[https://docs.microsoft.com/en-us/azure/machine-learning/concept-azure-machine-learning-architecture#glossary](https://docs.microsoft.com/en-us/azure/machine-learning/concept-azure-machine-learning-architecture" \l "glossary)

寿命预测

原件的使用寿命，预测性维护，损坏趋势？

<https://docs.microsoft.com/en-us/azure/machine-learning/>

Azure machine learning

寿命预测在Azure上怎么做

机器学习去做？machine learning

1. 有监督 --有失败数据

做回归。。

1. 无监督 ---无或者少量失败数据

Azure machine learning services

**What** are the machine learning products at MSoft

Machine learning is data science tech that allows computers to use exiting data to forecast future behaviors, outcomes, and trends. By using machine learning, computers learn without being explicitly programmed.

Machine learning solutions are built iteratively, and have distinct phases:

1. Preparing data
2. Experimenting and training models
3. Deploying trained models
4. Managing deployed models

Microsoft provides a variety of product options to prep, build, deploy, and manage your machine learning models. Compare these products and choose what you need to develop your machine learning solutions most effectively.

Options:

Cloud-based

| **Cloud options** | **What it is** | **What you can do with it** |
| --- | --- | --- |
| [Azure Machine Learning](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "azure-machine-learning) | Managed cloud service for machine learning | Train, deploy, and manage models in Azure using Python and CLI |
| [Azure Machine Learning Studio (classic)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "classic) | Drag–and–drop visual interface for machine learning | Build, experiment, and deploy models using preconfigured algorithms |

On-premises

| **On-premises options** | **What it is** | **What you can do with it** |
| --- | --- | --- |
| [SQL Server Machine Learning Services](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "sql-server-machine-learning-services) | Analytics engine embedded in SQL | Build and deploy models inside SQL Server |
| [Microsoft Machine Learning Server](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "microsoft-machine-learning-server) | Standalone enterprise server for predictive analysis | Build and deploy models on pre-processed data |

Development platforms and tools

| **Platforms/tools** | **What it is** | **What you can do with it** |
| --- | --- | --- |
| [Azure Data Science Virtual Machine](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "azure-data-science-virtual-machine) | Virtual machine with pre-installed data science tools | Develop machine learning solutions in a pre-configured environment |
| [Azure Databricks](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "azure-databricks) | Spark-based analytics platform | Build and deploy models and data workflows |
| [ML.NET](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "mlnet) | Open-source, cross-platform machine learning SDK | Develop machine learning solutions for .NET applications |
| [Windows ML](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "windows-ml) | Windows 10 machine learning platform | Evaluate trained models on a Windows 10 device |
| [MMLSpark](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-science-and-machine-learning?toc=/en-us/azure/machine-learning/toc.json&bc=/en-us/azure/bread/toc.json" \l "mmlspark) | Open-source, distributed, machine learning and microservice framework for Apache Spark | Create and deploy scalable machine learning applications for Scala and Python. |

## **Azure Machine Learning**

[Azure Machine Learning](https://docs.microsoft.com/en-us/azure/machine-learning/service/overview-what-is-azure-ml) is a fully managed cloud service used to train, deploy, and manage machine learning models at scale. It fully supports open-source technologies, so you can use tens of thousands of open-source Python packages such as TensorFlow, PyTorch, and scikit-learn.

