- **Vendor: Microsoft**
- **Exam Code: DP-100**
- **Exam Name: Designing and Implementing a Data Science Solution on Azure**
- **New Updated Questions from Braindump2go (Updated in May/2020)**

QUESTION 163
Hotspot Question
The finance team asks you to train a model using data in an Azure Storage blob container named finance- data.
You need to register the container as a datastore in an Azure Machine Learning workspace and ensure that an error will be raised if the container does not exist.
How should you complete the code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
The code will cause a previous version of the saved model to be overwritten.	○	○
The version number will now be 4.	○	○
The latest version of the stored model will have a property of value: 87.43.	○	○

Answer:

Answer Area

	Yes	No
The code will cause a previous version of the saved model to be overwritten.	○	○
The version number will now be 4.	○	○
The latest version of the stored model will have a property of value: 87.43.	○	○

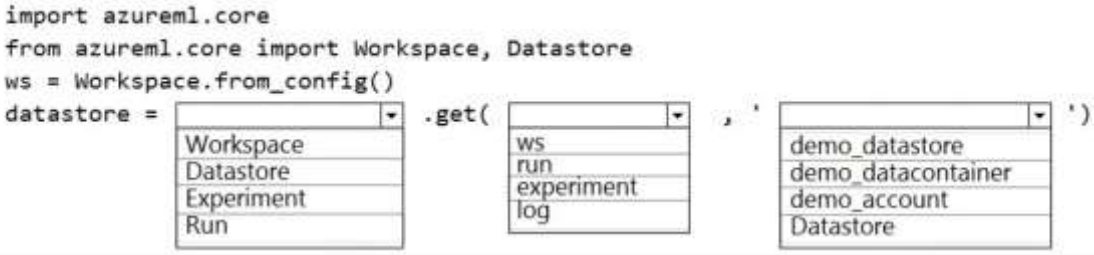
Explanation:
Box 1: register_azure_blob_container
Register an Azure Blob Container to the datastore.
Box 2: create_if_not_exists = False
Create the file share if it does not exists, defaults to False.
Reference:
https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.datastore.datastore

QUESTION 164
Hotspot Question
A coworker registers a datastore in a Machine Learning services workspace by using the following code:

```
Datastore.register_azure_blob_container(workspace=ws,
 datastore_name='demo_datastore',
 container_name='demo_datacontainer',
 account_name='demo_account',
 account_key='0A0A0A-0A0A00A-0A00A0A0A0A0A',
 create_if_not_exists=True)
```

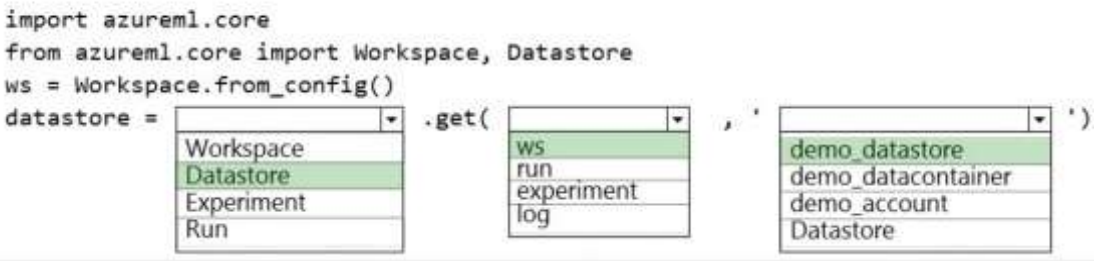
You need to write code to access the datastore from a notebook.
How should you complete the code segment? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area



Answer:

Answer Area



Explanation:
Box 1: DataStore
To get a specific datastore registered in the current workspace, use the get() static method on the Datastore class:
# Get a named datastore from the current workspace
datastore = Datastore.get(ws, datastore_name='your datastore name')
Box 2: ws
Box 3: demo_datastore
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-access-data

