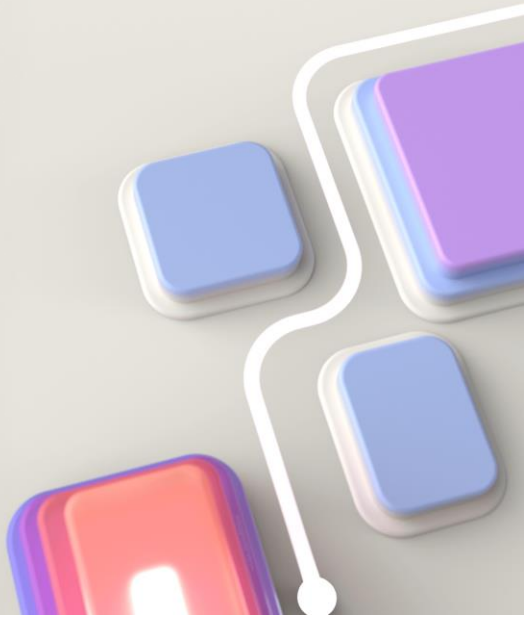
Get batch predictions:

If you want the model to score new data in batches and save the results as a file or in a database, you need batch predictions.



25

Design a machine learning operations (MLOps) solution



26

Understand MLOps

Machine learning operations or **MLOps** help you to scale your model from a proof of concept or pilot project to production.

A model in production is ready for *large-scale deployment* and can be retrained and redeployed when necessary.

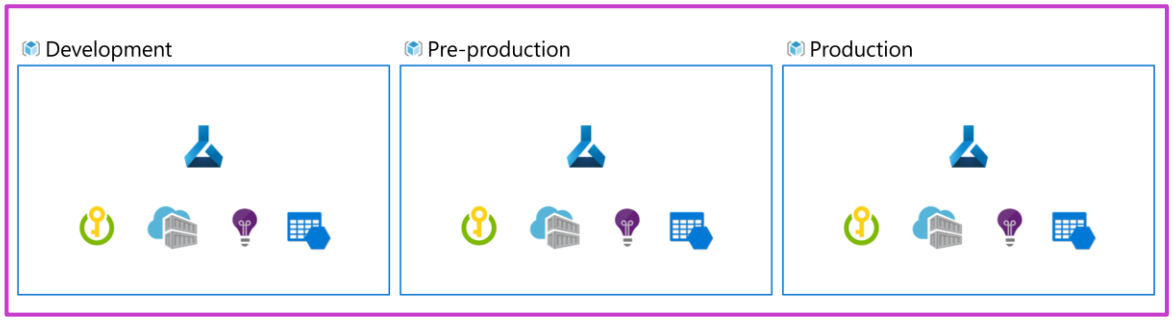
Implementing MLOps helps you to make your machine learning workloads *robust* and *reproducible*.

27

Set up environments for development and production

Within MLOps (and DevOps), an environment refers to a collection of resources.

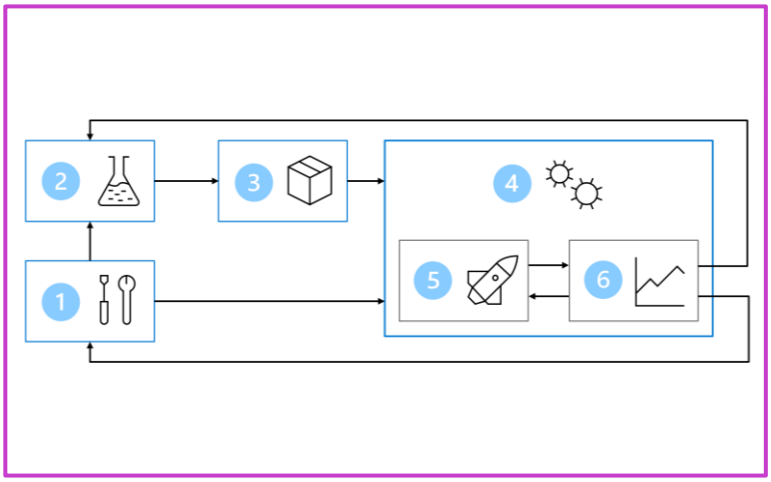
These resources are used to deploy an application, or with machine learning projects, to deploy a model. In an MLOps project, they refer to the Azure resources needed for a phase in the project.



28

Design an MLOps architecture

1. **Setup:** Create all necessary Azure resources for the solution.
2. **Model development (inner loop):** Explore and process the data to train and evaluate the model.
3. **Continuous integration:** Package and register the model.
4. **Model deployment (outer loop):** Deploy the model.
5. **Continuous deployment:** Test the model and promote to production environment.
6. **Monitoring:** Monitor model and endpoint performance.



29

Design for monitoring

Monitor the model:

To monitor a model in production, you can use the trained model to generate predictions on a small subset of new incoming data. By generating the **performance metrics** on that test data, you're able to verify whether the model is still achieving its goal.

Additionally, you may also want to monitor for any **responsible artificial intelligence** (AI) issues.

Monitor the data:

You typically train a machine learning model using a historical dataset that is representative of the new data that your model receives when deployed. However, over time there may be trends that change the profile of the data, making your model less accurate.

This change in data profiles between current and the training data is known as **data drift**, and it can be a significant issue for predictive models used in production.

30

Monitor the infrastructure

Within the workspace:

Review the compute utilization in the Azure Machine Learning workspace for your compute instance and compute cluster. Understanding how much of your compute you're using allows you to scale up to improve performance or scale down to minimize costs.

Use Azure Monitor:

Collect metrics of your Azure resources and set alerts to get notified when problems may arise.

You can monitor the workspace and the managed compute, the datastores, and the endpoints.

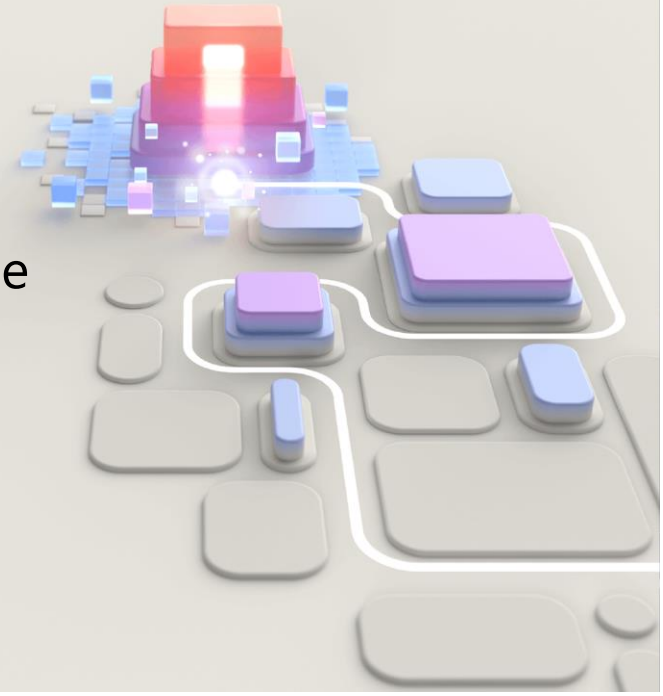
31

Microsoft

32

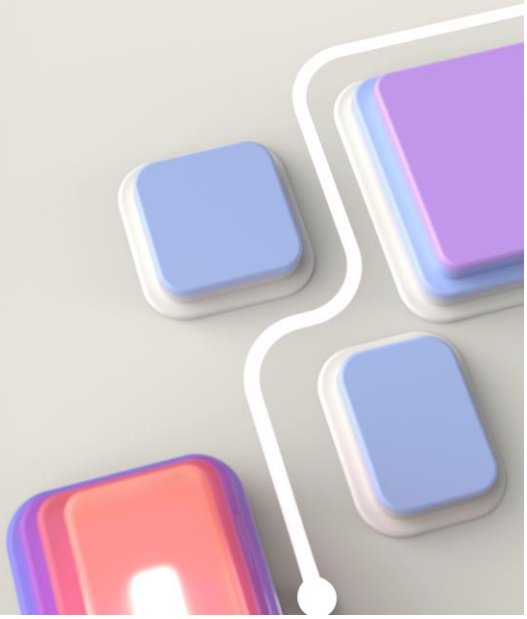
Microsoft

Explore and configure the Azure Machine Learning workspace



33

Explore the Azure Machine Learning workspace resources and assets



34

Introducing Azure Machine Learning

Azure Machine Learning provides a platform for data scientists to train, deploy, and manage their machine learning models on the Microsoft Azure platform.

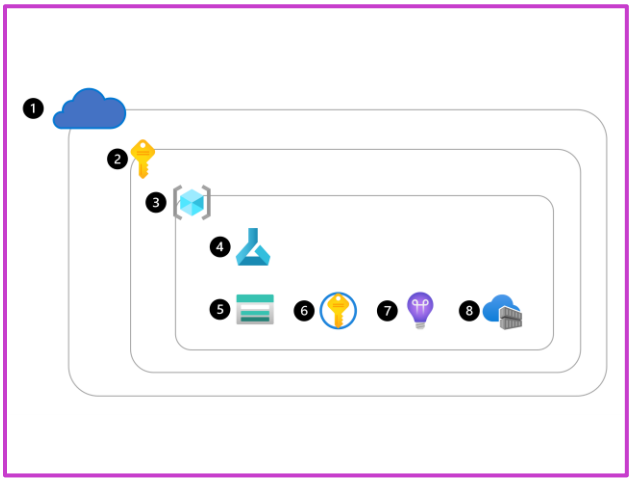
It provides a comprehensive set of resources and assets to train and deploy effective machine learning models.

35

Understand the Azure Machine Learning service

To create an Azure Machine Learning service, you'll have to:

1. Get access to **Azure**.
2. Sign in to get access to an **Azure subscription.**
3. Create a **resource group** within your subscription.
4. Create an **Azure Machine Learning** service, which will automatically also create:
    5. **Azure Storage Account**: Stores all metadata and artifacts.
    6. **Azure Key Vault**: Stores and manages secrets such as authentication keys and credentials.
    7. **Application Insights**: To monitor predictive services in the workspace.
    8. **Azure Container Registry**: Created when needed to store environment images.



