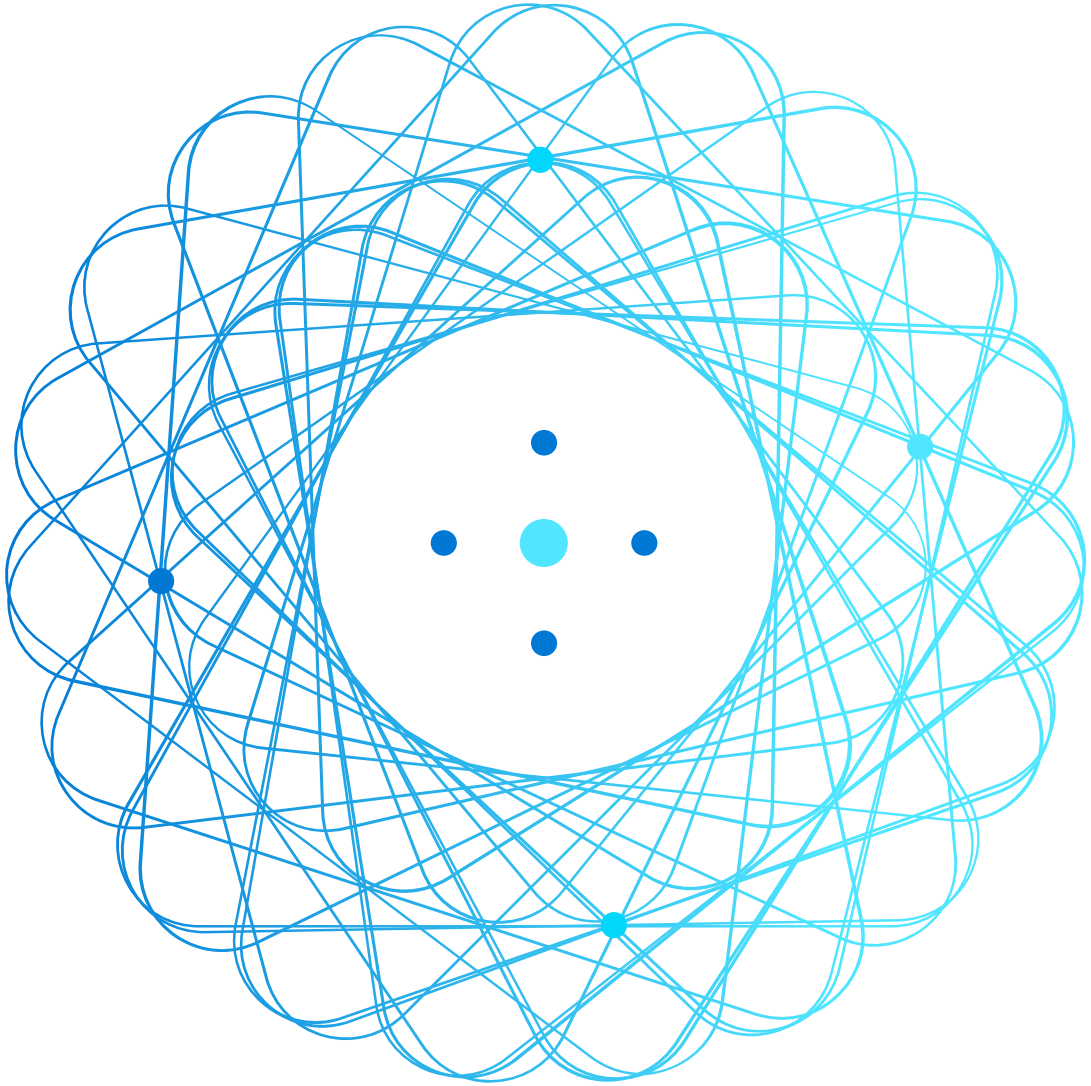
Microsoft Azure

# Train models with scripts in Azure Machine Learning



# Module Agenda

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

Track model training with MLflow in jobs

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

# 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'll need to:

- Remove nonessential code.
- Refactor your code into functions.
- Test your script (in the terminal).

# Configure a command job (1/2)

To configure a command job to run a script, you'll need to specify values for the following parameters:

- **code**: The folder that includes the script to run.
- **command**: Specifies which file to run.
- **environment**: The necessary packages to be installed on the compute before running the command.
- **compute**: The compute to use to run the command.
- **display_name**: The name of the individual job.
- **experiment_name**: The name of the experiment the job belongs to.

# Configure a command job (2/2)

To configure a command job with the Python SDK (v2), use the **command** function:

Python

```
from azure.ai.ml import command


job = command(
            code="./src",
            command="python train.py",
            environment="AzureML-sklearn-0.24-ubuntu18.04-py37-cpu@latest",
            compute="aml-cluster",
            display_name="train-model",
            experiment_name="train-classification-model")
returned_job = ml_client.create_or_update(job)
```

# Use parameters in a command job

### Define parameters in the script:

To use parameters in a script, you must use a library such as **argparse** to read arguments passed to the script and assign them to variables.