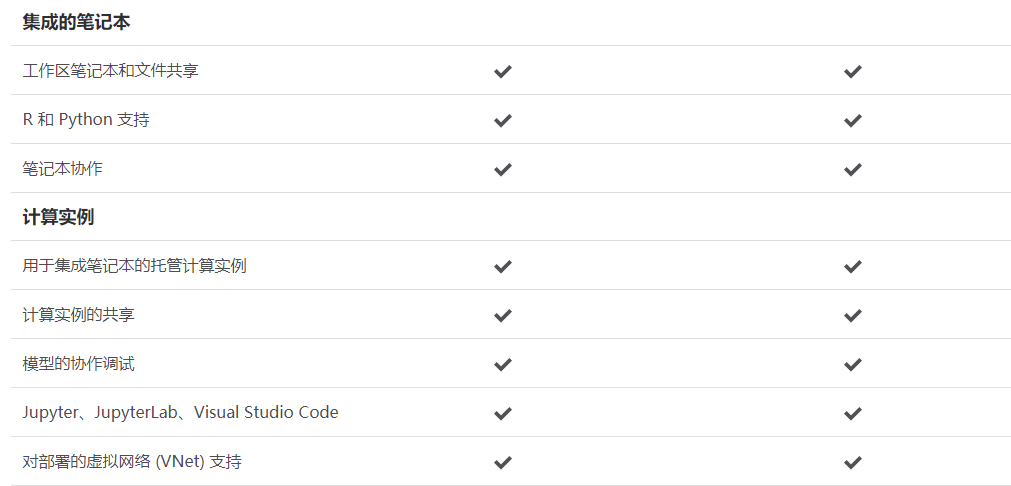
|  |  |
| --- | --- |
| **Type** | Cloud-based machine learning solution |
| **Supported languages** | Python |
| **Machine learning phases** | Data preparation Model training Deployment Management |
| **Key benefits** | Central management of scripts and run history, making it easy to compare model versions.  Easy deployment and management of models to the cloud or edge devices. |
| **Considerations** | Requires some familiarity with the model management model. |

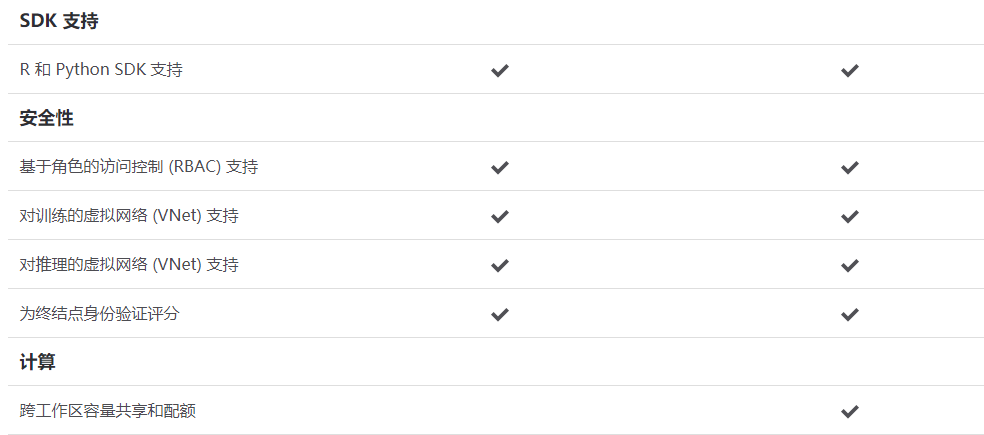
## **Azure ML Studio (Classic)**

|  |  |
| --- | --- |
| **Type** | Cloud-based, drag-and-drop machine learning solution |
| **Supported languages** | Python, R |
| **Machine learning phases** | Data preparation Model training Deployment Management |
| **Key benefits** | Interactive visual interface enables machine learning modeling with minimal code.  Built-in Jupyter Notebooks for data exploration.  Direct deployment of trained models as Azure web services. |
| **Considerations** | Limited scalability. The maximum size of a training dataset is 10 GB.  Online only. No offline development environment. |