36

Create the workspace

You can create a workspace in any of the following ways:

- Use the user interface in the Azure portal.
- Create an Azure Resource Manager (ARM) template.
- Use the Azure Command Line Interface (CLI) with the Azure Machine Learning CLI extension.
- Use the Azure Machine Learning Python SDK.

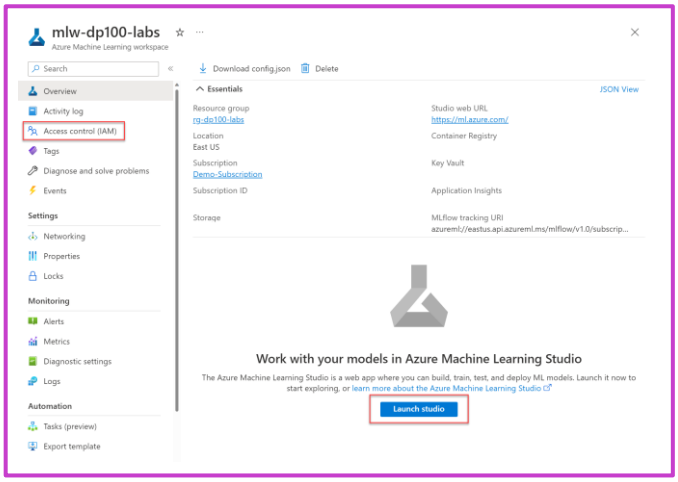
37

Python

```
from azure.ai.ml.entities import Workspace
workspace_name = "mlw-example"
ws_basic = Workspace(
    name=workspace_name,
    location="eastus",
)
ml_client.workspaces.begin_create(ws_basic)
```

Explore the workspace in the Azure portal

- Give others access to the Azure Machine Learning workspace, using the **Access control**.
- Launch the **Azure Machine Learning studio**, an easy-to-use interface to create, manage, and use resources and assets in the workspace.



38

Give access to the Azure Machine Learning workspace

Access is granted in Azure using role-based access control (RBAC)

There are three general built-in roles that you can use across resources and resource groups to assign permissions to other users:

- **Owner:** Gets full access to all resources and can grant access to others using access control.
- **Contributor:** Gets full access to all resources but can't grant access to others.
- **Reader:** Can only view the resource but isn't allowed to make any changes.

Additionally, Azure Machine Learning has specific built-in roles you can use:

- AzureML Data Scientist
- AzureML Compute Operator

To fully customize permissions, create a custom role.

39

Identify Azure Machine Learning resources

1. **The workspace** – The top-level resource for Azure Machine Learning. The workspace keeps an overview of all logs, metrics, outputs, models, and snapshots of your code.

2. **Compute resources** – There are five types of compute in the Azure Machine Learning workspace: compute instances, compute clusters, Kubernetes clusters, attached computes, and serverless compute.

3. **Datastores** – All data is stored in datastores, which are references to Azure data services. Four datastores will exist by default.

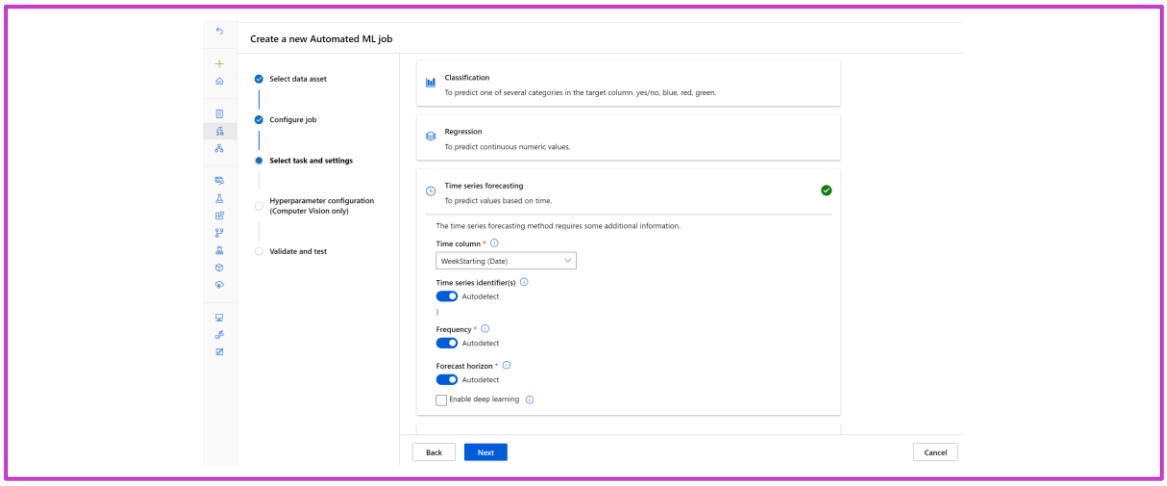
40

Identify Azure Machine Learning assets

1. **Models** – Save trained models in the workspace. A common way to store such models is to package the model as a Python pickle file (.pkl extension).
2. **Environments** – Specify software packages, environment variables, and software settings to run scripts. An environment is stored as an image in the Azure Container Registry created with the workspace when it's used for the first time.
3. **Data** – You can use data assets to easily access data every time, without having to provide authentication every time you want to access it.
4. **Components** – Make it easier to share code with component in a workspace.

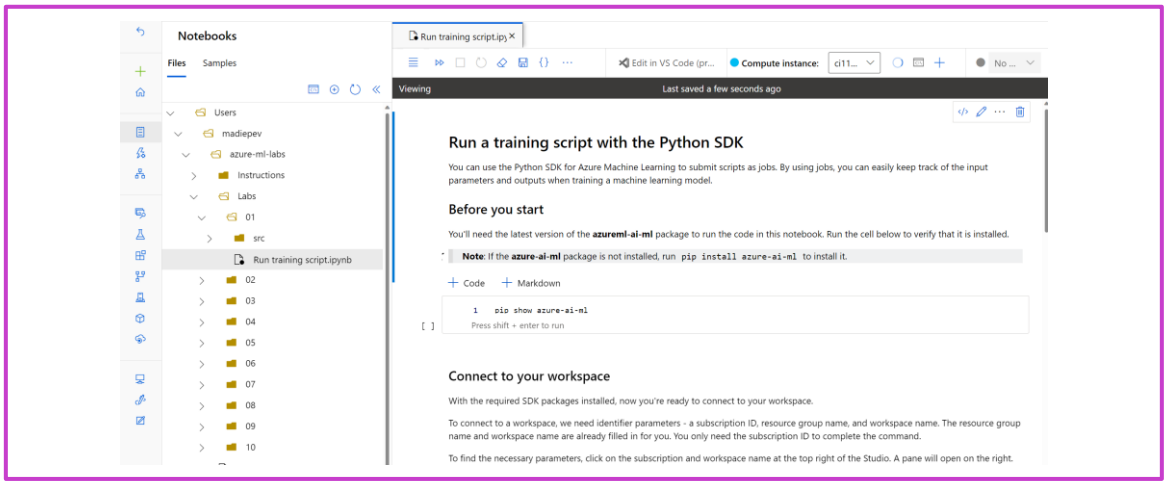
41

Explore algorithms and hyperparameter values with Automated Machine Learning



42

Run a notebook



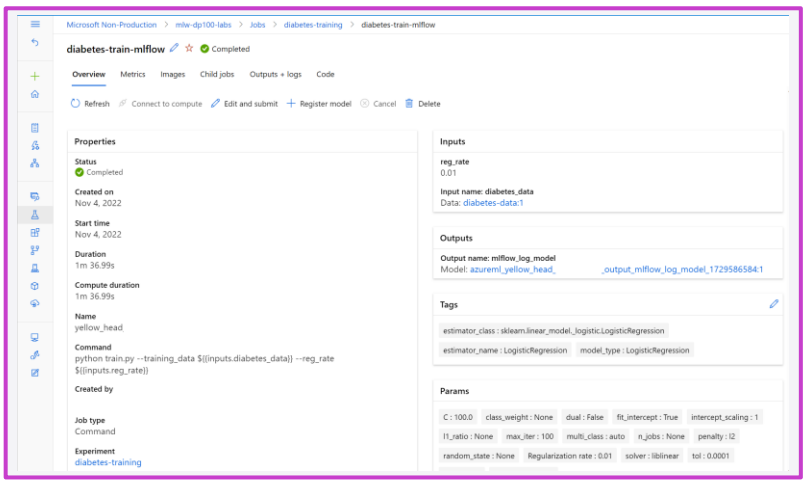
43

Run a script as a job

When you submit a job to the workspace, all inputs and outputs will be stored in the workspace.

There are different types of jobs:

- **Command**: Execute a single script.
- **Sweep**: Perform hyperparameter tuning when executing a single script.
- **Pipeline**: Run a pipeline consisting of multiple scripts or components.