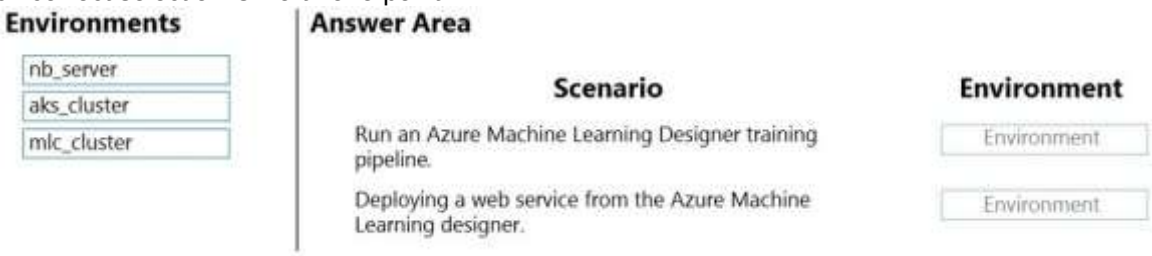
QUESTION 165
Drag and Drop Question
An organization uses Azure Machine Learning service and wants to expand their use of machine learning.
You have the following compute environments. The organization does not want to create another compute environment.

Environment name	Compute type
nb_server	Compute Instance
aks_cluster	Azure Kubernetes Service
mlc_cluster	Machine Learning Compute

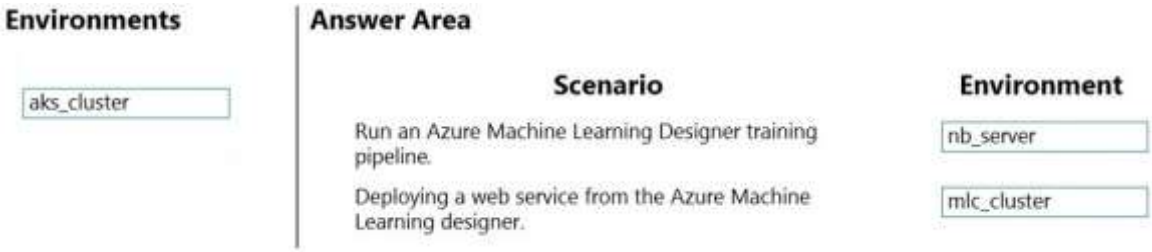
You need to determine which compute environment to use for the following scenarios.
Which compute types should you use? To answer, drag the appropriate compute environments to the correct

scenarios. Each compute environment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.



Answer:



Explanation:

Box 1: nb_server

Training targets	Automated ML	ML pipelines	Azure Machine Learning designer
Local computer	yes		
Azure Machine Learning compute cluster	yes & hyperparameter tuning	yes	yes
Azure Machine Learning compute instance	yes & hyperparameter tuning	yes	yes
Remote VM	yes & hyperparameter tuning	yes	
Azure Databricks	yes (SDK local mode only)	yes	
Azure Data Lake Analytics		yes	
Azure HDInsight		yes	
Azure Batch		yes	

Box 2: mlc_cluster

With Azure Machine Learning, you can train your model on a variety of resources or environments, collectively referred to as compute targets. A compute target can be a local machine or a cloud resource, such as an Azure Machine Learning Compute, Azure HDInsight or a remote virtual machine.

Reference:

https://docs.microsoft.com/en-us/azure/machine-learning/concept-compute-target

https://docs.microsoft.com/en-us/azure/machine-learning/how-to-set-up-training-targets

