Microsoft Azure

Designing and Implementing an Azure Data Science Solution [DP-100]

1

About This Course

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

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

© Copyright Microsoft Corporation. All rights reserved.

2

Course Agenda

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

3

Lab Environment

The course emphasizes hands-on learning

You will need:

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

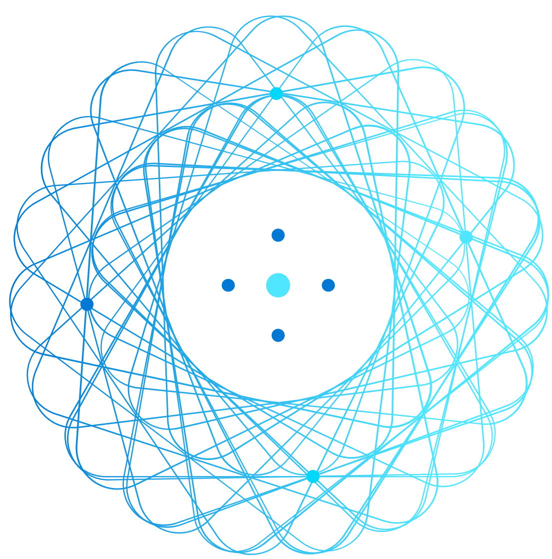
4

Microsoft Azure

5

Microsoft Azure

Module 1: Getting Started with Azure Machine Learning



6

Agenda

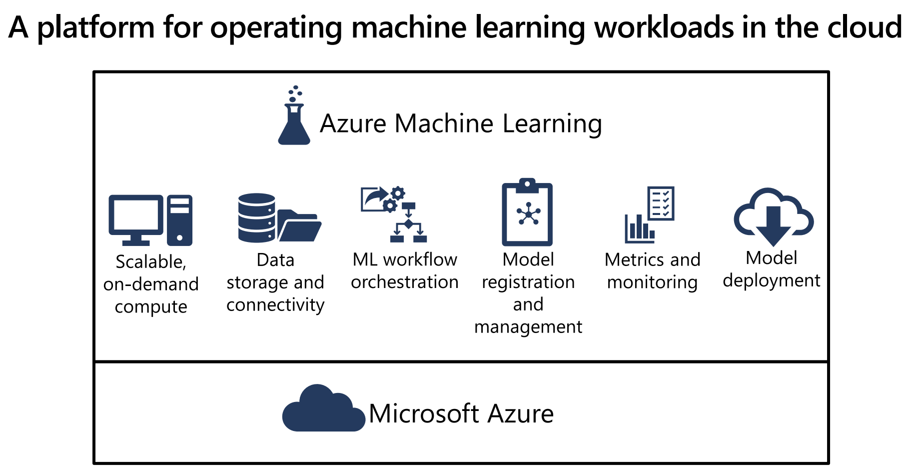
- Introduction to Azure Machine Learning
- Working with Azure Machine Learning

7

Introduction to Azure Machine Learning

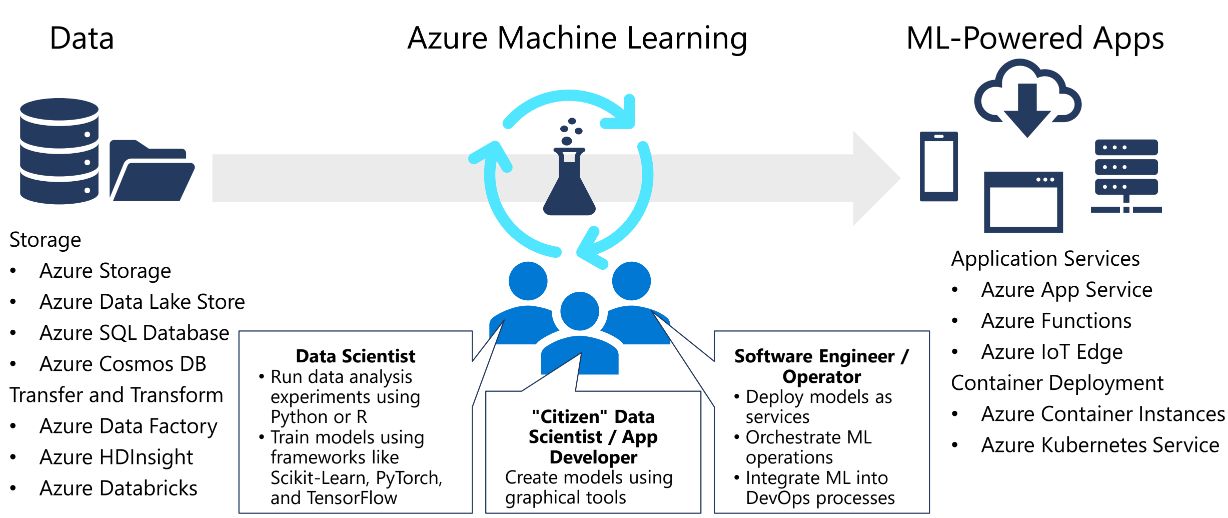
8

What is Azure Machine Learning?



9

Azure Machine Learning in Context

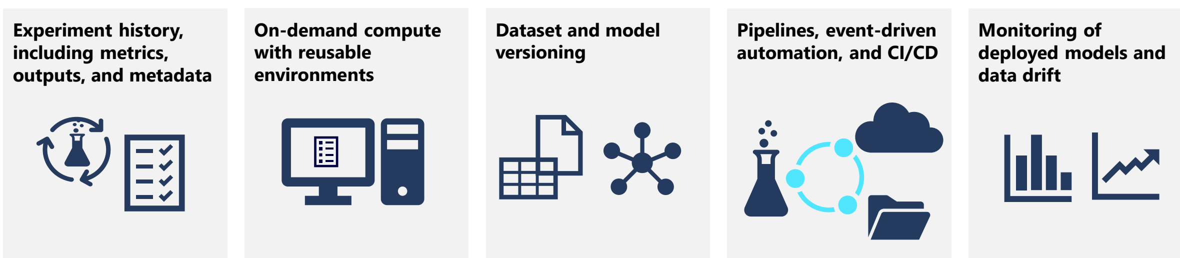


10

Machine Learning Operationalization (ML Ops)

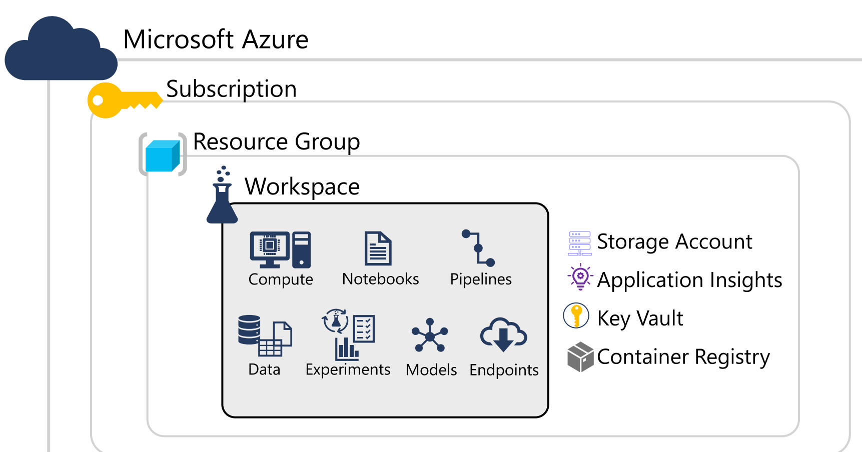
Based on *DevOps* principles, including:

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



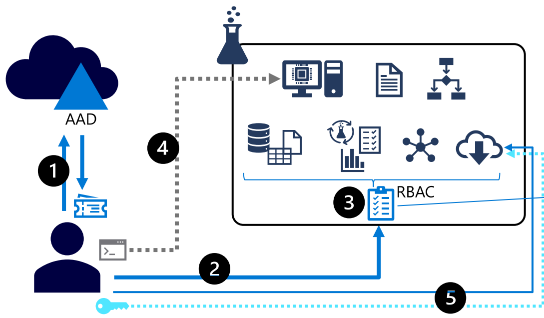
11

Azure Machine Learning Workspaces



12

Access Control and Permissions



Default RBAC permissions

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

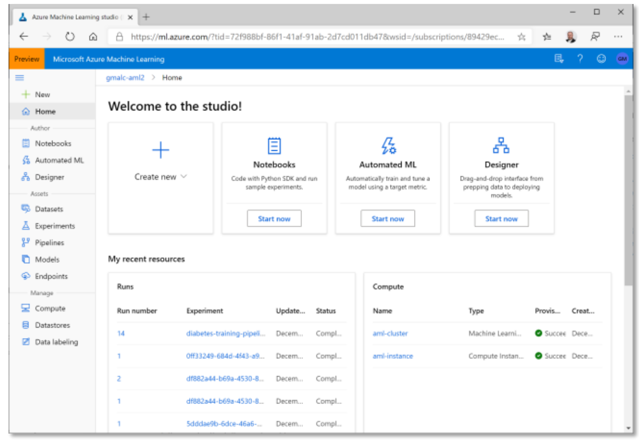
1. User signs into Azure Active Directory (AAD) and obtains token
2. Token grants access to Azure Machine Learning workspace
3. Role-based access control (RBAC) permissions control resource access
4. Compute resources can optionally allow access via SSH
5. Deployed service endpoints can use key or token-based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

13

Working with Azure Machine Learning

14

Azure Machine Learning studio



Manage compute and data

Run experiments

View metrics and logs

Manage and deploy models

Manage service endpoints

Label image data

Use graphical modeling tools:

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

15

The Azure Machine Learning SDK for Python

Python programming interface for Azure Machine Learning

```
pip install azureml-sdk
```

```
from azureml.core import Workspace

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


16

Azure Machine Learning CLI Extension

Cross-platform command-line interface for Azure Machine Learning

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

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

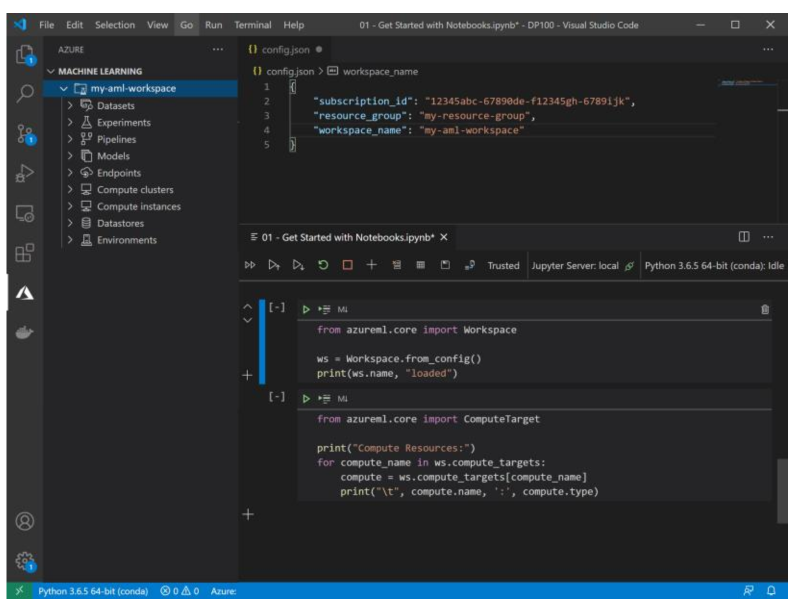

17

Visual Studio Code

Cross-platform code editor and integrated development environment

Tools for machine learning provided through *extensions*.

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

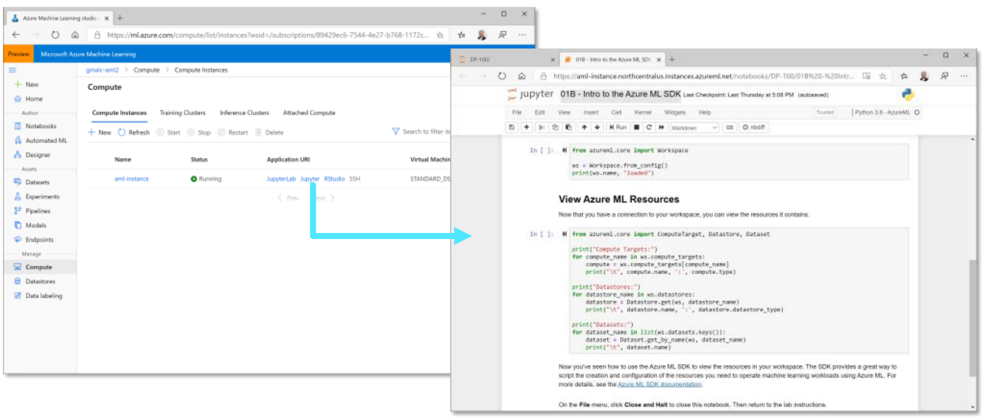


18

Azure Machine Learning Compute Instances

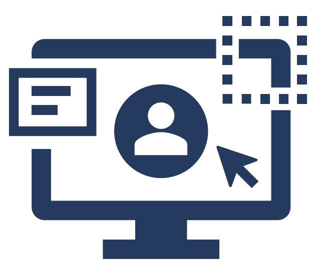
A cloud-based development workstation right in your workspace

Built-in Jupyter, JupyterLab, and RStudio



19

Lab: Create an Azure Machine Learning Workspace



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

20

References

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

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

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

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

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

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

21

Microsoft Azure

22

Microsoft Azure

Module 2: No-Code Machine Learning

Start : 13.00

23

Agenda

- Automated Machine Learning
- Azure Machine Learning Designer

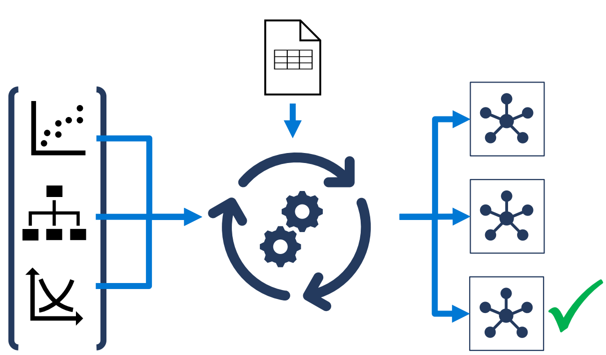
24

Automated Machine Learning

25

What is Automated Machine Learning?

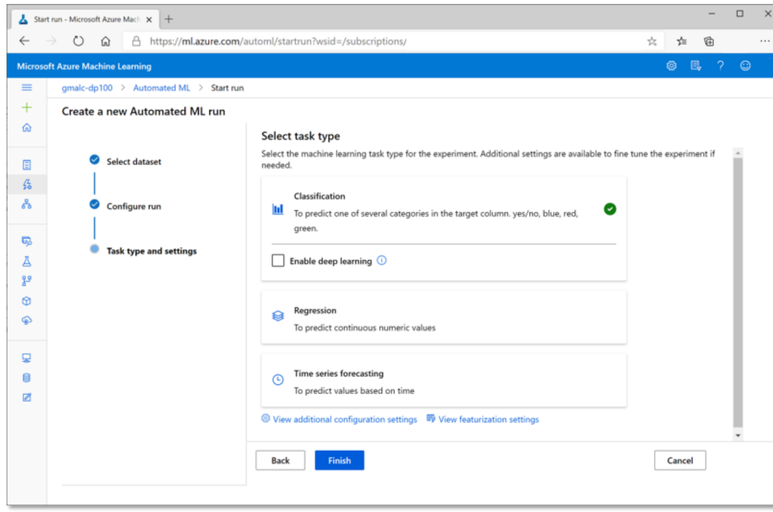
Train multiple models in parallel, varying algorithm and preprocessing
Find the "best" model based on a specific performance metric



26

Automated ML in Azure Machine Learning Studio

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



27

Configuration and Featurization

Configuration Options

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

Featurization

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

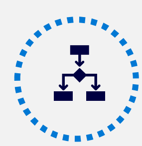
28

Lab: Use Automated Machine Learning

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

29

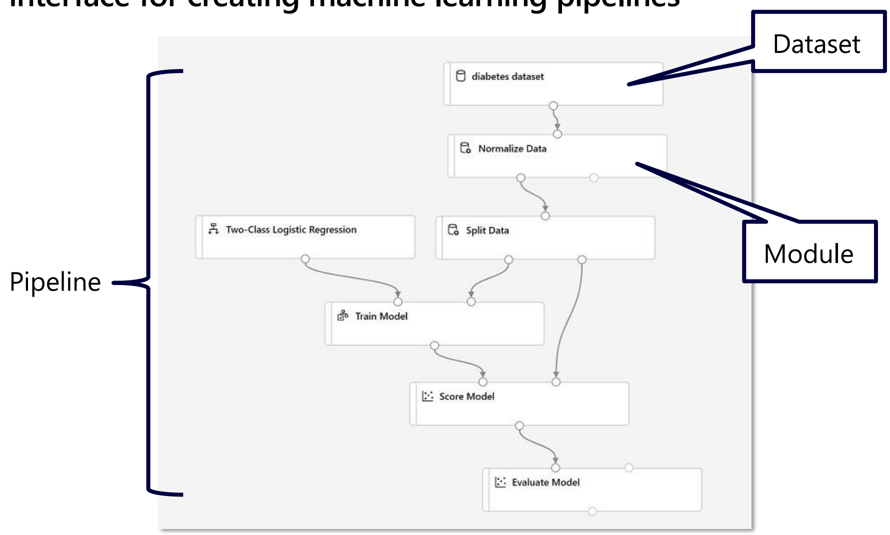
Azure Machine Learning Designer



30

What is Azure Machine Learning Designer?

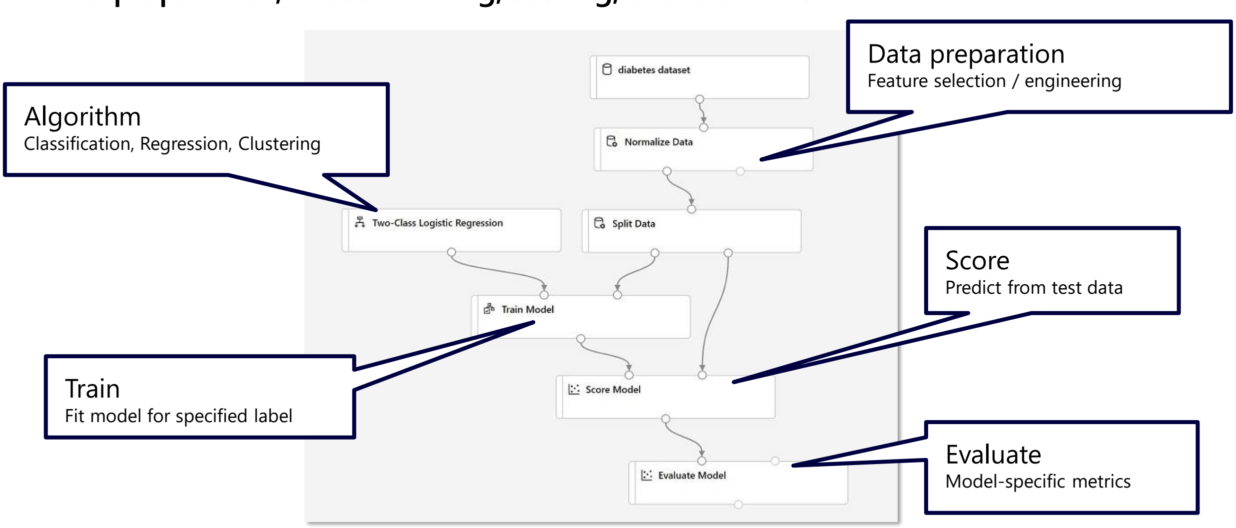
A visual interface for creating machine learning pipelines



31

Training Pipelines

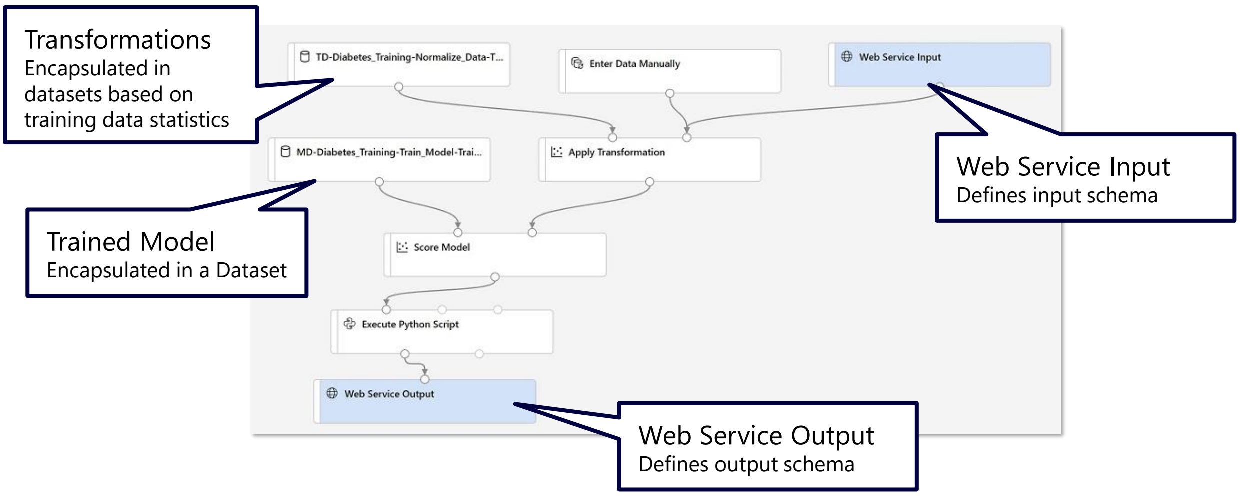
Data preparation, model training, scoring, and evaluation



32

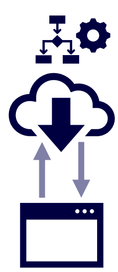
Inference Pipelines

Use the trained model to get predictions from new data



33

Publishing a Service Endpoint

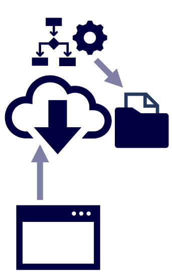


Deploy a Real-Time Pipeline:

Specify deployment target:

- Azure Container Instance
- Azure Kubernetes Services Inference Compute

Submit new data to an HTTP endpoint for immediate results



Publish a Batch Pipeline

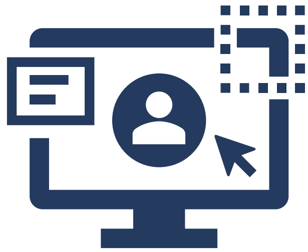
Runs on Azure Machine Learning Training Compute

Initiate a pipeline experiment run through an HTTP endpoint

Results are saved in the run output

34

Lab: Use Azure Machine Learning Designer



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

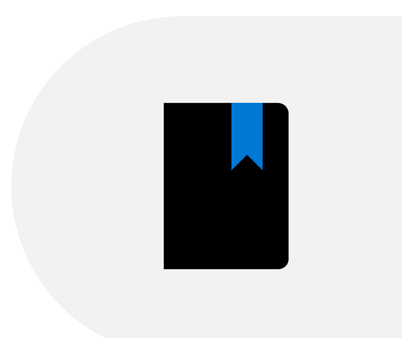
35

References

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

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

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



36

Microsoft Azure

37

Microsoft Azure

Module 3: Running Experiments and Training Models

Start 09.20

38

Agenda

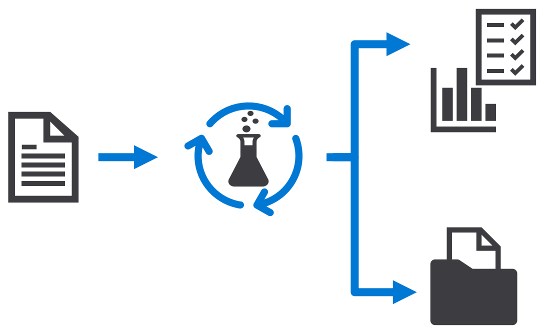
- Introduction to Experiments
- Training and Registering Models

39

Introduction to Experiments

40

What is an Experiment?



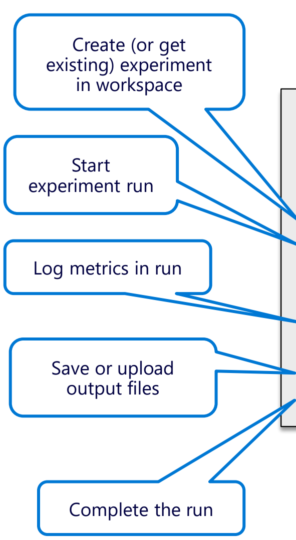
An executable process that is run one or more times – often a script

Each run generates metrics and output files

Metadata and events are recorded in log files

41

Running an Experiment Inline




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

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

42

Running a Script as an Experiment

Script:

```
from azureml.core import Run

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

Get the experiment run context for the current script

Control code (to initiate and monitor experiment run):

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

ws = Workspace.from_config()

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

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

Define run settings for the experiment script

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

43

Using MLflow

Using MLflow Inline

```
from azureml.core import Experiment
import mlflow

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

Configure MLflow to log to the Azure Machine Learning workspace

Create an experiment in the workspace

Create and start an MLflow run of the Azure ML experiment

Using MLflow with Scripts

Script:

```
import mlflow

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

Control code:

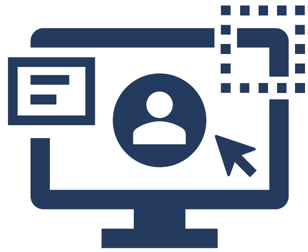
```
sc = ScriptRunConfig(source_directory='my_dir',
                     script='script.py'
                     environment=env)

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

An **Environment** that includes the mlflow package

Tracking URI is set to workspace automatically

44

Lab: Run Experiments



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

45

Training and Registering Models

46

Training a Model in a Script

Script:

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

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

Control code:

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

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

Run script in an environment that includes required ML framework

Scikit-Learn, PyTorch, TensorFlow, ...

47

Using Script Arguments

Script

Parse script arguments

```
import argparse

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

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

Use argument values in script

Control code:

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

Specify named arguments in ScriptRunConfig

48

Registering a Model

Register from training run:

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

Model saved in run **outputs**

Register from local file(s)

```
from azureml.core import Model

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

Local model (can be file or folder)

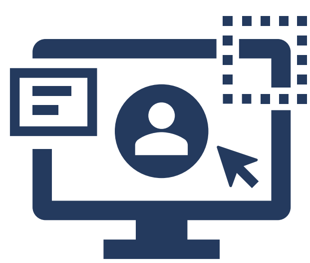
Retrieve registered models

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

Models are automatically versioned based on name

49

Lab: Train Models



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

50

References

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

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

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

51

Microsoft Azure

52

Microsoft Azure

Module 4: Working with Data

53

Agenda

- Working with Datastores
- Working with Datasets

54

Working with Datastores

55

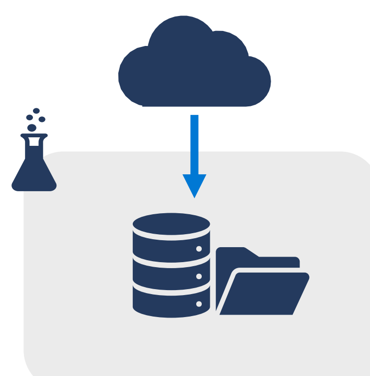
What are Datastores?

Abstractions for cloud data sources

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

Built-in Datastores

- workspaceblobstore (default)
- workspacefilestore
- azureml_globaldatasets*



* Added when open datasets are used

56

Working with Datastores

Add a datastore in Azure Machine Learning studio

or

Use the Azure Machine Learning SDK:

Register a new datastore of a specific type

```
from azureml.core import Workspace, Datastore

ws = Workspace.from_config()

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

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

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

Get registered datastore by name

Add or retrieve data

57

Considerations for Datastores

✓ Configure blob storage performance type and replication for your needs

✓ *Parquet* file format generally performs better than *CSV*

✓ You can manage the default datastore using the SDK

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

58

Working with Datasets

59

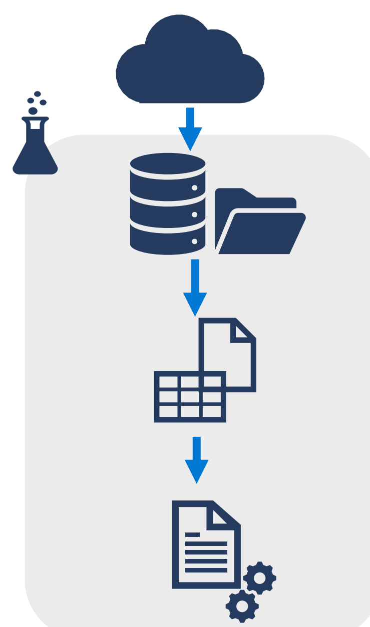
What are Datasets?

Versioned data objects for experiments

Usually based on datastore contents

Two types:

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



60

Creating and Registering Datasets

Add a dataset in Azure Machine Learning studio

or

Use the Dataset object in the SDK

```
from azureml.core import Dataset

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

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

Create tabular dataset

Register in workspace

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

```
from azureml.core import Dataset

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

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

Create file dataset

Register in workspace

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

61

Working with Tabular Datasets

Pass a dataset as a script argument

ScriptRunConfig:

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

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

Required to work with datasets in script

Pass dataset object as script argument

Script:

```
from azureml.core import Run, Dataset

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

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

Argument contains dataset ID

Get dataset by ID

Convert to dataframe

or

Pass a dataset as a named input

ScriptRunConfig:

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

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

Required to work with datasets in script

Pass dataset as named input

Script:

```
from azureml.core import Run

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

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

Argument still required!

Retrieve named dataset from input_datasets

Convert to dataframe

62

Working with File Datasets

Pass a dataset as a script argument

ScriptRunConfig:

Required to work with datasets in script

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

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

Pass dataset object as *download* or *mount*

Script:

Argument contains data reference

```
from azureml.core import Run
import glob

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

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

Get file paths from data reference

or

Pass a dataset as a named input

ScriptRunConfig:

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

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

Pass dataset as named input

Script:

Argument still required!

```
from azureml.core import Run
import glob

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

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

Retrieve named dataset from input_datasets

Get file paths from data reference

63

Dataset Versioning

Create a new version of an existing dataset

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

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

Specify a version to retrieve

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

Version number

64

Lab: Work with Data

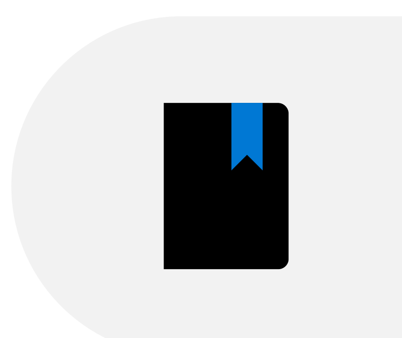
1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Work with data** exercise

65

References

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

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



66

67

Module 5: Working with Compute

68

Agenda

- Environments
- Compute Targets

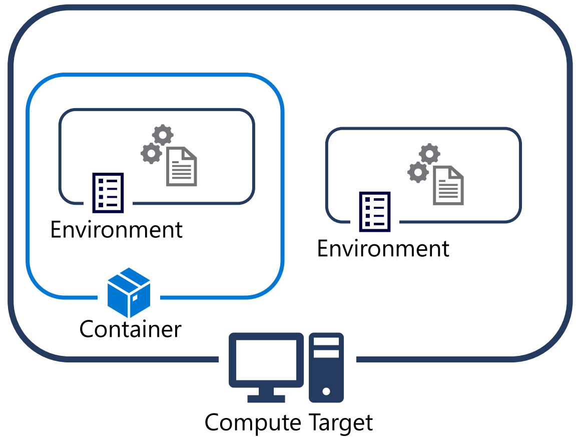
69

Environments

70

Run Contexts for Experiments

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



71

Explicitly Creating Environments

Create from specification file

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

File in standard YAML format for Conda environments

Create from existing conda environment

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

Existing conda environment on local compute

Create with specified packages

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

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

Most experiments require azureml-defaults

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

72

Configuring Environment Containers

Use the *docker* section of the environment

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

Create environment in a container (default)

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

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

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

…or create one from a dockerfile

Override managed Python configuration

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

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

73

Registering and Reusing Environments

Register an environment in the workspace

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

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

View Registered Environments

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

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

Retrieve and use an environment

Enables you to reuse the environment on any compute target

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

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

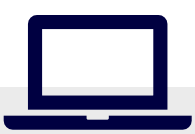
Environment will be created if not already on compute target

74

Compute Targets

75

Compute Options for Experiment Runs



Local Compute

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

Compute Cluster

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

Attached Compute

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

76

Creating a Compute Cluster

Create in Azure Machine Learning studio

or

Use the SDK

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

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

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

Cluster will scale up to this size as required

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

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

77

Attaching Azure Databricks Compute

Create in Azure Machine Learning studio

or

Use the SDK

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

compute_name = 'db_cluster'

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

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

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

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

78

Using Compute Targets

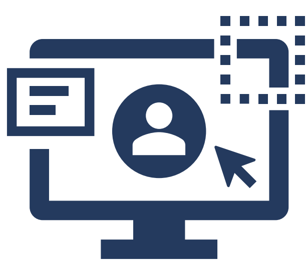
Specify the compute target for an experiment

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

Specify the compute target name or object

79

Lab: Work with Compute



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

80

References

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

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

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

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

81

Microsoft Azure

82

Microsoft Azure

Module 6: Orchestrating Machine Learning Workflows

83

Agenda

- Introduction to Pipelines
- Publishing and Running Pipelines

84

Introduction to Pipelines

85

What is a Pipeline?

A workflow of machine learning tasks

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

An executable process

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

The foundation for automating ML operationalization tasks

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

86

Pipeline Steps

Common Step Types:

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

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

87

Passing Data Between Steps

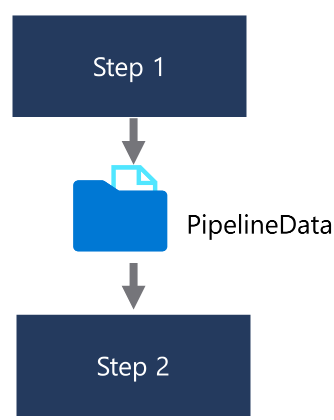
Use a PipelineData object:

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

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

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

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

PipelineData output ...

...PipelineData input



88

Pipeline Step Reuse

Reuse output without re-running the step

Control this behavior with the **allow_reuse** parameter

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

Reuse cached step output if unchanged

Force all steps to re-run:

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

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

Override step reuse

89

Publishing and Running Pipelines

90

Pipeline Endpoints

Publish a pipeline to create a REST endpoint

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

Post a JSON request to initiate a pipeline

- Requires an authorization header
- Returns a run ID

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

91

Pipeline Parameters

Parameterize a pipeline before publishing

Increases flexibility by allowing variable input

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

Pass parameters in the JSON request

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

92

Scheduling Pipelines

Schedule pipeline runs based on time

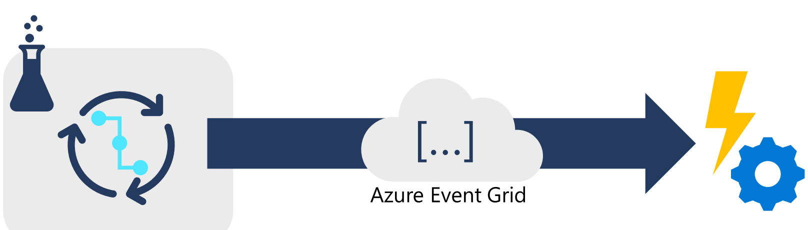
```
daily = ScheduleRecurrence(frequency='Day', interval=1)
pipeline_schedule = Schedule.create(ws, name='Daily Training',
                                    description='trains model every day',
                                    pipeline_id=published_pipeline_id,
                                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

Trigger pipeline runs when data changes

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

93

Event-Driven Workflows



Define events for:

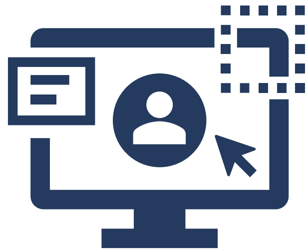
- Run completion
- Run failure
- Model registration
- Model deployment
- Data drift detection

Trigger automated actions:

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

94

Lab: Create a Pipeline



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

95

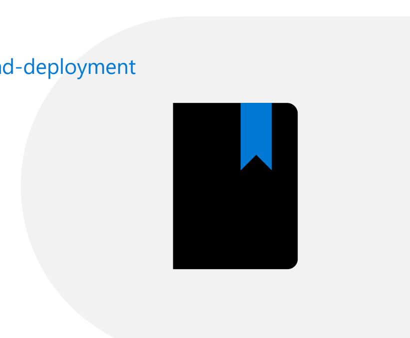
References

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

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

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

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



96

Microsoft Azure

97

Microsoft Azure

Module 7: Deploying and Consuming Models

Start : 09.15

98

Agenda

- Real-time Inferencing
- Batch Inferencing
- Continuous Integration and Delivery

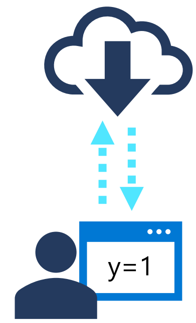
99

Real-time Inferencing

100

What is Real-Time Inferencing?

Immediate prediction from new data
Usually deployed as a web service endpoint



101

Deploying a Real-Time Inferencing Service

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

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

102

Consuming a Real-time Inferencing Service

Use the SDK

```
import json

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

Use the REST Endpoint

```
import json
import requests

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

103

Troubleshooting a Real-Time Inferencing Service

Check the service state

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

Review service logs

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

Deploy to a local container

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

Modify entry script to debug, and then reload to test

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

104

Lab: Create a Real-time Inference Service

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

105

Batch Inferencing

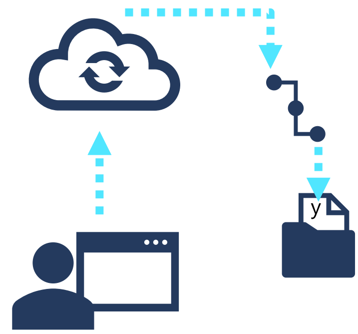
106

What is Batch Inferencing?

Asynchronous prediction from batched data

Implemented as a pipeline

- Typically using a ParallelRunStep for scalability



107

Creating a Batch Inferencing Pipeline

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

108

Publishing a Batch Inferencing Service

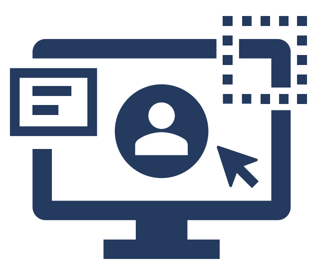
Publish the batch pipeline as a REST service
Use the pipeline endpoint to initiate batch inferencing

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

```
import requests

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

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

109

Lab: Create a Batch Inference Service



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

110

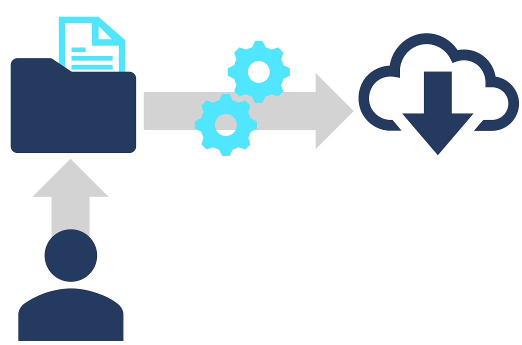
Continuous Integration and Delivery

111

What is Continuous Integration and Delivery (CI/CD)?

A core DevOps practice for software development and deployment

- Code and other assets are managed in a central source control system
- Updates can trigger build and release processes that:
  - Apply policies to accept/reject changes
  - Integrate multiple changes into a single build
  - Perform testing and validation
  - Deploy new versions of software (including machine learning models) into staging and production environments



112

Azure Machine Learning and Azure Pipelines

- **Define build and release pipelines to train and deploy models**
  - Using Python or CLI
- **Install the Azure Pipelines *Machine Learning* extension:**
  - Trigger a release pipeline on model registration
  - Use predefined tasks to:
    - Run a published Azure Machine Learning pipeline
    - Profile a model
    - Deploy a model

113

Azure Machine Learning and GitHub Actions

- **Create a workflow to run on a specified GitHub event**
  (for example, pushing an update to a branch)
  - Use the **aml-run** action to run an Azure machine Learning pipeline or experiment
  - Use the **aml-registermodel** action to register a model
  - Use the **aml-deploy** action to deploy a model

114

References

Microsoft Learn: Deploy real-time machine learning services with Azure Machine Learning
https://docs.microsoft.com/learn/modules/register-and-deploy-model-with-amls

Microsoft Learn: Deploy batch inference pipelines with Azure Machine Learning
https://docs.microsoft.com/learn/modules/deploy-batch-inference-pipelines-with-azure-machine-learning

Azure Machine Learning model deployment documentation
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where

CI/CD with Azure Pipelines documentation
https://docs.microsoft.com/azure/devops/pipelines/targets/azure-machine-learning

CI/CD with GitHub Actions documentation
https://docs.microsoft.com/azure/machine-learning/how-to-github-actions-machine-learning

115

Microsoft Azure

116

Microsoft Azure

Module 8: Training Optimal Models

117

Agenda

- Hyperparameter Tuning
- Automated Machine Learning

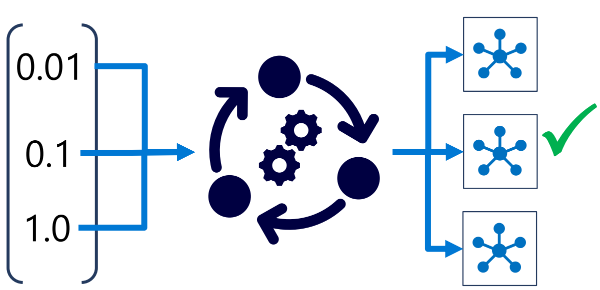
118

Hyperparameter Tuning

119

What is Hyperparameter Tuning?

Train multiple models, using the same algorithm but varying hyperparameter values
Find the "best" model based on a specific performance metric



120

Hyperparameter Search Space

Discrete Hyperparameters

Choice (any list or range)

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

Continuous Hyperparameters

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

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

121

Hyperparameter Sampling

Grid Sampling

Tries every combination of discrete hyperparameter values

Can only be used when all hyperparameters are discrete

Random Sampling

Randomly selects hyperparameter values

Can be used with discrete and continuous hyperparameter combinations

Bayesian Sampling

Selects hyperparameter values based on performance of previous selection

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

```
from azureml.train.hyperdrive import RandomParameterSampling

param_sampling = RandomParameterSampling(param_space)
```

122

Tuning Hyperparameters with Hyperdrive

Hyperparameters in sampling collection are passed as arguments

Experiment script

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

Log performance metric for evaluation

ScriptRunConfig for training script

Params added to script arguments

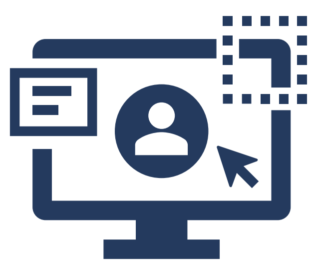
Name must match logged metric

Hyperdrive run configuration

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

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

123

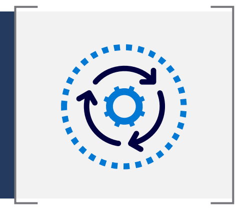
Lab: Tune Hyperparameters



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

124

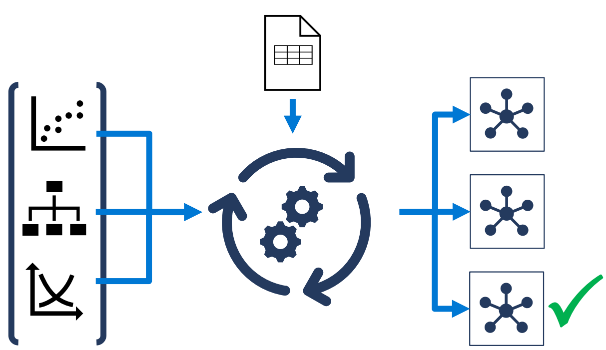
Automated Machine Learning



125

Automated Machine Learning – A Reminder

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



126

Preparing Data for Automated Machine Learning

Training Data – tabular data including features and label
Validation Data – optional table for model validation

You can use a **Dataset** or a Pandas dataframe

```
tab_ds = ws.datasets.get("tabular dataset")

train_ds, test_ds = tab_ds.random_split(percentage=0.7, seed=123)
```

Optional split for training and test
(if only training data is provided, cross-validation will be applied automatically)

127

Running an Automated Machine Learning Experiment

Configure an automated machine learning experiment run

```
from azureml.train.automl import AutoMLConfig

automl_config = AutoMLConfig(name='Automated ML Experiment',
                             task='classification',
                             compute_target=aml_cluster,
                             training_data = train_ds,
                             validation_data = test_ds,
                             label_column_name='Label',
                             iterations=20,
                             primary_metric = 'AUC_weighted',
                             max_concurrent_iterations=4,
                             featurization='auto')

automl_run = automl_experiment.submit(automl_config)
```

Metrics are dependent on task

(use **automl_utils.get_primary_metrics** to find them)

128

Monitoring and Reviewing Automated ML Runs

Monitor runs in Azure Machine Learning studio or widget

Find the best-performing run and the model it trained:

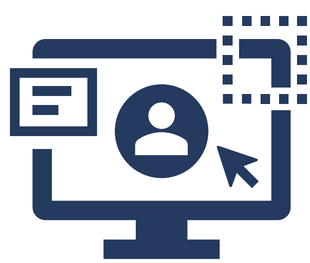
```
best_run, fitted_model = automl_run.get_output()
best_run_metrics = best_run.get_metrics()
for metric_name in best_run_metrics:
    metric = best_run_metrics[metric_name]
    print(metric_name, metric)
```

View model pipeline details:

```
for step_ in fitted_model.named_steps:
    print(step)
```

129

Lab: Use Automated Machine Learning from the SDK



1. View the lab instructions at https://aka.ms/mslearn-dp100
2. Complete the **Use automated machine learning from the SDK** exercise

130

References

Microsoft Learn: Tune hyperparameters with Azure Machine Learning
https://docs.microsoft.com/learn/modules/tune-hyperparameters-with-azure-machine-learning/

Microsoft Learn: Automate machine learning model selection with Azure Machine Learning
https://docs.microsoft.com/learn/modules/automate-model-selection-with-azure-automl/

Azure Machine Learning hyperparameter tuning documentation
https://docs.microsoft.com/azure/machine-learning/how-to-tune-hyperparameters

Azure Machine Learning automated machine learning documentation
https://docs.microsoft.com/azure/machine-learning/how-to-configure-auto-train

131

Microsoft Azure

132

Microsoft Azure

Module 9: Responsible Machine Learning

133

Agenda

- Differential Privacy
- Model Interpretability
- Fairness

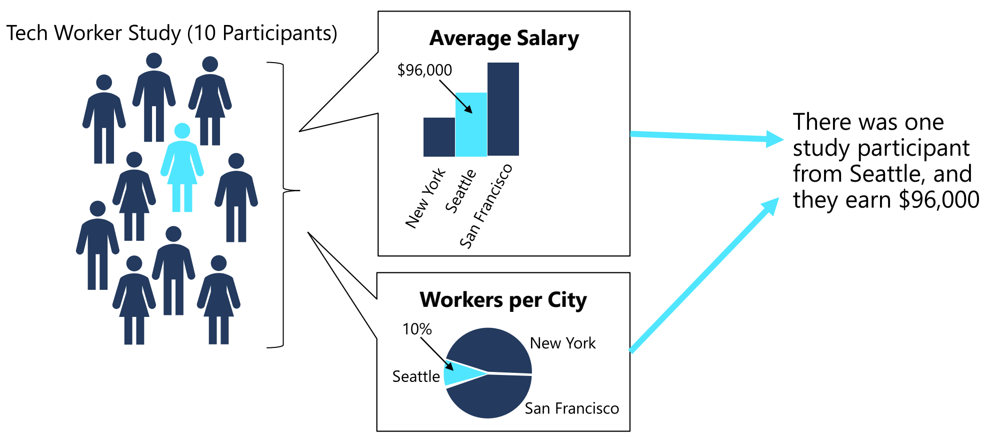
134

Differential Privacy

135

The Data Privacy Problem

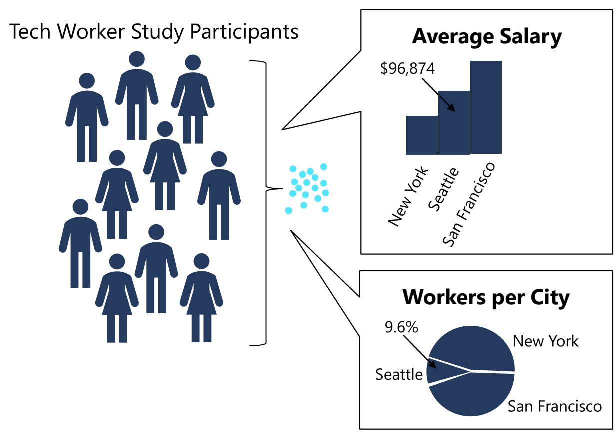
Studies are ethically and legally required to protect personal information
Repeated analyses of aggregated results can reveal details about individuals



136

What is Differential Privacy?

The analysis function adds random "noise" to the data
Results are statistically consistent, non-deterministic approximations



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

137

Model Interpretability

138

Model Interpretability in Azure Machine Learning

Statistical explanation of feature importance

Quantifies the influence of each feature on prediction

Important to identify bias or unintended correlation in the model

Based on the Open Source *Interpret-Community* package

Includes explainers based on common model interpretation algorithms like:

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

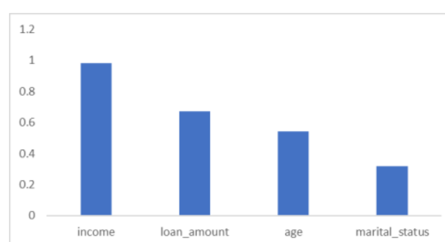
139

Global and Local Feature Importance

Global Feature Importance

Overall feature importance for all test data

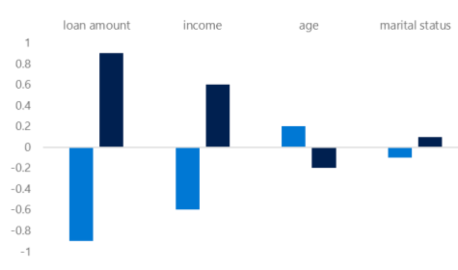
Indicates the relative influence of each feature on the predicted label



Local Feature Importance

Feature importance for an individual prediction

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



140

Explainers

Use the azureml-interpret package

Create an explainer:

MimicExplainer – global surrogate model that approximates your model

TabularExplainer – Invokes direct SHAP explainer based on model architecture

PFIExplainer – Permutation Feature Importance based on feature shuffling

Get global or local feature explanations

```
from interpret.ext.blackbox import TabularExplainer

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

141

Adding Explanations to Training Experiments

In the training script, import the ExplanationClient class

Generate explanations and upload them to the run

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

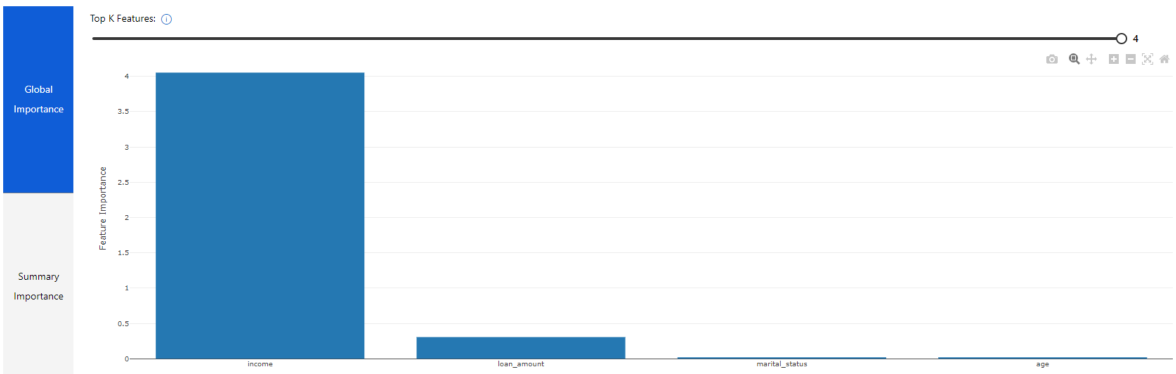
Use ExplanationClient to download explanations

```
from azureml.interpret.explanation_client import ExplanationClient

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

142

Visualizing Model Explanations

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



143

Interpretability During Inferencing

Register a lightweight scoring explainer with the model

```
scoring_explainer = KernelScoringExplainer(explainer)
save(scoring_explainer, directory='dir', exist_ok=True)
Model.register(ws, model_name='model', model_path='dir/model.pkl')
Model.register(ws, model_name='explainer', model_path='dir/scoring_explainer.pkl')
```

Use the model and the explainer in the service scoring script

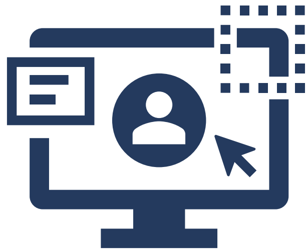
```
def run(raw_data):
    data = json.loads(raw_data)['data']
    predictions = model.predict(data)
    local_importance_values = explainer.explain(data)
    return {"predictions":predictions.tolist()), "importance":local_importance_values}
```

Deploy a service with the model and explainer

```
service = Model.deploy(ws, 'classify_svc', [model, explainer], inf_config, dep_config)
```

144

Lab: Interpret Models



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

145

Fairness

146

What is Fairness?

Absence of negative impact on groups based on:

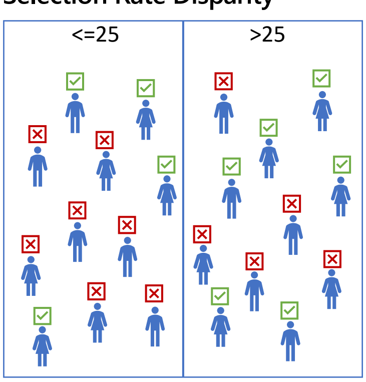
- Ethnicity
- Gender
- Age
- Physical disability
- other sensitive features

147

Evaluating Model Fairness

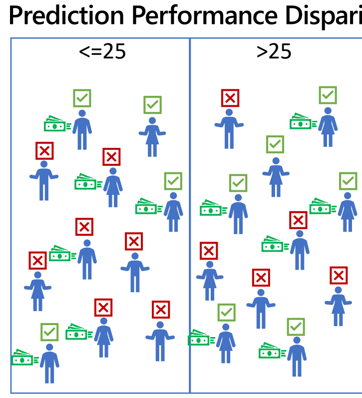
Example: Loan repayment binary classification for two age groups

Selection Rate Disparity



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

Prediction Performance Disparity



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

148

References

Microsoft Learn: Explore differential privacy
https://docs.microsoft.com/learn/modules/explore-differential-privacy

Microsoft Learn: Explain machine learning models with Azure Machine Learning
https://docs.microsoft.com/learn/modules/explain-machine-learning-models-with-azure-machine-learning

Microsoft Learn: Detect and mitigate unfairness in models with Azure Machine Learning
https://docs.microsoft.com/learn/modules/detect-mitigate-unfairness-models-with-azure-machine-learning

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

149

Microsoft Azure

150

Microsoft Azure

Module 10: Monitoring Models

151

Agenda

- Monitoring Models with Application Insights
- Monitoring Data Drift

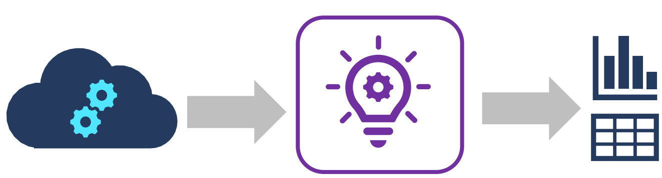
152

Monitoring Models with Application Insights

153

What is Application Insights?

An Application Performance Management service in Azure
Enables capture, storage, and analysis of telemetry data



154

Enabling Application Insights

Determine the Application Insights resource for your workspace

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

Enable in a new service deployment configuration using the SDK:

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

Enable for existing deployed services:

Configure AKS deployment in Azure Machine Learning studio

Update deployed service using the SDK

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

155

Capturing and Viewing Application Insights Data

Print log data in the scoring script

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

Query Logs in Application Insights

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

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

156

Lab: Monitor a Model

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

157

Monitoring Data Drift

158

What is Data Drift?

Changing data trends that can affect the accuracy of trained models

159

Creating a Data Drift Monitor

Monitor by Comparing Datasets

Baseline dataset (original training data)

Target dataset for comparison over time (requires timestamp column)

Backfill to populate a data drift profile from target dataset

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

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

160

Data Drift Schedules and Alerts

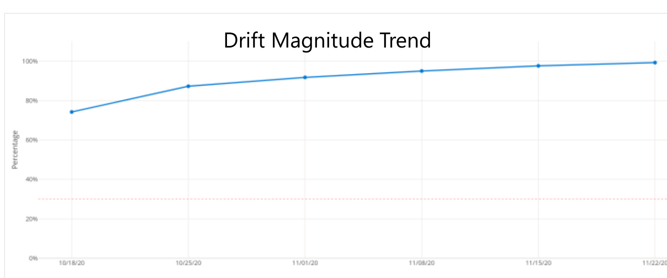
On creation, specify:

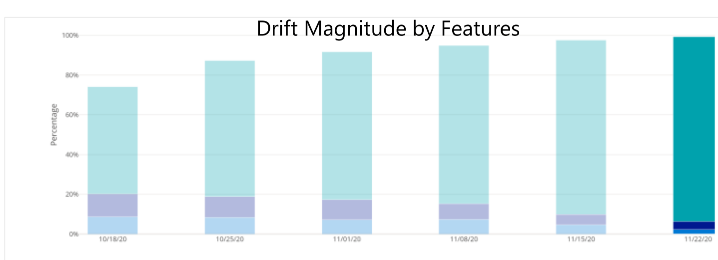
- Frequency
- Drift threshold for alerting
- Alert configuration
- Schedule start (for model data drift monitors)
- Data latency (for dataset data drift monitors)

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

161

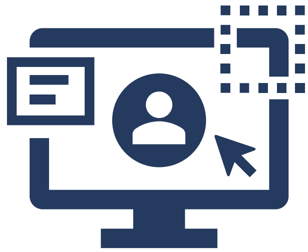
Reviewing Data Drift





162

Lab: Monitor Data Drift



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

163

References

Microsoft Learn: Monitor models with Azure Machine Learning
https://docs.microsoft.com/learn/modules/monitor-models-with-azure-machine-learning

Microsoft Learn: Monitor data drift with Azure Machine Learning
https://docs.microsoft.com/learn/modules/monitor-data-drift-with-azure-machine-learning

Azure Machine Learning monitoring with Application Insights documentation
https://docs.microsoft.com/azure/machine-learning/how-to-enable-app-insights

Azure Machine Learning data drift documentation
https://docs.microsoft.com/azure/machine-learning/how-to-monitor-datasets

164

Microsoft Azure

165