44

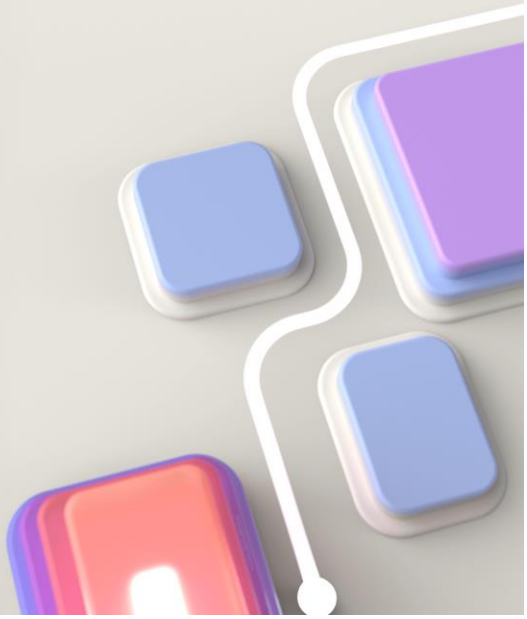
Exercise – Explore the Azure Machine Learning workspace

In this exercise, you learn how to:

- Create an Azure Machine Learning workspace
- Explore the Azure Machine Learning studio
- Author and run a training job

45

Explore developer tools for workspace interaction

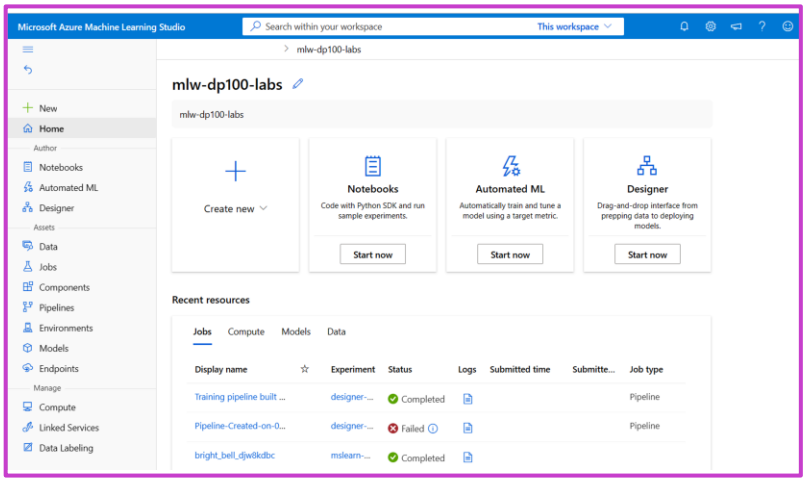


46

Explore the studio

There are two common ways to access the Azure Machine Learning studio:

- Launch the studio from the **Overview** page in the Azure portal
- Navigate to the studio directly by signing in at **https://ml.azure.com**



47

Explore the Python SDK

1. Install the Python SDK

```
pip install azure-ai-ml
```

2. Connect to the workspace

```
from azure.ai.ml import MLClient
from azure.identity import DefaultAzureCredential

ml_client = MLClient(
    DefaultAzureCredential(), subscription_id, resource_group, workspace
)
```


48

Explore the CLI

The Azure CLI is commonly used by administrators and engineers to automate tasks in Azure.

There are many advantages to using the Azure CLI with Azure Machine Learning. The Azure CLI allows you to:

- Automate the creation and configuration of assets and resources to make it **repeatable**.
- Ensure **consistency** for assets and resources that must be replicated in multiple environments
- Incorporate machine learning asset configuration into developer operations **(DevOps) workflows**, such as **continuous integration** and **continuous deployment (CI/CD)** pipelines.

49

Example: Use the Azure CLI to create compute

1. Install on Windows

```
az extension add -n ml -y
```

2. Work with the Azure CLI

```
az ml compute create --file compute.yml --resource-group my-resource-group
--workspace-name my-workspace
```


50

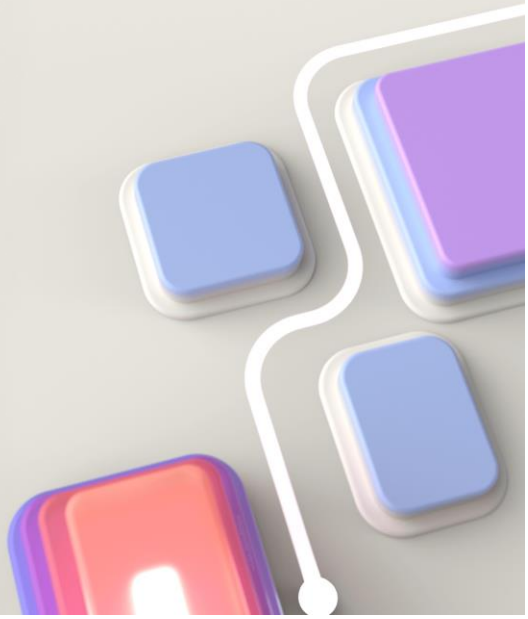
Exercise – Explore developer tools for workspace interaction

In this exercise, you learn how to:

- Create resources with the Azure CLI.
- Explore the Azure Machine Learning workspace with the studio
- Use the Python SDK to train a model

51

Make data available in Azure Machine Learning

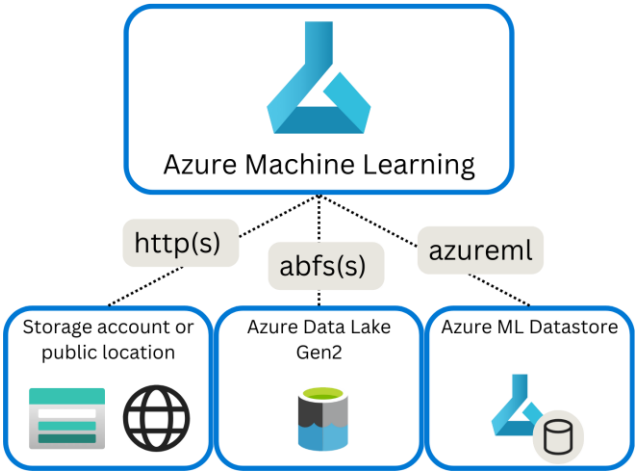


52

Understand URIs

A URI references the **location of your data.**

For Azure Machine Learning to connect to your data directly, you need to prefix the URI with the appropriate protocol.



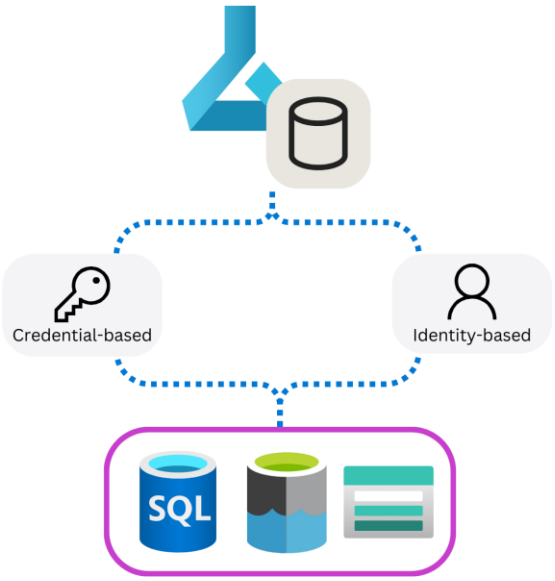
53

Understand datastores

Datastores are **abstractions** for cloud data sources, storing the **connection information**.

The benefits of datastores:

- Provide easy-to-use URIs to your data storage.
- Facilitates data discovery within Azure Machine Learning.
- Securely stores connection information, without exposing secrets and keys to data scientists.



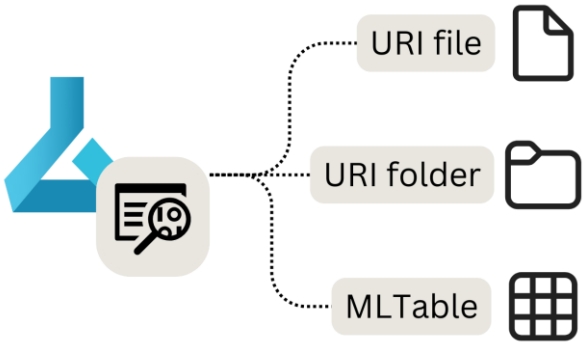
54

Understand data assets

Data assets are **references** to where the data is stored, how to get access, and any other relevant metadata.

The benefits of data assets:

- Share and reuse data with other members.
- Seamlessly access data during model training (on any supported compute type) without worrying about connection strings or data paths.
- Version the metadata of the data asset.



55

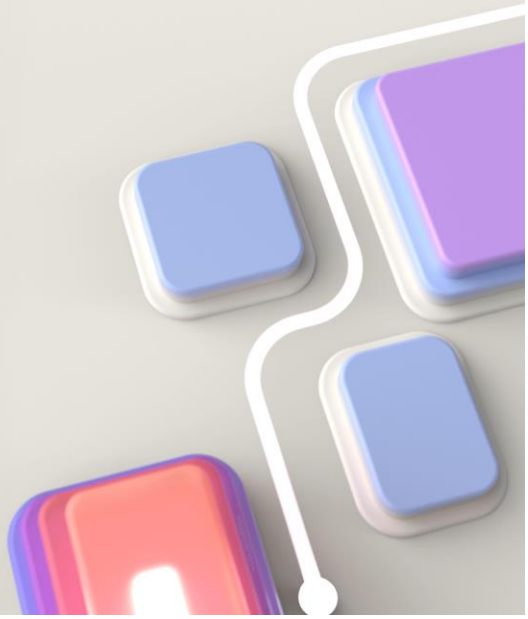
Exercise – Make data available in Azure Machine Learning

In this exercise, you learn how to:

- Explore the default datastores
- Create a datastore
- Create data assets

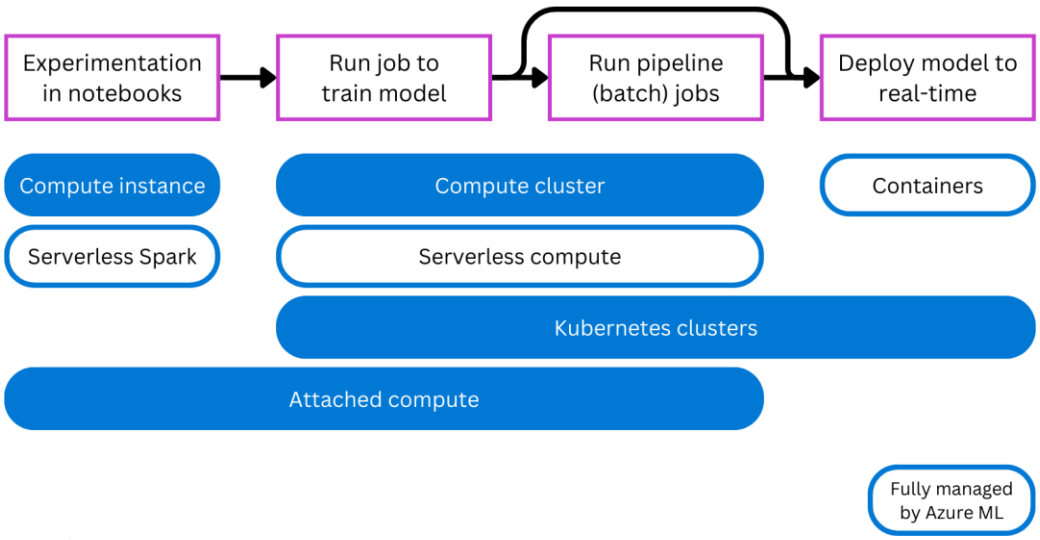
56

Work with compute resources in Azure Machine Learning



57

Choose the appropriate compute target



58

Configure the compute instance

Assign a compute instance to a user

- A compute instance can only be assigned to *one* user, as the compute instance isn't ideal for parallel workloads.
- When you create a new compute instance, you can assign it to someone else if you have the appropriate permissions.