### Set the parameter values in the command job:

To pass parameter values to a script, you need to provide the argument value in the **command**.

Python

```
from azure.ai.ml import command


job = command(
            code="./src",
            command="python train.py
            --training_data diabetes.csv",
            environment="AzureML-sklearn-
            0.24-ubuntu18.04-py37-cpu@latest",
            compute="aml-cluster",
            display_name="train-model",
            experiment_name="train-
            classification-model")
```

# Exercise – Run a training script as a command job

## In this exercise, you will:

**Task 1:**

Convert a notebook to a script and test the script in the terminal.

**Task 2:**

Run the script as a command job and use parameters when running a script.

## Instructions

Follow these instructions to complete the exercise:

1.View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.

2.Complete the **Run a training script as a command job in Azure Machine Learning** exercise.

# Knowledge check

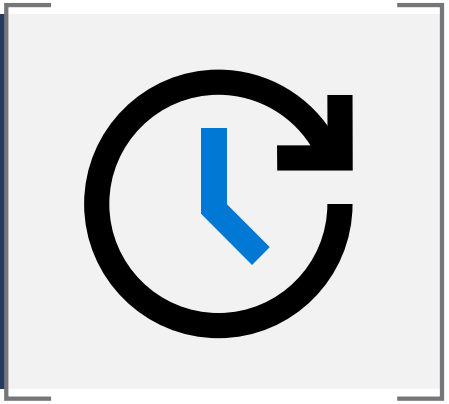
A data scientist wants to run a script as a command job to train a PyTorch model, setting the batch size and learning rate hyperparameters to specified values each time the job runs. What should be done by the data scientist?

- ❑ 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 command job before submitting the job.
- ☑ Add arguments for batch size and learning rate to the script and set them in the command property of the command job.

A data scientist has trained a model in a notebook. The model should be retrained every week on new data. What should the data scientist do to make the code production-ready?

- ❑ Copy and paste the code from each cell to a script.
- ❑ Convert the code to one function in a script that reads the data and trains the model.
- ☑ Convert the code to multiple functions in a script that read the data and train the model.

# Track model training with MLflow in jobs



# 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.

# 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()
```

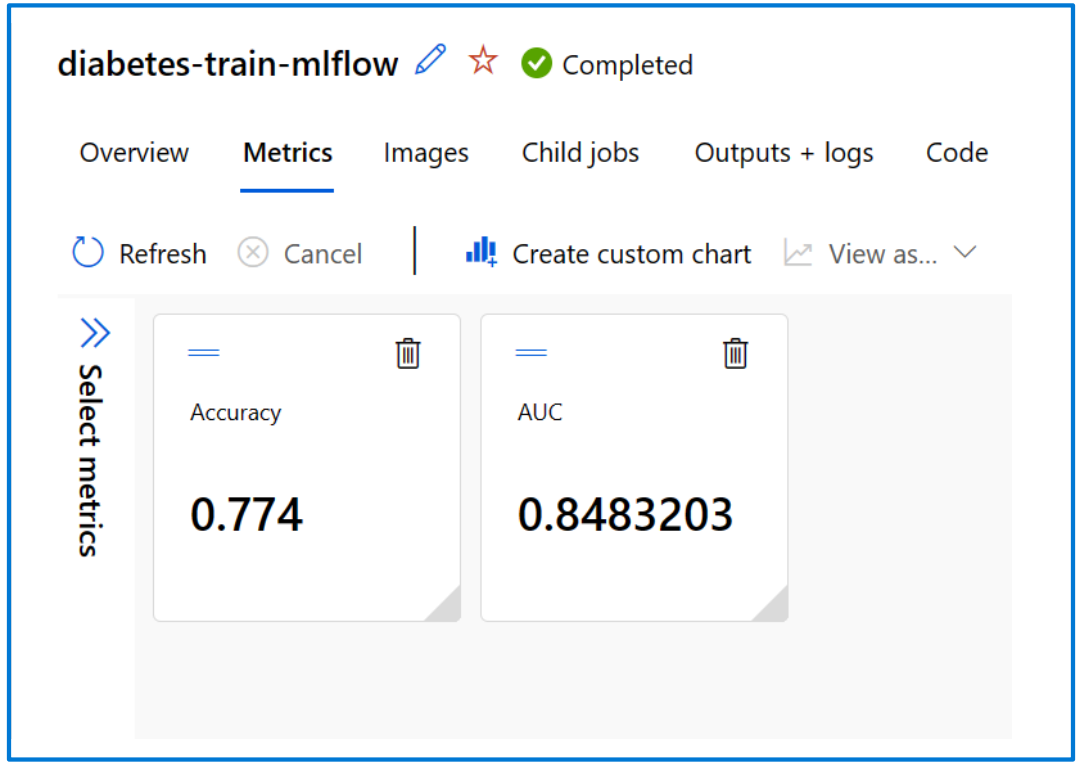
# Log custom metrics with MLflow

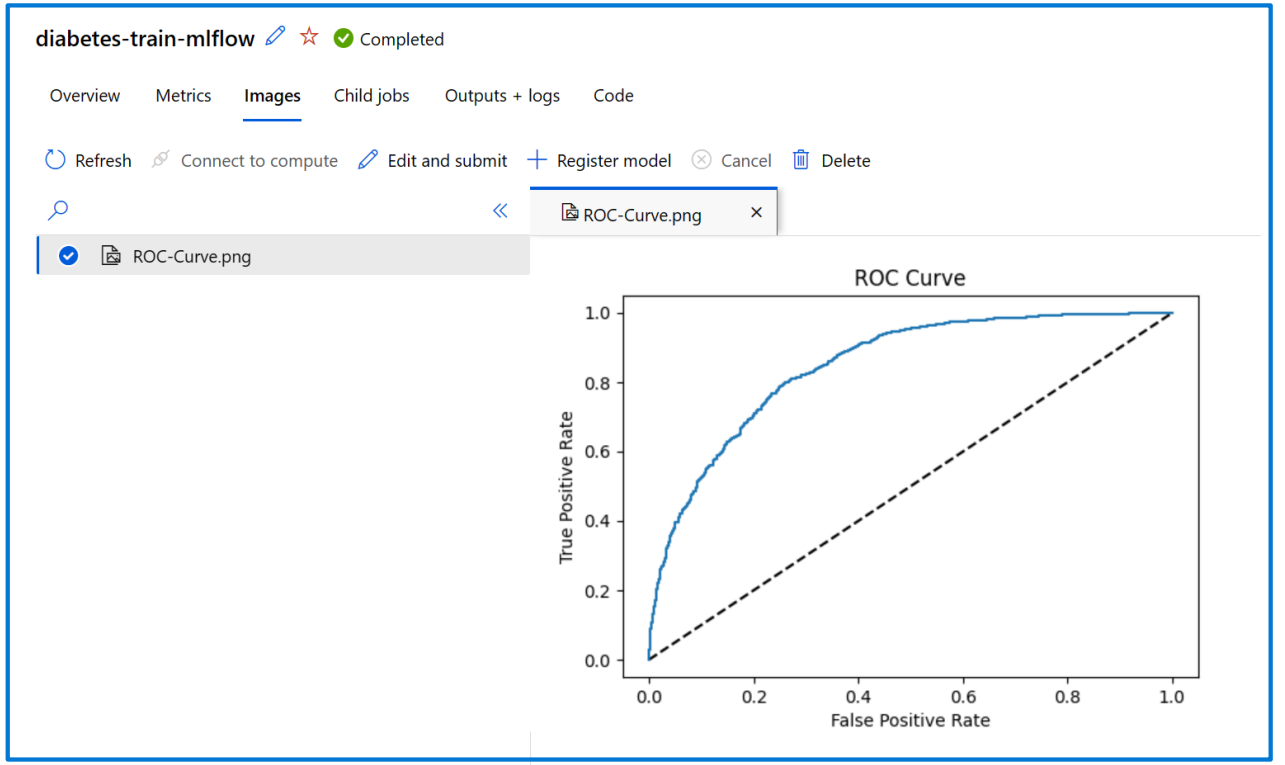
Depending on the type of value you want to log, use the MLflow command to store the metric with the experiment run:

- **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.

# 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**.



# Retrieve metrics with MLflow in a notebook

Use MLflow in a notebook to get more control over which runs you want to retrieve to compare.

- List experiments:

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

- Retrieve runs:

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

# Exercise – Use MLflow to track training jobs

## In this exercise, you will:

**Task 1:**

Train and track a model with custom logging.

**Task 2:**

Train and track a model with autologging.

## Instructions

Follow these instructions to complete the exercise:

1.View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.

2.Complete the **Use MLflow to track training jobs** exercise.

# Knowledge check

A data scientist trains a regression model and wants to track the model's performance by storing the Root Mean Squared Error (RMSE) with the experiment run. Which method can be used to log the RMSE?

- ❑ mlflow.log_param()
- ❑ mlflow.log_artifact()
- ☑ mlflow.log_metric()

When a data scientist enables MLflow autologging, where can all model assets be found?

- ☑ In the model folder under Outputs + logs.
- ❑ In the outputs folder under Outputs + logs.
- ❑ In the model folder under Metrics.

Microsoft Azure