QUESTION 166
Hotspot Question
You create an Azure Machine Learning compute target named **ComputeOne** by using the STANDARD_D1 virtual machine image.
You define a Python variable named was that references the Azure Machine Learning workspace. You run the following Python code:

```
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.compute_target import ComputeTargetException
the_cluster_name = "ComputeOne"
try:
  the_cluster = ComputeTarget(workspace=ws, name=the_cluster_name)
  print('Step1')
except ComputeTargetException:
  config = AmlCompute.provisioning_configuration(vm_size='STANDARD_DS12_v2', max_nodes=4)
  the_cluster = ComputeTarget.create(ws, the_cluster_name, config)
  print('Step2')
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
A new machine learning compute resource is created with a virtual machine size of STANDARD_DS12_v2 and a maximum of four nodes.	○	○
Any experiments configured to use `the_cluster` will run on ComputeOne.	○	○
The text **Step1** will be printed to the screen.	○	○

Answer:

Answer Area

	Yes	No
A new machine learning compute resource is created with a virtual machine size of STANDARD_DS12_v2 and a maximum of four nodes.	○	○
Any experiments configured to use `the_cluster` will run on ComputeOne.	○	○
The text **Step1** will be printed to the screen.	○	○

Explanation:
Box 1:Yes
ComputeTargetException class: An exception related to failures when creating, interacting with, or configuring a compute target. This exception is commonly raised for failures attaching a compute target, missing headers, and unsupported configuration values.
Create(workspace, name, provisioning_configuration)
Provision a Compute object by specifying a compute type and related configuration.
This method creates a new compute target rather than attaching an existing one.
Box 2: Yes
Box 3: No
The line before print('Step1') will fail.
Reference:
https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.compute.computetarget

QUESTION 167
Hotspot Question
You are developing a deep learning model by using TensorFlow. You plan to run the model training workload on an Azure Machine Learning Compute Instance.
You must use CUDA-based model training.
You need to provision the Compute Instance.
Which two virtual machines sizes can you use? To answer, select the appropriate virtual machine sizes in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Virtual machine size

Search by name...

Name ↑	vCPUs	GPUs	RAM	Resource disk
BASIC_A0	1		0.75 GB	20 GB
STANDARD_D3_V2	4		14 GB	200 GB
STANDARD_E64_V3	64		432 GB	1,600 GB
STANDARD_M64LS	64		512 GB	2,000 GB
STANDARD_NC12	12	2	112 GB	680 GB
STANDARD_NC24	24	4	224 GB	1,440 GB

Answer:

Answer Area

Virtual machine size

Search by name...

Name ↑	vCPUs	GPUs	RAM	Resource disk
BASIC_A0	1		0.75 GB	20 GB
STANDARD_D3_V2	4		14 GB	200 GB
STANDARD_E64_V3	64		432 GB	1,600 GB
STANDARD_M64LS	64		512 GB	2,000 GB
STANDARD_NC12	12	2	112 GB	680 GB
STANDARD_NC24	24	4	224 GB	1,440 GB

Explanation:

CUDA is a parallel computing platform and programming model developed by Nvidia for general computing on its own GPUs (graphics processing units). CUDA enables developers to speed up compute-intensive applications by harnessing the power of GPUs for the parallelizable part of the computation.

Reference:

https://www.infoworld.com/article/3299703/what-is-cuda-parallel-programming-for-gpus.html

QUESTION 168

Hotspot Question

You are preparing to use the Azure ML SDK to run an experiment and need to create compute. You run the following code:

```
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.compute_target import ComputeTargetException
ws = Workspace.from_config()
cluster_name = 'aml-cluster'
try:
  training_compute = ComputeTarget(workspace=ws, name=cluster_name)
except ComputeTargetException:
  compute_config = AmlCompute.provisioning_configuration(vm_size='STANDARD_D2_V2', vm_priority='lowpriority',
max_nodes=4)
  training_compute = ComputeTarget.create(ws, cluster_name, compute_config)
  training_compute.wait_for_completion(show_output=True)
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
If a training cluster named aml-cluster already exists in the workspace, it will be deleted and replaced.	○	○
The `wait_for_completion()` method will not return until the aml-cluster compute has four active nodes.	○	○
If the code creates a new aml-cluster compute target, it may be preempted due to capacity constraints.	○	○
The aml-cluster compute target is deleted from the workspace after the training experiment completes.	○	○

Answer:

Answer Area

	Yes	No
If a training cluster named aml-cluster already exists in the workspace, it will be deleted and replaced.	○	●
The `wait_for_completion()` method will not return until the aml-cluster compute has four active nodes.	●	○
If the code creates a new aml-cluster compute target, it may be preempted due to capacity constraints.	●	○
The aml-cluster compute target is deleted from the workspace after the training experiment completes.	○	●

Explanation:

Box 1: No
If a training cluster already exists it will be used.
Box 2: Yes
The wait_for_completion method waits for the current provisioning operation to finish on the cluster.
Box 3: Yes
Low Priority VMs use Azure's excess capacity and are thus cheaper but risk your run being pre-empted.
Box 4: No
Need to use training_compute.delete() to deprovision and delete the AmlCompute target.
Reference:
https://notebooks.azure.com/azureml/projects/azureml-getting-started/html/how-to-use-azureml/training/ train-on-amlcompute/train-on-amlcompute.ipynb
https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.compute.computetarget

QUESTION 169

Drag and Drop Question
You create a multi-class image classification deep learning experiment by using the PyTorch framework. You plan to run the experiment on an Azure Compute cluster that has nodes with GPU's.
You need to define an Azure Machine Learning service pipeline to perform the monthly retraining of the image classification model. The pipeline must run with minimal cost and minimize the time required to train the model.
Which three pipeline steps should you run in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Actions	Answer Area
Configure a DataTransferStep() to fetch new image data from public web portal, running on the cpu-compute compute target.	
Configure an EstimatorStep() to run an estimator that runs the bird_classifier_train.py model training script on the gpu_compute compute target.	
Configure a PythonScriptStep() to run both image_fetcher.py and image_resize.py on the cpu-compute compute target.	
Configure an EstimatorStep() to run an estimator that runs the bird_classifier_train.py model training script on the cpu_compute compute target.	
Configure a PythonScriptStep() to run image_fetcher.py on the cpu-compute compute target.	
Configure a PythonScriptStep() to run image_resize.py on the cpu-compute compute target.	
Configure a PythonScriptStep() to run bird_classifier_train.py on the cpu-compute compute target.	
Configure a PythonScriptStep() to run bird_classifier_train.py on the gpu-compute compute target.	

Answer:

Actions	Answer Area
	Configure a DataTransferStep() to fetch new image data from public web portal, running on the cpu-compute compute target.
Configure an EstimatorStep() to run an estimator that runs the bird_classifier_train.py model training script on the gpu_compute compute target.	Configure a PythonScriptStep() to run image_resize.py on the cpu-compute compute target.
Configure a PythonScriptStep() to run both image_fetcher.py and image_resize.py on the cpu-compute compute target.	Configure an EstimatorStep() to run an estimator that runs the bird_classifier_train.py model training script on the cpu_compute compute target.
Configure a PythonScriptStep() to run image_fetcher.py on the cpu-compute compute target.	
Configure a PythonScriptStep() to run bird_classifier_train.py on the cpu-compute compute target.	
Configure a PythonScriptStep() to run bird_classifier_train.py on the gpu-compute compute target.	

Explanation:
Step 1: Configure a DataTransferStep() to fetch new image data...
Step 2: Configure a PythonScriptStep() to run image_resize.y on the cpu-compute compute target.

Step 3: Configure the EstimatorStep() to run training script on the gpu_compute computer target.
The PyTorch estimator provides a simple way of launching a PyTorch training job on a compute target.
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-pytorch

QUESTION 170
Hotspot Question
You are a lead data scientist for a project that tracks the health and migration of birds. You create a multi- image classification deep learning model that uses a set of labeled bird photos collected by experts. You plan to use the model to develop a cross-platform mobile app that predicts the species of bird captured by app users.
You must test and deploy the trained model as a web service. The deployed model must meet the following requirements:
- An authenticated connection must not be required for testing.
- The deployed model must perform with low latency during inferencing.
- The REST endpoints must be scalable and should have a capacity to handle large number of requests when multiple end users are using the mobile application.
You need to verify that the web service returns predictions in the expected JSON format when a valid REST request is submitted.
Which compute resources should you use? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

Context	Resource
Test	ds-workstation notebook VM / aks-compute cluster / cpu-compute cluster / gpu-compute cluster
Production	ds-workstation notebook VM / aks-compute cluster / cpu-compute cluster / gpu-compute cluster

Answer:

Answer Area

Context	Resource
Test	**ds-workstation notebook VM** / aks-compute cluster / cpu-compute cluster / gpu-compute cluster
Production	ds-workstation notebook VM / aks-compute cluster / cpu-compute cluster / **gpu-compute cluster**