Minimize compute time

- Manually start and stop a compute instance whenever you need it.
- Schedule your compute instance to start or stop at set times to avoid unnecessary costs.
- Configure when to shut down your compute instance due to inactivity.

59

Exercise – Work with compute resources in Azure Machine Learning

In this exercise, you learn how to:

- Create the compute setup script
- Create and use a compute instance
- Create and use a compute cluster

60

Work with environments in Azure Machine Learning

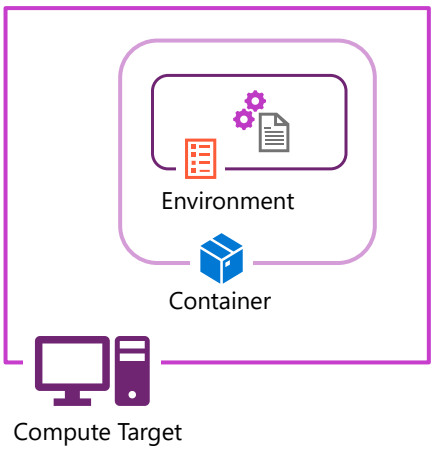
61

Understand environments

Environments define the context (runtime and packages) needed to run code on a compute target.

An environment is used to create the **Docker container** that your code runs in on the specified compute target.

Use a predefined **curated** environment or create your own **custom** environment.



62

Create custom environments

Creating a custom environment within the Azure Machine Learning workspace from a:

- **Docker image**: choose an existing image from a (public) repository
- **Docker build context**: reference a path that includes a Dockerfile and requirements.txt
- **Conda specification**: reference an image and add a conda YAML file that includes additional dependencies

Dockerfile

```
#FROM mcr.microsoft.com/azureml/openmpi4.1.0-ubuntu20.04
FROM python:3.8

# python installs
COPY requirements.txt .
RUN pip install -r requirements.txt && rm requirements.txt

# set command
CMD ["bash"]
```

Conda YAML file

```
name: pydata-example
channels:
  - conda-forge
dependencies:
  - python=3.8
  - pip=21.2.4
  - pip:
    - numpy==1.22
    - scipy==1.7.1
    - pandas==1.3.0
    - scikit-learn==0.24.2
```


63

Exercise – Work with environments in Azure Machine Learning

In this exercise, you learn how to:

- Explore environments
- Run a job with a curated environment
- Create and use a curated environment

64

Microsoft

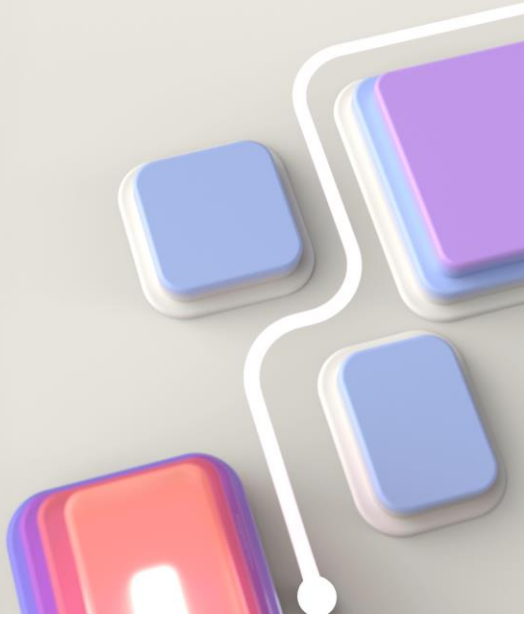
65

Microsoft

Experiment with Azure Machine Learning

66

Explore Automated Machine Learning



67

Explore Automated Machine Learning

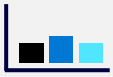
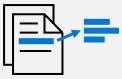
Instead of manually having to test and evaluate various configurations to train a machine learning model, you can automate it with automated machine learning or AutoML.

- Train multiple models in parallel, varying preprocessing and algorithm selection.
- Find the "best" model based on a specific performance metric.

68

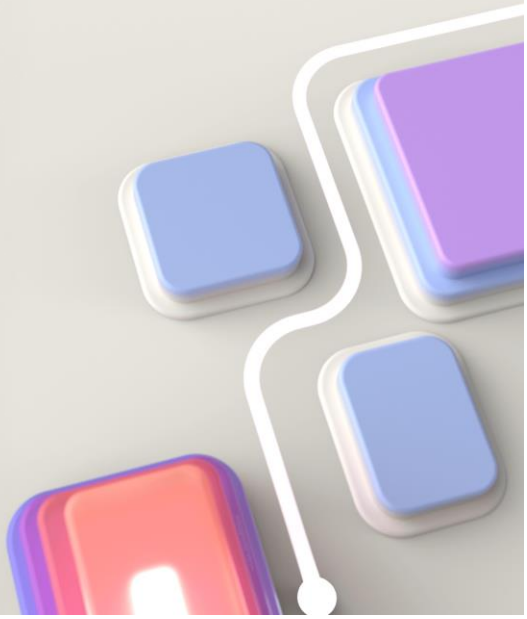
Choose a task

The training data, featurization options, algorithms, and performance metrics will depend on the task you choose.

Classification	Regression	Time-series forecasting	Computer vision	Natural language processing (NLP)
Predict a categorical value.	Predict a numerical value.	Predict future numerical values based on time-series data.	Classify images or detect objects in images.	Text classification or named entity recognition.
				

69

Find the best classification model with Automated Machine Learning



70

Preprocess data and configure featurization

Before you can run an AutoML experiment, you need to prepare your data.

- Classification requires tabular data.
- Create a **data asset** in Azure Machine Learning.
- Create a **MLTable** data asset: Store your data in a folder together with a MLTable file.

Python

```
from azure.ai.ml.constants import AssetTypes
from azure.ai.ml import Input

my_training_data_input = Input(type=AssetTypes.MLTABLE, path="azureml:input-data-automl:1")
```


71

Understand scaling and normalization

AutoML applies scaling and normalization to numeric data automatically, helping prevent any large-scale features from dominating training.
During an AutoML experiment, multiple scaling or normalization techniques will be applied.

Configure optional featurization:

- Missing value imputation to eliminate nulls in the training dataset
- Categorical encoding to convert categorical features to numeric indicators
- Dropping high-cardinality features, such as record IDs
- Feature engineering (for example, deriving individual date parts from DateTime features)

72

Run an Automated Machine Learning experiment

The algorithms AutoML uses will depend on the task you specify.

When you want to train a classification model, AutoML will choose from a list of classification algorithms:

- Logistic Regression
- Light Gradient Boosting Machine (GBM)
- Decision Tree
- Random Forest
- Naïve Bayes
- Linear Support Vector Machine (SVM)
- XGBoost
- And others...

73

Restrict algorithm selection

- By default, AutoML will randomly select from the full range of algorithms for the specified task.
- You can choose to block individual algorithms from being selected; which can be useful if you know that your data isn't suited to a particular type of algorithm.
- You also may want to block certain algorithms if you have to comply with a policy that restricts the type of machine learning algorithms you can use in your organization.

74

Configure an AutoML experiment (1/2)

When you use the Python SDK (v2) to configure an AutoML experiment or job, you configure the experiment using the *automl class.*

1. **Specify the primary metric:** The "best" model is based on the **primary_metric**
2. **Set the limits:** To minimize costs and time spent on training, you can set limits to an AutoML experiment by using *set_limits()*
3. **Set the training properties:** AutoML will try various combinations of featurization and algorithms to train a machine learning model

75

Configure an AutoML experiment (2/2)

To configure an AutoML experiment for classification, use the **automl.classification** function:

Python

```
from azure.ai.ml import automl

classification_job = automl.classification(
                compute="aml-cluster",
                experiment_name="auto-ml-class-dev",
                training_data=my_training_data_input,
                target_column_name="Diabetic",
                primary_metric="accuracy",
                n_cross_validations=5,
                enable_model_explainability=True)
```


76

Set the limits

There are several options to set limits to an AutoML experiment:

- Timeout_minutes: **Number of minutes after which the complete AutoML experiment is terminated.**
- **Trial_timeout_minutes:** Maximum number of minutes one trial can take.
- **Max_trials:** Maximum number of trials, or models that will be trained.
- **Enable_early_termination:** Whether to end the experiment if the score isn't improving in the short term.

Python

```
classification_job.set_limits(
        timeout_minutes=60,
        trial_timeout_minutes=20,
        max_trials=5,
        enable_early_termination=True)
```


77

Submit an AutoML experiment

You can submit an AutoML experiment with the following code:

Python

```
returned_job = ml_client.jobs.create_or_update(classification_job)
```

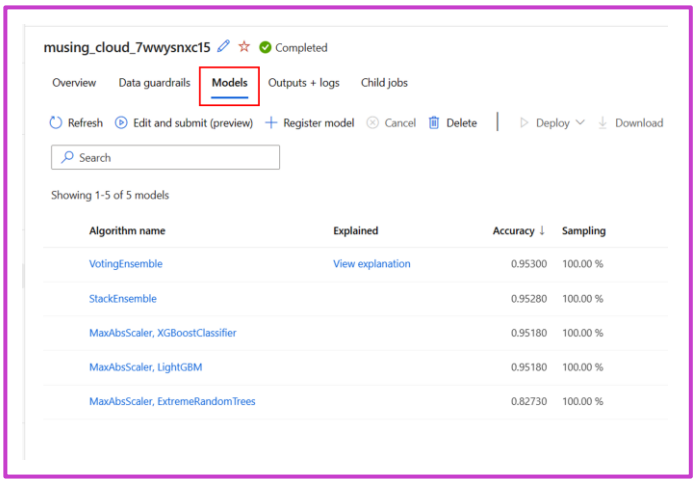
The experiment will consist of child jobs:

- Featurization is performed in a child job.
- Each model is trained in a separate child job.