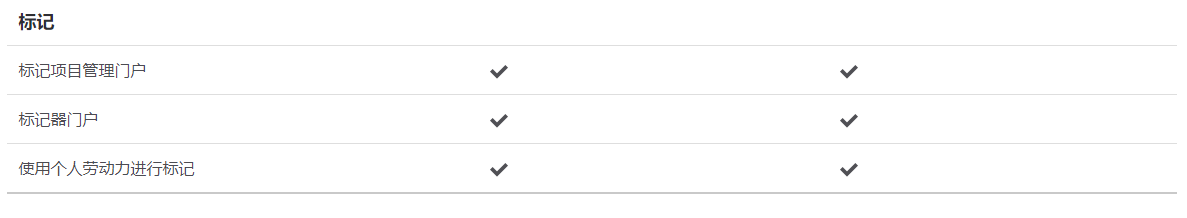




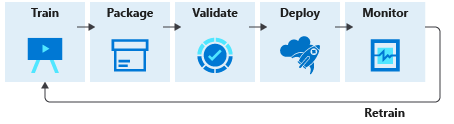








**How** Azure Machine Learning works:Architecture and concepts



1. Train
   1. Develop machine learning training scripts in Python or with the visual designer
   2. Create and configure a computer target
   3. Submit the scripts to the configured compute target to run in that environment. During training, the scripts can read from or write to datastore. And the records of execution are saved as runs in the workspace and grouped under experiments.
2. Package
   1. After a satisfactory run is found, register the persisted model in the model registry.
3. Validate -Query the experiment for logged metrics from the current and past runs. If the metrics don't indicate a desired outcome, loop back to step 1 and iterate on your scripts.
4. Deploy
   1. Develop a scoring script that uses the model and Deploy the model as a web service in Azure,or to an IOT Edge device.
5. Monitor

Monitor for data drift between the training dataset and inference data of deployed model. When necessary, loop back to step 1 to retrain the model with new training data.

Glossary

Activities

Workspace

The workspace is the top-level resource for Azure Machine learning. It provides a centralized place to work with all the artifacts you create when you use AML.

Experiments

An experiment is a grouping of many runs from a specified script.

Runs

A run is a single execution of a training script.

Run configurations

A run configuration is a set of instructions that defines how a script should be run in a specified compute target.

Snapshot

When you submit a run, Azure Machine Learning compresses the directory that contains the script as a zip file and sends it to the compute target.

Logging

ML Pipelines

You use machine learning pipelines to create and manage workflows that stitch together machine learning phases.

For example, a pipeline might include data preparation, model training, model deployment, and inference/scoring phases.

Models

Environment

Training Scripts

Estimators

To facilitate model training with popular frameworks, the estimator class allows you to easily construct run configurations. You can create and use a generic Estimator to submit training scripts that use any learning framework you choose (such as scikit-learn).

Endpoints

Compute instance

Datasets and datastores

Compute targets

A compute target lets you specify the compute resource where you run your training script or host your service deployment. This location may be your local machine or a cloud-based compute resource

Azure Machine Learning documentation

Azure machine learning enterprise

Get started with Azure ML

Experiment: Python SDK

Use ML Pipelines

Deploy & manage models

Azure machine learning workspaces

AML workspaces

How to create



AML HOW TO - Guides

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | create | 3/3 | Use Azure portal  Azure CLI  REST  Res M template |  |  |
| 2 | Set up your environment | 3/3 | Development environment | AML compute instance  Local environment  Azure Databricks  DSVM(data science VM) | 我创建了一个AML  Compute instance  ，它所有配置都有了。 |
| Software environment |  |  |
| 3 | Work with data | 3/3 |  |  |  |
| 4 | Train models | 3/3 |  |  |  |
| 5 | Interpret models | 3/4 |  |  |  |
| 6 | Automate machine learning | 3/4 |  |  |  |
| 7 | Track & monitor experiments | 3/4 |  |  |  |
| 8 | Deploy & serve models | 3/4 |  |  |  |
| 9 | Build & use ML pipelines | 3/5 |  |  |  |
| 10 | Manage resource quotas | 3/5 |  |  |  |
| 11 | Export and delete data | 3/5 |  |  |  |
| 12 | Create event driven workflow | 3/5 |  |  |  |

create

Set up your environment

Work with data

Train models

Interpret models

Automate machine learning

Track & monitor experiments

Deploy & serve models

Build & use ML pipelines

Manage resource quotas

Export and delete data

Create event driven workflow

Tutorial: Get started creating your first ML experiment with the Python SDK

1. Set up workspace & dev environment

In this tutorial, you :

Create an AML workspace to use in the next tutorial

Create the tutorial notebook to your folder in the workspace

Create a cloud-based compute instance wit AML Python SDK installed and pre-configured.

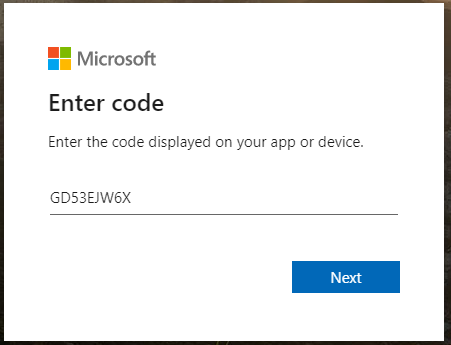
1. Train your first model

Connect your workspace and create an experiment

Load data and train scikit-learn models

View training results in the portal

Retrieve the best model



Connect workspace and create experiment

Import the Workspace class, and load your subscription information from the file config.json using the function from\_config(). This looks for the JSON file in the current directory by default, but you can also specify a path parameter to point to the file using from\_config(path="your/file/path"). If you are running this notebook in a cloud notebook server in your workspace, the file is automatically in the root directory.