Explanation:

Box 1: ds-workstation notebook VM
An authenticated connection must not be required for testing. On a Microsoft Azure virtual machine (VM), including a Data Science Virtual Machine (DSVM), you create local user accounts while provisioning the VM. Users then authenticate to the VM by using these credentials.
Box 2: gpu-compute cluster
Image classification is well suited for GPU compute clusters
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/data-science-virtual-machine/dsvm-common-identity
https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/ai/training-deep-learning

QUESTION 171
Hotspot Question
You deploy a model in Azure Container Instance.
You must use the Azure Machine Learning SDK to call the model API.
You need to invoke the deployed model using native SDK classes and methods.
How should you complete the command? To answer, select the appropriate options in the answer areas.
NOTE: Each correct selection is worth one point.

Answer Area

```
from azureml.core import Workspace
[ from azureml.core.webservice import requests
  from azureml.core.webservice import Webservice
  from azureml.core.webservice import LocalWebservice ]
import json
ws = Workspace.from_config()
service_name = "mlmodel1-service"
service = Webservice(name=service_name, workspace=ws)
x_new = [[2,101.5,1,24,21], [1,89.7,4,41,21]]
input_json = json.dumps({"data": x_new})
[ predictions = service.run(input_json)
  predictions = requests.post(service.scoring_uri, input_json)
  predictions = service.deserialize(ws, input_json) ]
```

Answer:

Answer Area

```
from azureml.core import Workspace
```

[▼]
- from azureml.core.webservice import requests
- from azureml.core.webservice import Webservice
- from azureml.core.webservice import LocalWebservice

```
import json
ws = Workspace.from_config()
service_name = "mlmodel1-service"
service = Webservice(name=service_name, workspace=ws)
x_new = [[2,101.5,1,24,21], [1,89.7,4,41,21]]
input_json = json.dumps({"data": x_new})
```

[▼]
- predictions = service.run(input_json)
- predictions = requests.post(service.scoring_uri, input_json)
- predictions = service.deserialize(ws, input_json)

Explanation:
Box 1: from azureml.core.webservice import Webservice
The following code shows how to use the SDK to update the model, environment, and entry script for a web service to Azure Container Instances:
from azureml.core import Environment
from azureml.core.webservice import Webservice
from azureml.core.model import Model, InferenceConfig
Box 2: predictions = service.run(input_json)
Example: The following code demonstrates sending data to the service:
import json
test_sample = json.dumps({'data': [
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
]})
test_sample = bytes(test_sample, encoding='utf8')
prediction = service.run(input_data=test_sample)
print(prediction)
Reference:
https://docs.microsoft.com/bs-latn-ba/azure/machine-learning/how-to-deploy-azure-container-instance
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment

QUESTION 172
Drag and Drop Question
You plan to explore demographic data for home ownership in various cities. The data is in a CSV file with the following format:

```
age,city,income,home_owner
21,Chicago,50000,0
35,Seattle,120000,1
23,Seattle,65000,0
45,Seattle,130000,1
18,Chicago,48000,0
```

You need to run an experiment in your Azure Machine Learning workspace to explore the data and log the results. The experiment must log the following information:

```
- the number of observations in the dataset
- a box plot of income by home_owner
- a dictionary containing the city names and the average income for each city
```

You need to use the appropriate logging methods of the experiment's run object to log the required information.
How should you complete the code? To answer, drag the appropriate code segments to the correct locations. Each

code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.