78

Evaluate and compare models

- In the Azure Machine Learning studio, you can select an AutoML experiment to explore its details.
- On the **Overview** page of the AutoML experiment run, you can review the input data asset and the summary of the best model. To explore all models that have been trained, you can select the **Models** tab:



79

Explore preprocessing steps

When you've enabled featurization for your AutoML experiment, data guardrails will automatically be applied too.

The three data guardrails that are supported for classification models are:

- Class balancing detection
- Missing feature values imputation
- High cardinality feature detection

Each of these data guardrails will show one of three possible states:

- **Passed:** No problems were detected and no action is required
- **Done:** Changes were applied to your data. You should review the changes AutoML has made to your data
- **Alerted:** An issue was detected but couldn't be fixed. You should review the data to fix the issue

80

Retrieve the best run and its model

- When you're reviewing the models in AutoML, you can easily identify the best run based on the primary metric you specified.
- In the **Models** tab of the AutoML experiment, you can **edit the columns** if you want to show other metrics in the same overview.
- To explore a model even further, you can generate explanations for each model that has been trained.

81

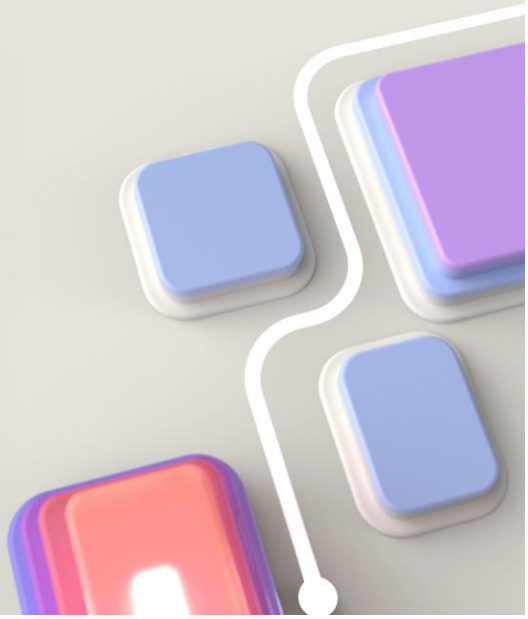
Exercise – Find the best classification model with Automated Machine Learning

In this exercise, you learn how to:

- Prepare the data
- Configure an Automated Machine Learning experiment
- Run an Automated Machine Learning job

82

Track model training in notebooks with MLflow



83

Use MLflow for model tracking in notebooks

You can create and edit notebooks within Azure Machine Learning or on a local device. Notebooks are ideal for exploration and development. To track your work, you can use MLflow.

MLflow is an open-source library for tracking and managing your machine learning experiments. **MLflow Tracking** is a component of MLflow that logs everything about the model you're training, such as **parameters**, **metrics**, and **artifacts**.

MLflow is already configured on Azure Machine Learning compute instances.

84

Configure MLflow for model tracking in notebooks

To use MLflow when running notebooks on your local device:

- Install the `mlflow` and `azureml-mlflow` package.
- Get the value of the MLflow tracking URI from the Azure portal.
- Use the following code in your local notebook to configure MLflow to point to the Azure Machine Learning workspace:

Python

```
mlflow.set_tracking_uri = "MLFLOW-TRACKING-URI"
```


85

Train and track models in notebooks

Create an MLflow experiment:

- You can create a MLflow experiment, which allows you to group runs.
- To create an experiment, run the command on the right in your notebook.

Log results with Mlflow:

- To track the model, you can enable automatic logging and use custom logging.

Python

```
import mlflow
mlflow.set_experiment(experiment_name="heart-condition-classifier")
```


86

Exercise – Track model training in notebooks with MLflow

In this exercise, you learn how to:

- Train and track a model with autologging.
- Train and track a model with custom logging.

87

Microsoft

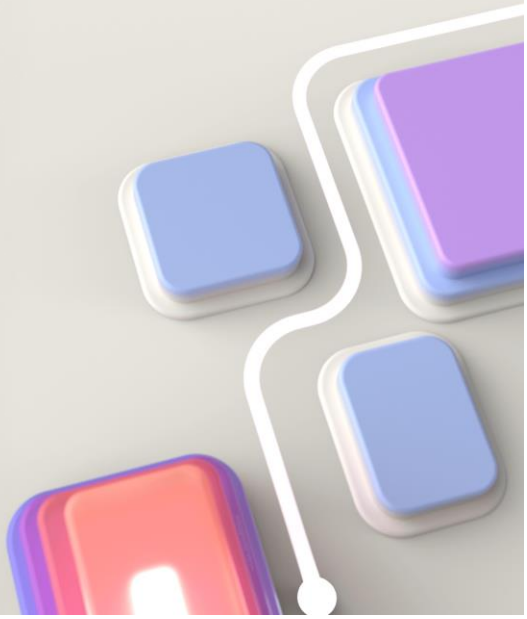
88

Microsoft

Optimize model training with Azure Machine Learning

89

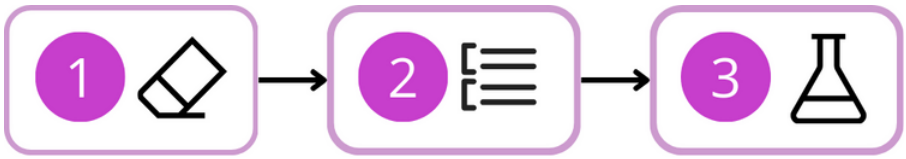
Run a training script as a command job in Azure Machine Learning



90

Convert a notebook to a script

Notebooks are ideal for exploration and development.
Scripts are ideal for testing and automation in your production environment.



To create a production-ready script, you need to:

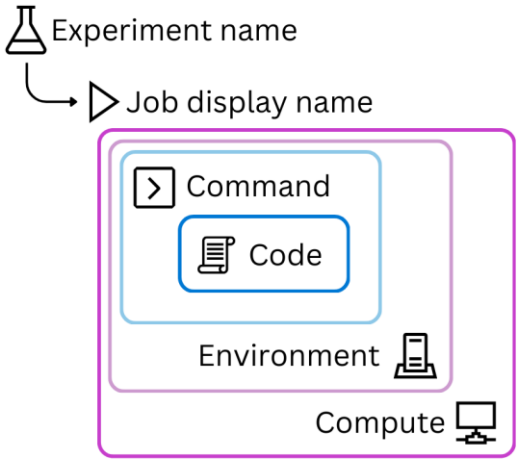
1. Remove nonessential code.
2. Refactor your code into functions.
3. Test your script (in the terminal).

91

Configure a command job

When you have a **script** that you want to execute, you can run it as a **command job**.

Configure the command job by specifying the necessary job parameters and submit the job to the Azure Machine Learning workspace to run the script.



92

Use parameters in a command job

To run one script with different inputs, use **parameters**:

1. Import the **argparse** library in the script.
2. Use the ArgumentParser() method to define arguments for parameters.
3. Specify the parameters in the script, include name, type, and default value.
4. When running the script, specify the value for the defined parameters you want the script to use for this specific run.

```
python train.py --training_data diabetes.csv --reg_rate 0.01“
```


93

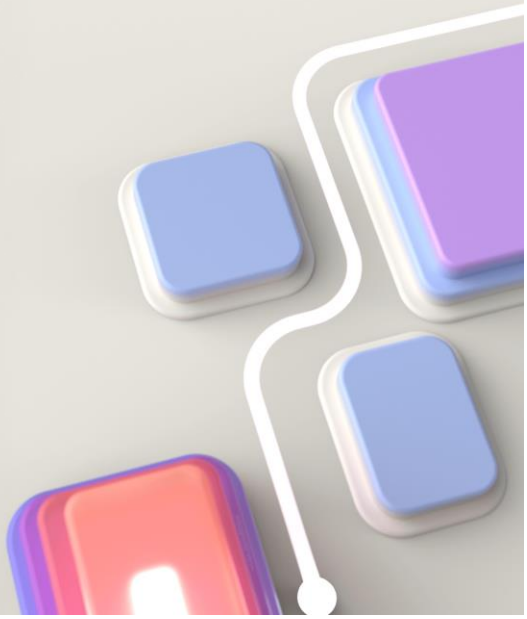
Exercise – Run a training script as a command job in Azure Machine Learning

In this exercise, you learn how to:

- Convert a notebook to a script and test the script in the terminal.
- Run the script as a command job and use parameters when running a script.

94

Track model training with MLflow in jobs



95

Track metrics with MLflow

MLflow is an open-source platform, designed to manage the complete machine learning lifecycle.

There are two options to track machine learning jobs with MLflow:

- Enable autologging using **mlflow.autolog()**
- Use logging functions to track custom metrics using **mlflow.log_***

Include the **mlflow** and **azureml-mlflow** in the environment to ensure the pip packages are installed on the compute before running the script.

96

Enable autologging

When working with one of the common libraries for machine learning, you can enable **autologging** in MLflow.

Autologging logs **parameters, metrics,** and model **artifacts** without anyone needing to specify what needs to be logged.

Python

```
import mlflow
mlflow.autolog()
```


97

Log custom metrics with MLflow

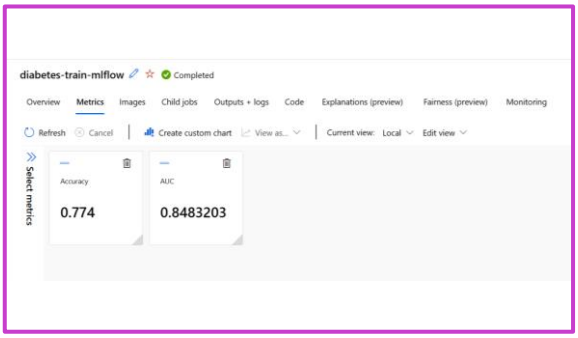
Depending on the type of value you want to log, use the relevant MLflow method to store the parameter, metric, or artifact with the experiment run:

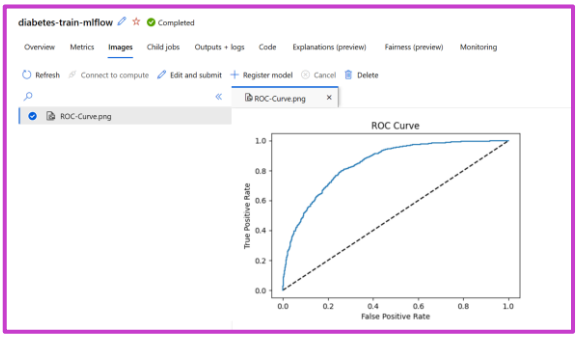
- **mlflow.log_param():** Log single key-value parameter. Use this function for an input parameter you want to log.
- **mlflow.log_metric():** Log single key-value metric. Value must be a number. Use this function for any output you want to store with the run.
- **mlflow.log_artifact():** Log a file. Use this function for any plot you want to log, save as image file first.

98

View the metrics in the Azure Machine Learning studio

- Logged metrics will show in **Overview** and **Metrics** tabs.
- Plots that are logged as artifacts are shown under **Images**.
- Find other artifacts like model files under **Outputs + logs**.





99

Retrieve metrics with MLflow in a notebook

Use MLflow in a notebook connected to the Azure Machine Learning workspace to get more control over which runs you want to retrieve to compare.

List experiments:

```
experiments = mlflow.search_experiments(max_results=2)
for exp in experiments:
        print(exp.name)
```

Retrieve runs:

```
mlflow.search_runs(exp.experiment_id)
```


100

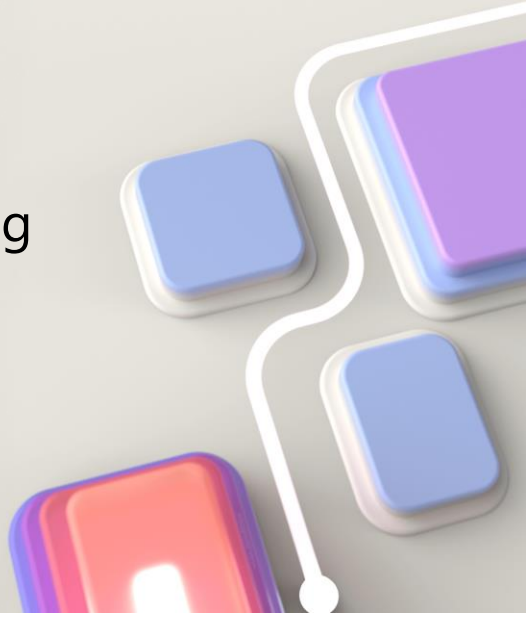
Exercise – Use MLflow to track training jobs

In this exercise, you learn how to:

- Train and track a model with custom logging.
- Train and track a model with autologging.

101

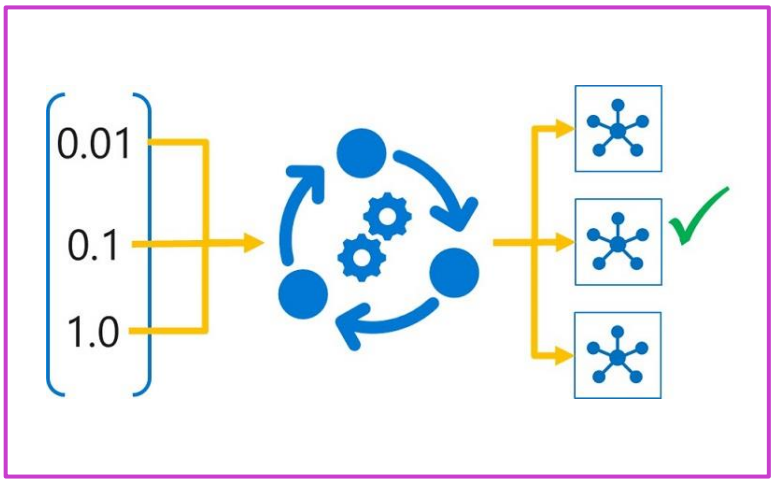
Perform hyperparameter tuning with Azure Machine Learning



102

Understand Hyperparameter Tuning

Hyperparameter tuning is accomplished by training the multiple models, using the same algorithm and training data but different hyperparameter values.



103

Use a sweep job for hyperparameter tuning

In Azure Machine Learning, you can tune hyperparameters by running a sweep job.

1. Create a training script for hyperparameter tuning
2. Configure and run a sweep job
3. Monitor and review sweep (child) jobs

104

Define a search space

The set of hyperparameter values tried during hyperparameter tuning is known as the **search space**. The definition of the range of possible values that can be chosen depends on the type of hyperparameter.

1. **Discrete hyperparameters**: Some hyperparameters require discrete values – In other words, you must select the value from a particular *finite* set of possibilities.

2. **Continuous hyperparameters**: Some hyperparameters are *continuous* – In other words you can use any value along a scale, resulting in an *infinite* number of possibilities

Defining a search space: To define a search space for hyperparameter tuning, create a dictionary with the appropriate parameter expression for each named hyperparameter

105

Configure a sampling method

The specific values used in a hyperparameter tuning run, or **sweep job**, depend on the type of **sampling** used.

There are three main sampling methods available in Azure Machine Learning:

1. **Grid sampling**: Tries every possible combination

2. **Random sampling**: Randomly chooses values from the search space
   - **Sobol**: Adds a seed to random sampling to make the results reproducible

3. **Bayesian sampling**: Chooses new values based on previous results

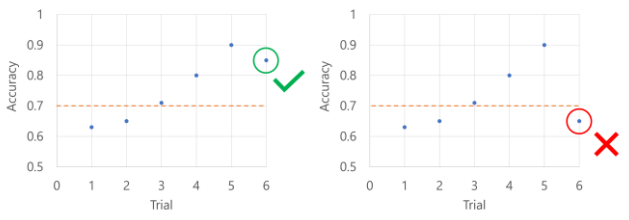
106

Bandit policy

You can use a bandit policy to stop a trial if the target performance metric underperforms the best trial so far by a specified margin.

Python

```
from azure.ai.ml.sweep import
BanditPolicy

sweep_job.early_termination =
BanditPolicy(
    slack_amount = 0.2,
    delay_evaluation = 5,
    evaluation_interval = 1
)
```



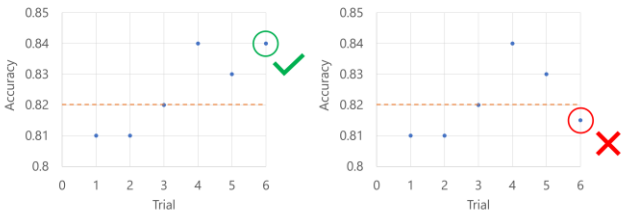
107

Median stopping policy

A median stopping policy abandons trials where the target performance metric is worse than the median of the running averages for all trials.

Python

```
from azure.ai.ml.sweep import
MedianStoppingPolicy

sweep_job.early_termination =
MedianStoppingPolicy(
    delay_evaluation = 5,
    evaluation_interval = 1
)
```



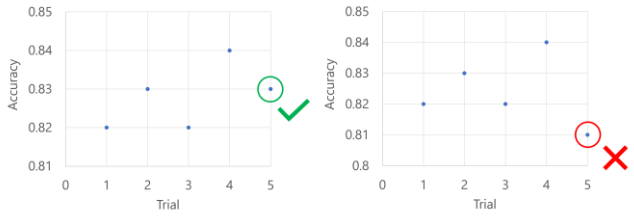
108

Truncation selection policy

A truncation selection policy cancels the lowest performing X% of trials at each evaluation interval based on the *truncation_percentage* value you specify for X.

Python

```
from azure.ai.ml.sweep import
TruncationSelectionPolicy

sweep_job.early_termination =
TruncationSelectionPolicy(
    evaluation_interval=1,
    truncation_percentage=20,
    delay_evaluation=4
)
```



109

Exercise – Perform hyperparameter tuning with a sweep job

In this exercise, you learn how to:

- Run a command job
- Configure a sweep job using the command job as a base
- Submit the sweep job

110

Run pipelines in Azure Machine Learning

111

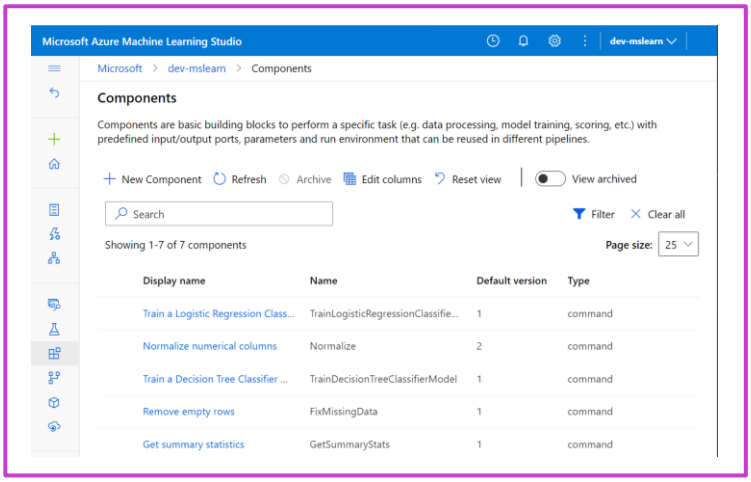
Create components

Components allow you to create reusable scripts that can easily be shared across users within the same Azure Machine Learning workspace.

You can also use components to build an Azure Machine Learning pipeline.

Use a component:

- To build a pipeline
- To share ready-to-go code



112

Create a component

A component consists of three parts:

Metadata

Includes the component's name, version, etc.

Interface

Includes the expected input parameters (like a dataset or hyperparameter) and expected output (like metrics and artifacts)

Command, code and environment

Specifies how to run the code

To create a component, you need two files:

- A script that contains the workflow you want to execute
- A YAML file to define the metadata, interface, and command, code, and environment of the component

Register a component

- To use components in a pipeline, you'll need the script and the YAML file.
- To make the components accessible to other users in the workspace, you can also register components to the Azure Machine Learning workspace.

You can register a component with the following code:

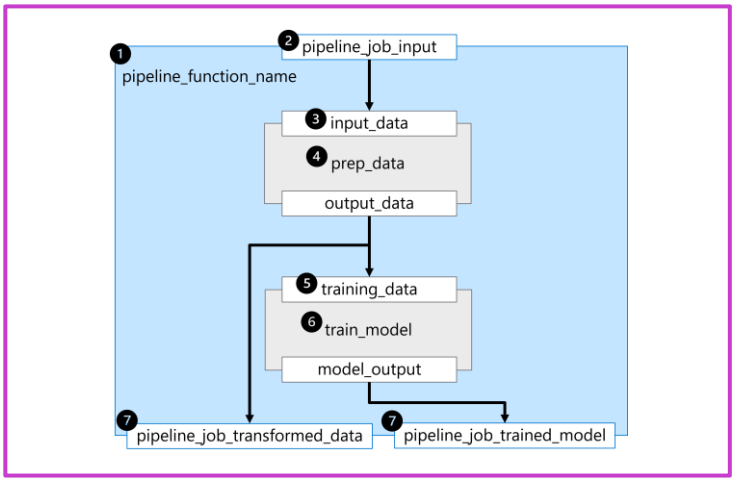
Python

```
prep = ml_client.components.create_or_update(prepare_data_component)
```

Create a pipeline

In Azure Machine Learning, a **pipeline** is a workflow of machine learning tasks in which each task is defined as a **component**.

An Azure Machine Learning pipeline is defined in a YAML file. The YAML file includes the pipeline job name, inputs, outputs, and settings.



115

Run a pipeline job

When you've built a component-based pipeline in Azure Machine Learning, you can run the workflow as a **pipeline job.**

Configure a pipeline job

A pipeline is defined in a YAML file, which you can also create using the `@pipeline()` function

Run a pipeline job

When you've configured the pipeline, you're ready to run the workflow as a pipeline job

Schedule a pipeline job

To schedule a pipeline job, you'll use the `JobSchedule` class to associate a schedule to a pipeline job

116

Exercise – Run pipelines in Azure Machine Learning

In this exercise, you learn how to:

- Create components
- Build a pipeline
- Run a pipeline
- Schedule a pipeline

117

Microsoft

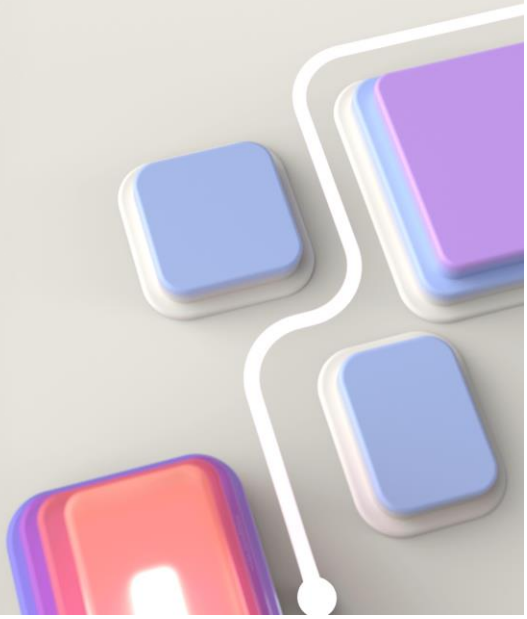
118

Microsoft

Manage and evaluate models in Azure Machine Learning

119

Register an MLflow model in Azure Machine Learning



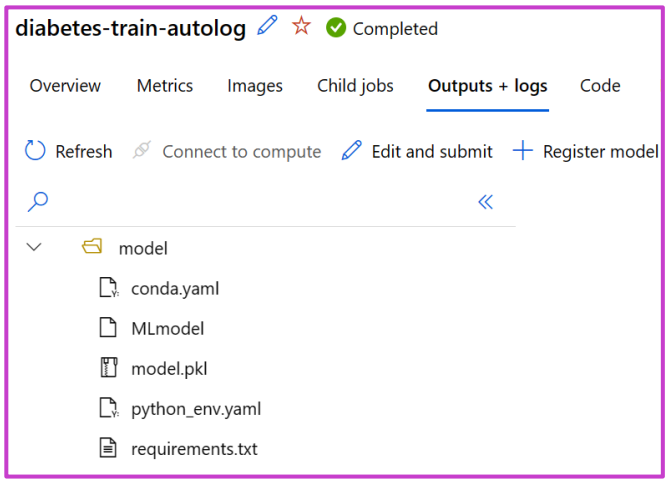
120

Log a model with MLflow

MLflow allows you to log a model as an **artifact**, or as a **model.**

- When you log a model as an artifact, the model is treated as a file.
- When you log a model as a model, you're adding information to the registered model that enables you to use the model directly in pipelines or deployments.

When you log as a model, an MLmodel file is created in the output directory. The MLmodel file contains the model's metadata, which allows for model traceability.



121

Understand the MLmodel file format

The MLmodel file may include:

- **artifact_path:** During the training job, the model is logged to this path.
- **flavor:** The machine learning library with which the model was created.
- **model_uuid:** The unique identifier of the registered model.
- **run_id:** The unique identifier of job run during which the model was created.
- **signature:** Specifies the schema of the model's inputs and outputs:
  - **inputs:** Valid input to the model. For example, a subset of the training dataset.
  - **outputs:** Valid model output. For example, model predictions for the input dataset.

122

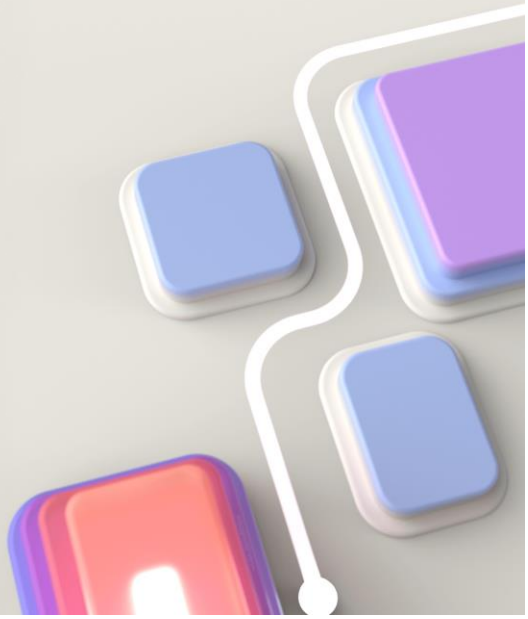
Exercise – Log and register models with MLflow

In this exercise, you learn how to:

- Log models with MLflow.
- Register an MLflow model in the Azure Machine Learning model registry.

123

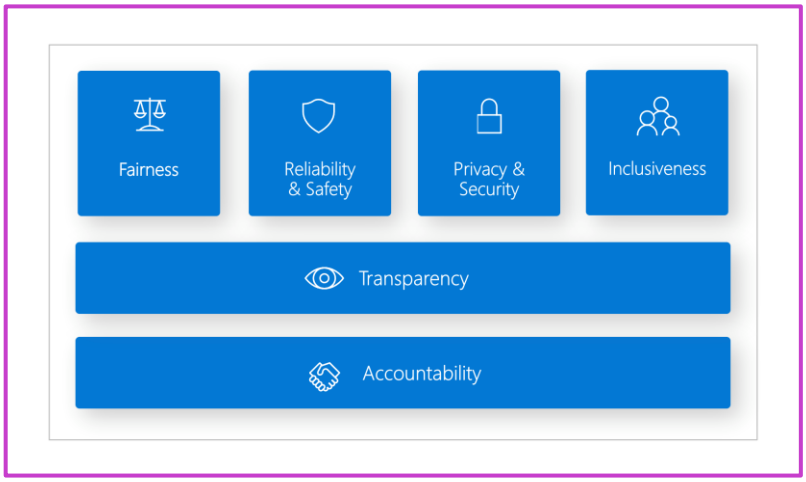
Create and explore the Responsible AI dashboard



124

Understand Responsible Artificial Intelligence (AI)

Responsible Artificial Intelligence (Responsible AI) is an approach to developing, assessing, and deploying AI systems in a safe, trustworthy, and ethical way.



125

Create a Responsible AI dashboard

To create a Responsible AI (RAI) dashboard, you need to create a **pipeline** by using the built-in components. The pipeline should:

1. Start with the RAI Insights dashboard constructor.
2. Include one of the **RAI tool components**.
3. End with Gather RAI Insights dashboard to collect all insights into one dashboard.
4. *Optionally* you can also add the Gather RAI Insights score card at the end of your pipeline

126

Exercise – Create and explore the Responsible AI dashboard

In this exercise, you learn how to:

- Prepare your data and create a Responsible AI dashboard.
- Explore the Responsible AI dashboard.

127

Microsoft

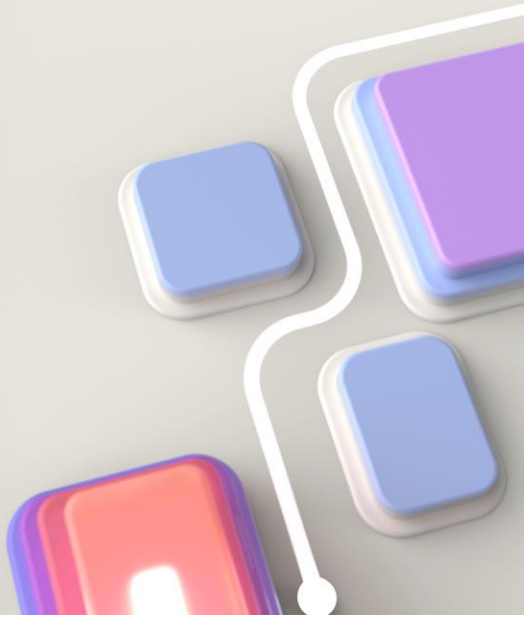
128

Microsoft

Deploy and consume models with Azure Machine Learning

129

Deploy a model to a managed online endpoint



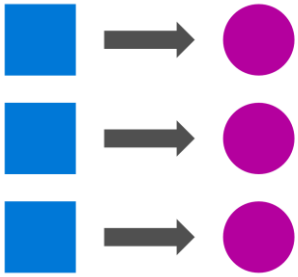
130

Explore managed online endpoints

An **endpoint** is an HTTPS endpoint to which you can send data, and which will return a response (almost) immediately.

Online endpoints are used to generate real-time predictions for individual data points.

With managed online endpoints, Azure Machine Learning manages all the underlying infrastructure.



131

Deploy a model to a managed online endpoint

When you deploy a model to a managed online endpoint, you have two options:

Deploy a MLflow model:

1. Register a MLflow model with a MLModel file.
2. Create deployment.
3. Deploy model to endpoint.

Deploy a (custom) model:

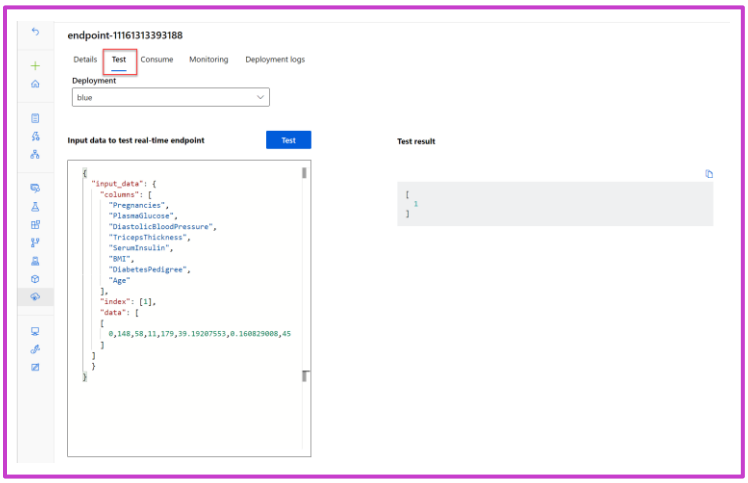
1. Register a model with the necessary model files (subject to model type).
2. Create scoring script.
3. Define execution environment.
4. Create deployment.
5. Deploy model to endpoint.

132

Test managed online endpoints

Use the Azure Machine Learning studio to:

- List all endpoints.
- View an endpoint's details and deployment logs.
- Test the endpoint.



133

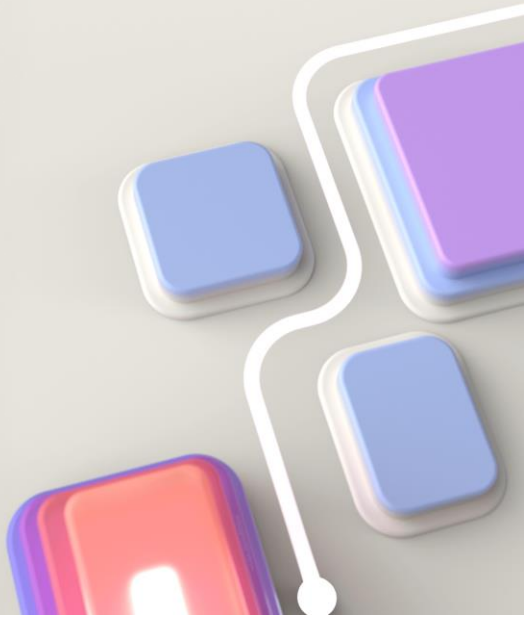
Exercise – Deploy a model to an online endpoint

In this exercise, you learn how to:

- Create a managed online endpoint
- Deploy an MLflow model
- Test the endpoint

134

Deploy a model to a batch endpoint



135

Understand batch endpoints

1. To get batch predictions, you can deploy a model to an endpoint
2. An **endpoint** is an HTTPS endpoint that you can call to trigger a batch scoring job
3. The advantage of such an endpoint is that you can trigger the batch scoring job from another service, such as Azure Synapse Analytics or Azure Databricks
4. Whenever the endpoint is invoked, a batch scoring job is submitted to the Azure Machine Learning workspace

136

Create a batch endpoint

To deploy a model, you first have to create the batch endpoint.

To create a batch endpoint, you'll use the `BatchEndpoint` class. Batch endpoint names need to be unique within an Azure region.

Python

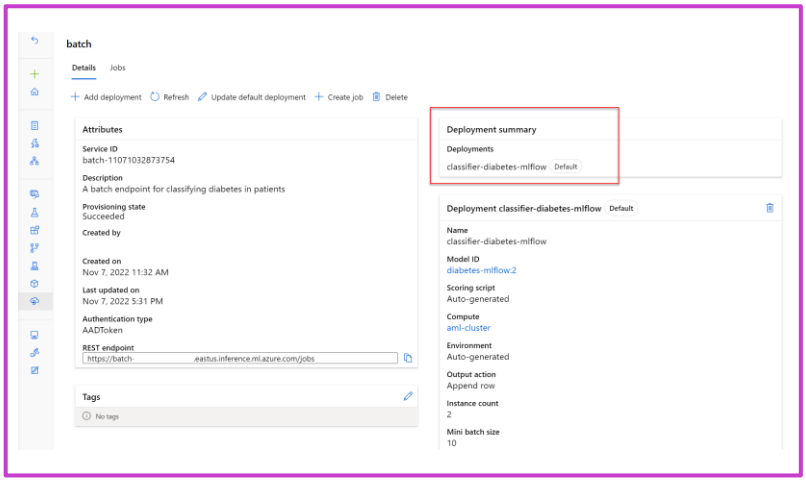
```
# create a batch endpoint
endpoint = BatchEndpoint(
    name="endpoint-example",
    description="A batch endpoint",
)
 ml_client.batch_endpoints.begin_create_or_update(endpoint)
```


137

Deploy a model to a batch endpoint

You can deploy multiple models to a batch endpoint.

Whenever you call the batch endpoint, which triggers a batch scoring job, the **default deployment** will be used unless specified otherwise.



138

Run batch endpoints

Use compute clusters for batch deployments

- To process the new data in parallel batches, you need to provision a compute cluster with more than one maximum instances.
- To create a compute cluster, you can use the `AMLCompute` class.

Python

```
from azure.ai.ml.entities import AmlCompute

cpu_cluster = AmlCompute(
    name="aml-cluster",
    type="amlcompute",
    size="STANDARD_DS11_V2",
    min_instances=0,
    max_instances=4,
    idle_time_before_scale_down=120,
    tier="Dedicated",
)

cpu_cluster =
ml_client.compute.begin_create_or_update(cpu_clu
ster)
```

139

Deploy your MLflow model to a batch endpoint

Deploy an MLflow model to an endpoint

- To deploy an MLflow model to a batch endpoint, you'll use the `BatchDeployment` class.
- When you deploy a model, you need to specify how you want the batch scoring job to behave.
- When you configure the model deployment, you can specify:
  - `instance_count`
  - `max_concurrency_per_instance`
  - `mini_batch_size`
  - `output_action`
  - `output_file_name`

140

Deploy a custom model to a batch endpoint

If you want to deploy a model to a batch endpoint without using the MLflow model format, you need to create the scoring script and environment.

1. **Create the scoring script:** The scoring script is a file that reads the new data, loads the model, and performs the scoring.
2. **Create an environment:** Your deployment requires an execution environment in which to run the scoring script. Any dependency your code requires should be included in the environment.
3. **Configure and create the deployment:** Finally, you can configure and create the deployment with the BatchDeployment class.

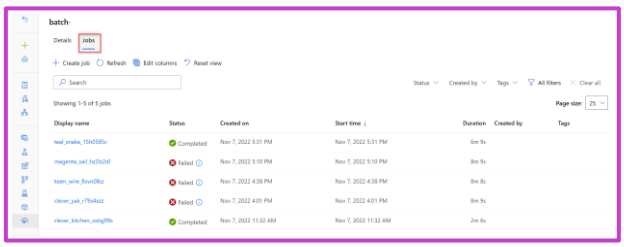
141

Invoke a batch endpoints

- To prepare data for batch predictions, you can register a folder as a data asset in the Azure Machine Learning workspace.
- You can then use the registered data asset as input when invoking the batch endpoint with the Python SDK.

Python

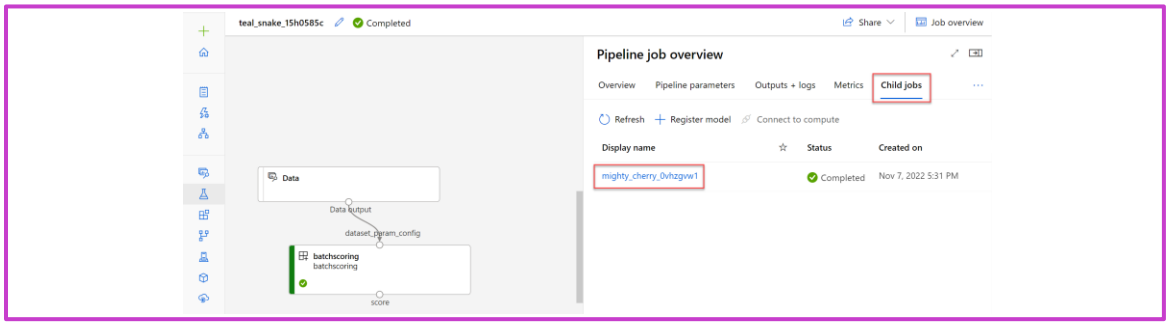
```
from azure.ai.ml import Input
from azure.ai.ml.constants import AssetTypes

input = Input(type=AssetTypes.URI_FOLDER,
path="azureml:new-data:1")

job = ml_client.batch_endpoints.invoke(
    endpoint_name=endpoint.name,
    input=input)
```



142

Troubleshoot a batch scoring job

The batch scoring job runs as a **pipeline job**. If you want to troubleshoot the pipeline job, you can review its details and the outputs and logs of the pipeline job itself.



143

Exercise – Deploy a model to a batch endpoint

In this exercise, you learn how to:

- Create a batch endpoint
- Deploy an MLflow model to the endpoint
- Invoke the endpoint

144

145