If the following code asks for additional authentication, simply paste the link in a browser and enter the authentication token.

from azureml.core import Workspace

ws = Workspace.from\_config()

Performing interactive authentication. Please follow the instructions on the terminal.

To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code GD53EJW6X to authenticate.

Example :

<https://github.com/Azure/AI-PredictiveMaintenance>

What is AI-PredictiveMaintenance

Is an application of predictive analytics

The data requirements and modeling techniques to build PdM solutions are also provided

The main content of the guide is on the data science process -including the steps of data preparation, feature engineering, model creation, and model operationalization

## **Data Science for predictive maintenance**

This section provides general guidelines of data science principles and practice for PdM.

Data requirements for predictive maintenance

Relevant Data

Sufficient data

Quality data

Data preparation for predictive maintenance

Data resource

The relevant data resources for predictive maintenance include, but are not limited to:

Failure History

Maintenance/repair history

Machine operating conditions

Equipment metadata

Data types

Temporal data

Static data

Data preprocessing

As a prerequisite to feature engineering, prepare the data from various streams to it is easy to build features.

Feature engineering

Feature engineering is the first step prior to modeling the data.

A feature is a predictive attribute for the model - such as temperature, pressure,vibration, and so on.

For PdM, feature engineering involves abstracting a machine’s health over historical data collected over a sizable duration.

Time windows

Lag features //滞后特性

Rolling aggregates //轧机设备

Tumbling aggregates

Static features

## **Modeling techniques for predictive maintenance**

Binary classification

Binary classification is used to predict the probability that a piece of equipment fails within a future time period

Label construction for binary classification

Regression for predictive maintenance

Label construction for regression

Multi-class classification for predictive maintenance

Label construction for multi-class classification

Training, validation, and testing methods for predictive maintenance

Cross validation

Testing for model performance

Time-dependent split

Handling imbalanced data

Model evaluation

Accuracy

Precision

Recall

F1 score

Model operationalization for predictive maintenance

## **Solution templates for predictive maintenance**

The final section of this guide provides a list of solution template, tutorials ,and experiments implemented in Azure.

They can be used as proof-of-concept demos, sandboxes to experiment with alternatives, or accelerators for actual production implementations. These templates are located in the Azure AI Gallery or Azure GitHub.

## **Solution templates for predictive maintenance**

How

DataGeneration

DataIngestion

FeatureEngineering

In machine learning, a feature is an individual measurable attribute of a phenomenon being observed. The extraction of feature from raw training data is called feature engineering. In other words, freature

Engineering is a process of transforming raw training data into

A representation suitable for the application of machine learning algorithm.

This process usually requires a certain degree of domain expertise and can be divided into the following stages.

Brainstorming on features;

Deciding what features to create

Creating features

Studying how the features impact model’s predicitive accuracy

Iterating if necessary

ModelTraining

Get Started with AML

Set up training environment

Set up and use compute targets for model training

A compute target can be a local machine or a cloud resource.

The steps for all compute targets follow the same workflow:

1. Create
2. Attach
3. Configure

Compute targets for training

Compute targets can be reused from one training job to the next.

For machine learning pipelines, use the appropriate pipelines step for each compute target.

|  | | | |
| --- | --- | --- | --- |
| **Training  targets** | **[Automated ML](https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml)** | **[ML pipelines](https://docs.microsoft.com/en-us/azure/machine-learning/concept-ml-pipelines)** | **[Azure Machine Learning designer](https://docs.microsoft.com/en-us/azure/machine-learning/concept-designer)** |
| [Local computer](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets" \l "local) | yes |  |  |
| [Azure Machine Learning compute cluster](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets" \l "amlcompute) | yes & hyperparameter tuning | yes | yes |
| [Remote VM](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets" \l "vm) | yes & hyperparameter tuning | yes |  |
| [Azure Databricks](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-your-first-pipeline" \l "databricks) | yes (SDK local mode only) | yes |  |
| [Azure Data Lake Analytics](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-your-first-pipeline" \l "adla) |  | yes |  |
| [Azure HDInsight](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets" \l "hdinsight) |  | yes |  |
| [Azure Batch](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets" \l "azbatch) |  | yes |  |

What’s a run configuration

When training, it is common to start on your local computer, and later run that training script on a different compute target. With Azure Machine Learning, you can run your script on various compute targets without having to change your script.

## **What's an estimator**

## **What's an ML Pipeline?**

Tutorial tutorial-1st-experiment-sdk-train.ipynb

Connected to your workspace and created an experiment

Loaded data and trained scikit-learn models

Viewed

Connect to workspace

Ws = workspace.from\_config()

Create experiment

Exp = Experiment(workspace=ws, name = experiment\_name)

Create or Attach existing compute resource.

compute\_target = ComputeTarget.create

compute\_target = ws.compute\_targets[compute\_name]

Explore data

Download the MNIST data

Display some sample images

Train on a remote cluster

Create a directory

Create a training script

Create an estimator object

Submit the job

Monitor a remote run

Register model

The last step in the training script wrote the file outputs/sk...pkl in a directory named outputs in the VM of the cluster where the is run.

Outputs is a special directory in that all content in this directory is automatically uploaded to your workspace.

# **Tutorial: Deploy an image classification model in Azure Container Instances**

1. Set up your testing environment
2. Retrieve the model from your workspace
3. Test the model locally
4. Deploy the model to Container Instances.
5. Test the deployed model.

Retrieve the model

You registered a model in your WK in the previous tutorial.Now, load this WK and download the model to your local directory.

Model = model(ws, ‘sklearn\_mnist’)

Model.download(target\_dir=os.gecwd, exit\_ok=true)

Test model locally

Downloading the test data

Loading test data

Predicting test data

Examining the confusion matrix

Predict test data

Feed the test dataset to the model to get predictions

Import joblib

Clf = joblib.load(..,...pkl)

Y\_hat = clf.preditct(X-test)

Print(y\_hat)

Examine the confusion matrix

From sklearn.metrics import confusion\_matrix

Conf\_mx = confusion\_matrix(y\_test, yhat)

Print(conf\_mx)

Deploy as web service

To build the correct environment for ACI, provide the following:

A scoring script how to use the model

An environment file to show what packages need to be installed.

A configuration file to build the ACI.

The model you trained before.

Create scoring script

Y\_hat = model.predict(data)

Return y\_hat.tolist()

Create environment file

Create configuration file

Aiconfig = AciWebservice.deploy\_configuration(cpu\_cores =1,

Memory\_gb=1,

Tags = [data:MNIST, method:sklearn])

Description = predict )

Deploy in ACI

Service = Model.deploy (workspace = WK,

Name = xx,

Models=xx,

Inference\_config = xxx,

Deployment\_config = xxx)

Service.wait\_for\_deployment(show\_output= True)

Get the scoring web service’s HTTP endpoint, which accepts REST client calls.

Print(service.scoring\_uri)

[http://3cc18b2d-eec4-4a60-b5b5-efbb76b2423f.eastasia.azurecontainer.io/score](http://3cc18b2d-eec4-4a60-b5b5-efbb76b2423f.eastasia.azurecontainer.io/score" \t "https://sl-cpt01.eastasia.instances.azureml.net/tree/users/wenbo.he/tutorials/image-classification-mnist-data/_blank)

Test deployed service

1. Send the data as a JSON array to the web service hosted in ACI.
2. Use the SDK’s a `run` API to invoke the service.
3. Print the returned predictions and plot them along with the input images. Red font and inverse image

Import requests

Resp = request.post(service.scoring\_uri,input\_data,headers =

headers)

# **Tutorial: Use automated machine learning to predict taxi fares**

In this tutorial you learn the following tasks:

Download, transform, and clean data using Azure Open Datasets

Train an automated machine learning regression model.

Calculate model accuracy

Download and prepare data

Automatically train a model

1. Define settings for the experiment run.Attach your training data to the configuration, and modify settings that control the training process.
2. Submit the experiment for model tuning.

After submitting the experiment, the process iterates through different machine learning algorithms and hyperparameter setting, adhering your defined constraints. It choose the best-fit model by optimizing an accuracy metric

# **Tutorial: Build an Azure Machine Learning pipeline for batch scoring**

Create&manage workspaces

Set up your environment

Work with data

Train models

Interpret models

Automate machine learning

# **Access data in Azure storage services**

How to easily access your data in Azure Storage services via AML datastorages.

Datastorages are used to store connection information, like your subscription ID and token authorization.

When you use datastores, you can access your stroage without having to hard code connection information in your scripts.

**Supported data storage service types**

...

**Create and register datastores**

Via python or AML studio

Blob container

File share

Azure data lake storage Generation2

**Get datastores from you workspace**

Datastore = Datastore.get(ws, datastore\_name=’’)

**Upload and download data**

Datastore.upload(src=’’,

Target\_path=’’,

Overwrite=True,

Show\_process=True)

**Download data from a datastore to your local filesystem:**

Datastore.download(target\_path=’’,

Prefix=’’,

Show\_process=True)

**Access your data during training**

Azure blob storage has higher throughput speeds than an Azure file

Share and will scale to large numbers of jobs started in parallel.

For this reason, we recommend configuring your runs to use Blob storage for transferring source code files.

The following code example specifies in the run configuration which blob datastore to use for source code transfers;

Run\_config.source\_directory\_data\_store = ‘workspaceblobstorage’

Access data during scoring

Create AML datasets

Dataset Type：

tabular dataset

Ws = ws.from\_config()

Datastore = datastore.get(ws, datastore\_name)

Datastore\_paths = [(datastore, ‘ather/2018/11.csv’),

...]

Weather\_ds =dataset.Tabular.from\_delimited\_files(path=datastore\_paths)

Use the from\_sql\_query() method on the TabularDatafactory class to read from Azure SQL Database

Sql\_datastore = datastore.get(ws, ‘mysql’)

Sql\_ds = dataset.Tabular.from\_sql\_query((sql\_datastore, ‘SELECT \* FROM my\_table’))

File dataset

TBD

Register datasets

TBD

# **Train with datasets in Azure Machine Learning**

AML datasets provides a seamless integration with AML training products

Like ScriptRun, Estimator, HyperDrive and AML pipelines.

Use datasets directly in training scripts

1. Create a TabularDataset
   1. Web\_path = ‘xxx.csv’
   2. Titanic\_ds = dataset.Tabular.from\_delimited\_files(path=web\_path)
2. Access the input dataset in your training script
   1. Run = Run.get\_context()
   2. Dataset = run.input\_dataset[‘titanic\_ds’]
   3. Df = dataset.to\_pandas\_dataframe()
3. Configure the estimator

An estimator object is used to submit the experiment run.

* 1. Est = Estimator(src\_ditrectory = script\_folder,

Entry\_script = train\_titanic.py

Input= [titanic\_ds.as\_named\_input(‘titanic’)]

Compute\_target = compute\_target,

Environment\_definition = conda\_env)

* 1. Experiment\_run = experiment.submit(est)
  2. ...waitforcompletion

Detect data drift(preview)on datasets

With Azure Machine Learning dataset monitors, you can:

* ****Analyze drift in your data**** to understand how it changes over time.
* ****Monitor model data**** for differences between training and serving datasets.
* ****Monitor new data**** for differences between any baseline and target dataset.
* ****Profile features in data**** to track how statistical properties change over time.
* ****Set up alerts on data drift**** for early warnings to potential issues.

# **Version and track datasets in experiments**

# **Create a data labeling project and export labels**

Data ingestion with Azure Data factory

How to build a data ingestion pipeline with Azure Data Factory(ADF).

This pipeline is used to ingest data for use with AML.

<https://docs.microsoft.com/en-us/azure/machine-learning/how-to-data-ingest-adf>

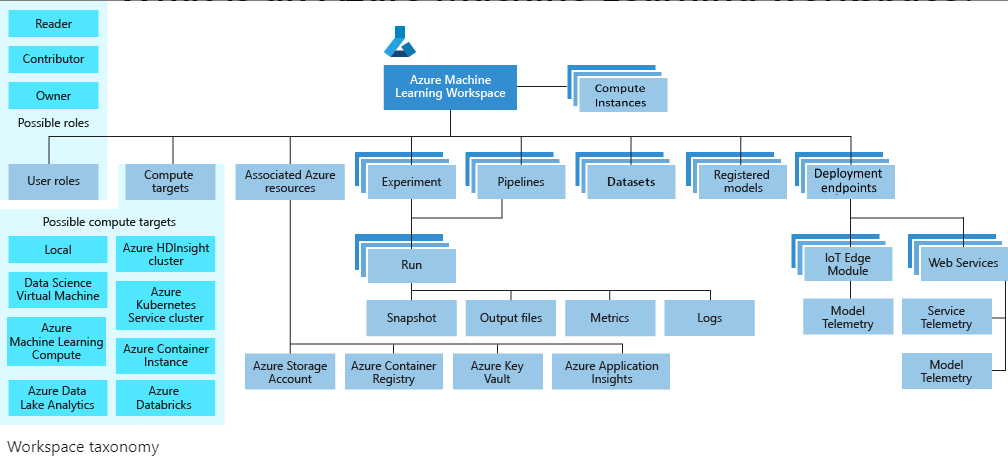
# **DevOps for a data ingestion pipeline**

<https://docs.microsoft.com/en-us/azure/machine-learning/how-to-cicd-data-ingestion>

AML

WS

A taxonomy of the workspace is illustrated in the following diagram:



The diagram shows the following components of a workspace:

A workspace can contain Azure Machine Learning compute instances, cloud resources configured with the Python environment necessary to run Azure Machine Learning.

User roles enable you to share your workspace with other users, teams or projects.

Compute targets are used to run your experiments.

When you create the workspace, associated resources are also created for you.

Experiments are training runs you use to build your models.

Pipelines are reusable workflows for training and retraining your model.

Datasets aid in management of the data you use for model training and pipeline creation.

Once you have a model you want to deploy, you create a registered model.

Use the registered model and a scoring script to create a deployment endpoint.

Machine learning tasks read and/or write artifacts to your workspace.

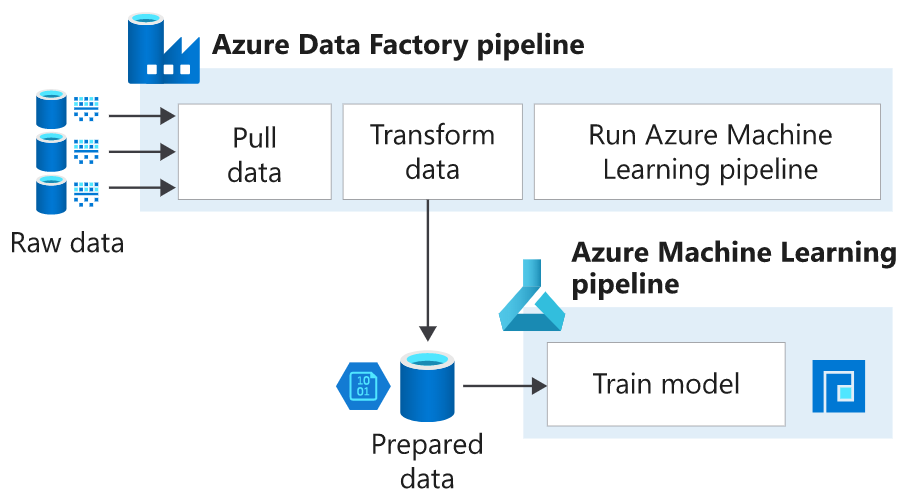
* Run an experiment to train a model - writes experiment run results to the workspace.
* Use automated ML to train a model - writes training results to the workspace.
* Register a model in the workspace.
* Deploy a model - uses the registered model to create a deployment.
* Create and run reusable workflows.
* View machine learning artifacts such as experiments, pipelines, models, deployments.
* Track and monitor models.

You can use an Environment object on your local compute to:

* Develop your training script.
* Reuse the same environment on Azure Machine Learning Compute for model training at scale.
* Deploy your model with that same environment.

# **Data ingestion in Azure Machine Learning**

## **Use Azure Data Factory**



## **Use the Python SDK**

## 