Code segments

log
log_list
log_row
log_table
log_image

Answer Area

```
from azureml.core import Experiment, Run
import pandas as pd
import matplotlib.pyplot as plt
# Create an Azure ML experiment in workspace
experiment = Experiment(workspace = ws, name = "demo-experiment")
# Start logging data from the experiment
run = experiment.start_logging()
# load the dataset
data = pd.read_csv('research/demographics.csv')
# Log the number of observations
row_count = (len(data))
run. Segment ("observations", row_count)
# Log box plot for income by home_owner
fig = plt.figure(figsize=(9, 6))
ax = fig.gca()
data.boxplot(column = 'income', by = "home_owner", ax = ax)
ax.set_title('income by home_owner')
ax.set_ylabel('income')
run. Segment (name = 'income_by_home_owner', plot = fig)
# Create a dataframe of mean income per city
mean_inc_df = data.groupby('city')['income'].agg(np.mean).to_frame().reset_index()
# Convert to a dictionary
mean_inc_dict = mean_inc_df.to_dict('dict')
# Log city names and average income dictionary
run. Segment (name="mean_income_by_city", value= mean_inc_dict)
# Complete tracking and get link to details
run.complete()
```

Answer:

Code segments

log_list
log_row

Answer Area

```
from azureml.core import Experiment, Run
import pandas as pd
import matplotlib.pyplot as plt
# Create an Azure ML experiment in workspace
experiment = Experiment(workspace = ws, name = "demo-experiment")
# Start logging data from the experiment
run = experiment.start_logging()
# load the dataset
data = pd.read_csv('research/demographics.csv')
# Log the number of observations
row_count = (len(data))
run. log ("observations", row_count)
# Log box plot for income by home_owner
fig = plt.figure(figsize=(9, 6))
ax = fig.gca()
data.boxplot(column = 'income', by = "home_owner", ax = ax)
ax.set_title('income by home_owner')
ax.set_ylabel('income')
run. log_image (name = 'income_by_home_owner', plot = fig)
# Create a dataframe of mean income per city
mean_inc_df = data.groupby('city')['income'].agg(np.mean).to_frame().reset_index()
# Convert to a dictionary
mean_inc_dict = mean_inc_df.to_dict('dict')
# Log city names and average income dictionary
run. log_table (name="mean_income_by_city", value= mean_inc_dict)
# Complete tracking and get link to details
run.complete()
```

Explanation:
Box 1: log
The number of observations in the dataset.
run.log(name, value, description='')
Scalar values: Log a numerical or string value to the run with the given name. Logging a metric to a run causes that metric to be stored in the run record in the experiment. You can log the same metric multiple times within a run, the result being considered a vector of that metric.
Example: run.log("accuracy", 0.95)
Box 2: log_image
A box plot of income by home_owner.

log_image Log an image to the run record. Use log_image to log a .PNG image file or a matplotlib plot to the run. These images will be visible and comparable in the run record.
Example: run.log_image("ROC", plot=plt)
Box 3: log_table
A dictionary containing the city names and the average income for each city.
log_table: Log a dictionary object to the run with the given name.

QUESTION 173
Hotspot Question
You have a multi-class image classification deep learning model that uses a set of labeled photographs. You create the following code to select hyperparameter values when training the model.

```
from azureml.train.hyperdrive import BayesianParameterSampling
param_sampling = BayesianParametersSampling ({
 "learning_rate": uniform(0.01, 0.1),
 "batch_size": choice(16, 32, 64, 128)}
)
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
Hyperparameter combinations for the runs are selected based on how previous samples performed in the previous experiment run.	○	○
The learning rate value 0.09 might be used during model training.	○	○
You can define an early termination policy for this hyperparameter tuning run.	○	○

Answer:

Answer Area

	Yes	No
Hyperparameter combinations for the runs are selected based on how previous samples performed in the previous experiment run.	●	○
The learning rate value 0.09 might be used during model training.	●	○
You can define an early termination policy for this hyperparameter tuning run.	○	●

Explanation:
Box 1: Yes
Hyperparameters are adjustable parameters you choose to train a model that govern the training process itself. Azure Machine Learning allows you to automate hyperparameter exploration in an efficient manner, saving you significant time and resources. You specify the range of hyperparameter values and a maximum number of training runs. The system then automatically launches multiple simultaneous runs with different parameter configurations and finds the configuration that results in the best performance, measured by the metric you choose. Poorly performing training runs are automatically early terminated, reducing wastage of compute resources. These resources are instead used to explore other hyperparameter configurations.
Box 2: Yes
uniform(low, high) - Returns a value uniformly distributed between low and high
Box 3: No
Bayesian sampling does not currently support any early termination policy.
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters