DUMPSARENA

Designing and Implementing a Data Science Solution on Azure

Microsoft DP-100

Version Demo

Total Demo Questions: 48

Total Premium Questions: 487

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Design and prepare a machine learning solution	100
Topic 2, Explore data and train models	284
Topic 3, Prepare a model for deployment	17
Topic 4, Deploy and retrain a model	64
Topic 5, Case Study 1	11
Topic 6, Case Study 2	11
Total	487

QUESTION NO: 1

You use the Azure Machine Learning Python SDK to define a pipeline that consists of multiple steps.

When you run the pipeline, you observe that some steps do not run. The cached output from a previous run is used instead.

You need to ensure that every step in the pipeline is run, even if the parameters and contents of the source directory have not changed since the previous run.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Use a PipelineData object that references a datastore other than the default datastore.

B. Set the regenerate_outputs property of the pipeline to True.

C. Set the allow_reuse property of each step in the pipeline to False.

D. Restart the compute cluster where the pipeline experiment is configured to run.

E. Set the outputs property of each step in the pipeline to True.

ANSWER: B C

Explanation:

Set the regenerate_outputs property of the pipeline to True is correct because Azure Machine Learning SDK v1 supports forcing all steps in a submitted pipeline run to regenerate their outputs. When regenerate_outputs is enabled at submission time, Azure ML does not reuse previously cached step outputs for that run, so each step is executed again even when its inputs, parameters, and source snapshot appear unchanged. This is useful when you want a full rerun of the pipeline without editing each individual step. See the Pipeline submit parameter documentation in Microsoft Learn.

Set the allow_reuse property of each step in the pipeline to False is also correct because step reuse is controlled at the step level. Azure ML pipeline steps such as PythonScriptStep include an allow_reuse setting; when it is disabled, the service treats the step as non-reusable and runs it instead of using results from a previous run. Applying this setting to every step ensures the whole pipeline executes. Microsoft documents this behavior for pipeline step classes such as PythonScriptStep.

QUESTION NO: 2

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You plan to use a Python script to run an Azure Machine Learning experiment. The script creates a reference to the experiment run context, loads data from a file, identifies the set of unique values for the label column, and completes the experiment run:

```
from azureml.core import Run

import pandas as pd

run = Run.get_context()

data = pd.read_csv( ' data.csv ' )

label_vals = data[ ' label ' ].unique()

# Add code to record metrics here

run.complete()
```

The experiment must record the unique labels in the data as metrics for the run that can be reviewed later.

You must add code to the script to record the unique label values as run metrics at the point indicated by the comment.

Solution: Replace the comment with the following code:

run.upload_file(' outputs/labels.csv ' , ' ./data.csv ')

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

The solution does not meet the goal because `run.upload_file()` uploads a local file as a run artifact; it does not record values as run metrics. The requirement is specifically to record the unique label values as metrics that can be reviewed later in the experiment run history. In Azure Machine Learning SDK v1, run metrics are logged by using methods such as `run.log()`, `run.log_list()`, `run.log_table()`, or related logging APIs on the `Run` object. Since `label_vals` contains a collection of unique values, the appropriate pattern would be to log the collection as a list metric, for example by using `run.log_list()` after converting the values to a supported list format if needed. Uploading `./data.csv` to `outputs/labels.csv` would only make the file available as an output artifact for the run, and it would not create a metric entry for the unique labels. Microsoft documents the metric logging methods on the Azure ML Run class and describes how metrics can be logged and reviewed in Azure Machine Learning experiments in Log and view metrics.

QUESTION NO: 3 - (HOTSPOT)

HOTSPOT

You are using the Azure Machine Learning Service to automate hyperparameter exploration of your neural network classification model.

You must define the hyperparameter space to automatically tune hyperparameters using random sampling according to following requirements:

- The learning rate must be selected from a normal distribution with a mean value of 10 and a standard deviation of 3.
- Batch size must be 16, 32 and 64.
- Keep probability must be a value selected from a uniform distribution between the range of 0.05 and 0.1.

You need to use the param_sampling method of the Python API for the Azure Machine Learning Service.

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.

Hot Area:

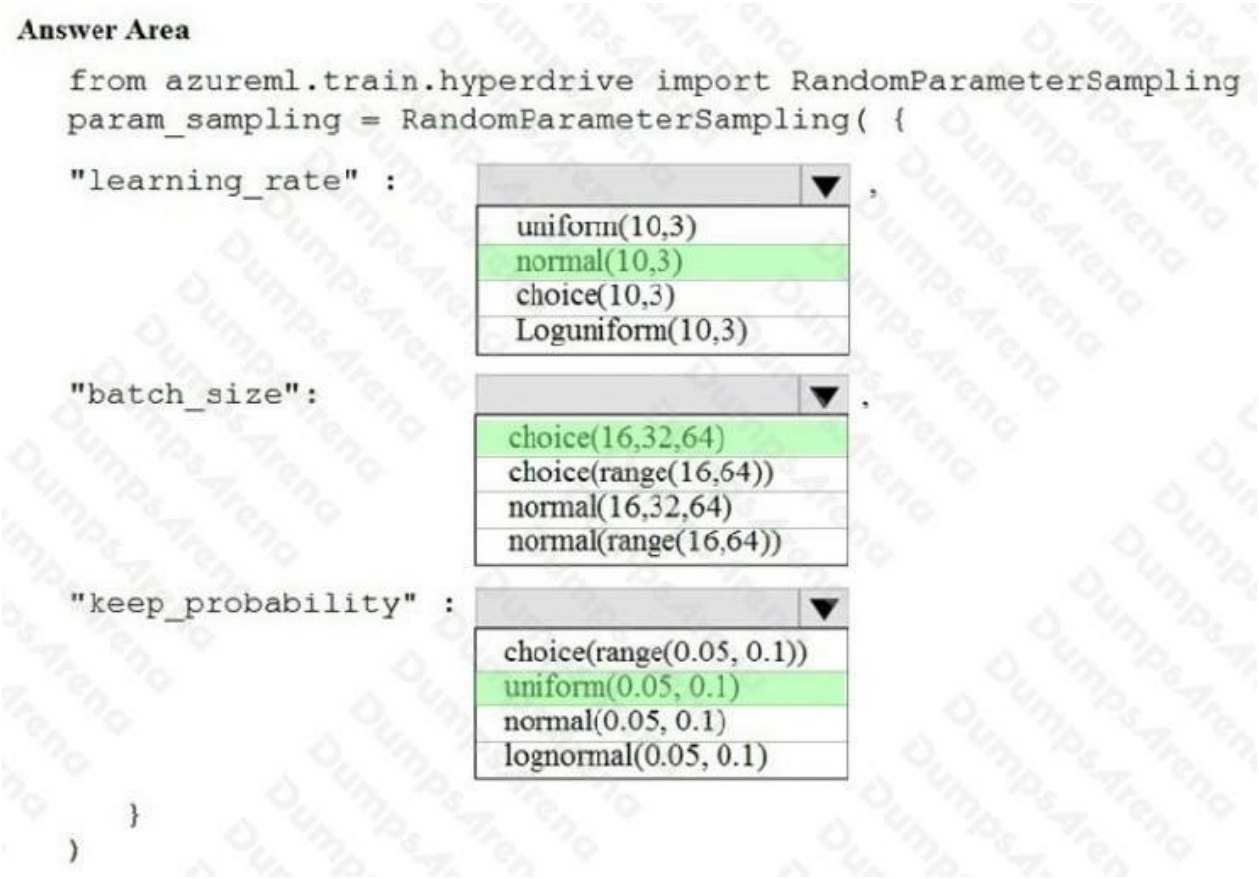
Answer Area

```
from azureml.train.hyperdrive import RandomParameterSampling
param_sampling = RandomParameterSampling( {
"learning_rate" :  [                    ] ,
"batch_size":      [                    ] ,
"keep_probability" : [                  ]
    }
)
```

- "learning_rate" : uniform(10,3) / normal(10,3) / choice(10,3) / Loguniform(10,3)
- "batch_size": choice(16,32,64) / choice(range(16,64)) / normal(16,32,64) / normal(range(16,64))
- "keep_probability" : choice(range(0.05, 0.1)) / uniform(0.05, 0.1) / normal(0.05, 0.1) / lognormal(0.05, 0.1)

ANSWER:

Answer Area



Explanation:

The correct code uses RandomParameterSampling with the Azure Machine Learning HyperDrive parameter expression that matches each required search behavior. The learning rate requirement says the value must be selected from a normal distribution with a mean of 10 and a standard deviation of 3, so the correct expression is normal(10,3). In Azure Machine Learning hyperparameter tuning, normal takes the mean and standard deviation as its arguments, making it the direct match for that requirement.

The batch size requirement is different because it is not asking for a continuous distribution. It says the batch size must be one of the specific values 16, 32, and 64. For this kind of discrete set, the correct HyperDrive expression is choice(16,32,64). This tells random sampling to select one value from the provided list for each trial, rather than generating a numeric value across a range.

The keep probability requirement says the value must be selected from a uniform distribution between 0.05 and 0.1. The correct expression is uniform(0.05, 0.1), because Azure Machine Learning uses uniform to sample floating-point values evenly from the specified lower and upper bounds. Together, these definitions create a valid random hyperparameter search space containing both discrete and continuous parameters, which is exactly what RandomParameterSampling is designed to support. Microsoft's HyperDrive documentation describes these parameter expressions and their use in automated hyperparameter tuning: Tune hyperparameters for your model with Azure Machine Learning and azureml.train.hyperdrive Python API reference.

QUESTION NO: 4

You use the Two-Class Neural Network module in Azure Machine Learning Studio to build a binary

classification model. You use the Tune Model Hyperparameters module to tune accuracy for the model.

You need to select the hyperparameters that should be tuned using the Tune Model Hyperparameters module.

Which two hyperparameters should you use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Number of hidden nodes

B. Learning Rate

C. The type of the normalizer

D. Number of learning iterations

E. Hidden layer specification

ANSWER: D E

Explanation:

Number of learning iterations and Hidden layer specification are the correct selections for tuning the Two-Class Neural Network model in Azure Machine Learning Studio. Number of learning iterations controls the maximum number of times the neural network training algorithm processes the training data, so tuning it can directly affect convergence and classification accuracy. Hidden layer specification controls the neural network architecture created between the input and output layers, which is a core design choice for neural network performance. When used with the Tune Model Hyperparameters module, Azure Machine Learning evaluates combinations of configured trainer parameters and selects the model that performs best according to the chosen metric, such as accuracy. Microsoft's module documentation identifies these settings as configurable parameters for the Two-Class Neural Network trainer, and the hyperparameter tuning module is designed to sweep trainer parameter values to optimize model performance. See the Microsoft documentation for the Two-Class Neural Network module and Tune Model Hyperparameters for details.

QUESTION NO: 5

You plan to use the Hyperdrive feature of Azure Machine Learning to determine the optimal hyperparameter values when training a model.

You must use Hyperdrive to try combinations of the following hyperparameter values:

• learning_rate: any value between 0.001 and 0.1

• batch_size: 16, 32, or 64

You need to configure the search space for the Hyperdrive experiment.

Which two parameter expressions should you use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. a choice expression for learning_rate

B. a uniform expression for learning_rate

C. a normal expression for batch_size

D. a choice expression for batch_size

E. a uniform expression for batch_size

ANSWER: B D

Explanation:

In Azure Machine Learning HyperDrive, the search space should match the type of values that each hyperparameter can take. For learning_rate, the requirement says HyperDrive can try any value between 0.001 and 0.1. That is a continuous numeric interval, so a uniform expression for learning_rate is appropriate because it samples values from a continuous range with equal probability across the specified lower and upper bounds.

For batch_size, the requirement lists only three allowed values: 16, 32, or 64. That is a discrete set rather than a continuous range, so a choice expression for batch_size is appropriate. The choice expression tells HyperDrive to select from the explicitly provided values during hyperparameter sampling.

Microsoft's Azure Machine Learning documentation describes continuous hyperparameters as being defined by distributions such as uniform, and discrete hyperparameters as being defined by choice values. See Tune hyperparameters for your model with Azure Machine Learning and the HyperDrive parameter expression reference at azureml.train.hyperdrive.parameter_expressions.

QUESTION NO: 6

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are creating a model to predict the price of a student's artwork depending on the following variables: the student's length of education, degree type, and art form.

You start by creating a linear regression model.

You need to evaluate the linear regression model.

Solution: Use the following metrics: Relative Squared Error, Coefficient of Determination, Accuracy, Precision, Recall, F1 score, and AUC.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because the proposed evaluation set mixes regression and classification metrics, so it is not an appropriate way to evaluate a linear regression model for predicting artwork price. Price is a continuous numeric target, and linear regression models should be assessed with regression metrics that measure prediction error or goodness of fit, such as mean absolute error, root mean squared error, relative squared error, or coefficient of determination, also known as R-squared. Microsoft's Azure Machine Learning documentation lists different metrics for regression and classification tasks; regression uses metrics such as normalized errors and R2 score, while classification uses metrics such as accuracy, precision, recall, F1 score, and AUC. Since the solution specifically includes classification metrics as required evaluation metrics for a regression problem, it does not fully meet the stated goal. See Microsoft's guidance on regression metrics in Azure Machine Learning at Regression/forecasting metrics and the Azure Machine Learning Evaluate Model component reference at Evaluate Model.

QUESTION NO: 7

You have been tasked with ascertaining if two sets of data differ considerably. You will make use of Azure Machine Learning Studio to complete your task.

You plan to perform a paired t-test.

Which of the following are conditions that must apply to use a paired t-test? (Choose all that apply.)

A. All scores are independent from each other.

B. You have a matched pairs of scores.

C. The sampling distribution of d is normal.

D. The sampling distribution of x1- x2 is normal.

ANSWER: B C

Explanation:

A paired t-test is appropriate when the two measurements are naturally linked, such as before-and-after measurements on the same subject or results from matched subjects. Therefore, “You have a matched pairs of scores.” is correct because the test analyzes each pair as a single unit by calculating a within-pair difference. The hypothesis test is then performed on those differences rather than on two unrelated samples. “The sampling distribution of d is normal.” is also correct because the paired t-test assumes that the distribution of the paired differences, often represented as d, is approximately normal, especially when the sample size is small. In Azure Machine Learning Studio’s classic t-test module documentation, the paired t-test is described as requiring matched pairs and a normal sampling distribution of the differences. This aligns with standard statistical guidance for paired-sample t-tests, where the key data analyzed is the set of differences between paired observations. For more detail, see Microsoft’s documentation for the Test Hypothesis Using t-Test component and the general paired t-test assumptions described by NIST.

QUESTION NO: 8

You are developing a hands-on workshop to introduce Docker for Windows to attendees.

You need to ensure that workshop attendees can install Docker on their devices.

Which two prerequisite components should attendees install on the devices? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Microsoft Hardware-Assisted Virtualization Detection Tool

B. Kitematic

C. BIOS-enabled virtualization

D. VirtualBox

E. Windows 10 64-bit Professional

ANSWER: C E

Explanation:

BIOS-enabled virtualization is correct because Docker for Windows relies on a virtualization backend to run Linux containers on a Windows host. The processor must support hardware virtualization, and that capability must be enabled in the device firmware settings, typically BIOS or UEFI. Without enabled hardware virtualization, Docker Desktop cannot start the required virtualized environment reliably.

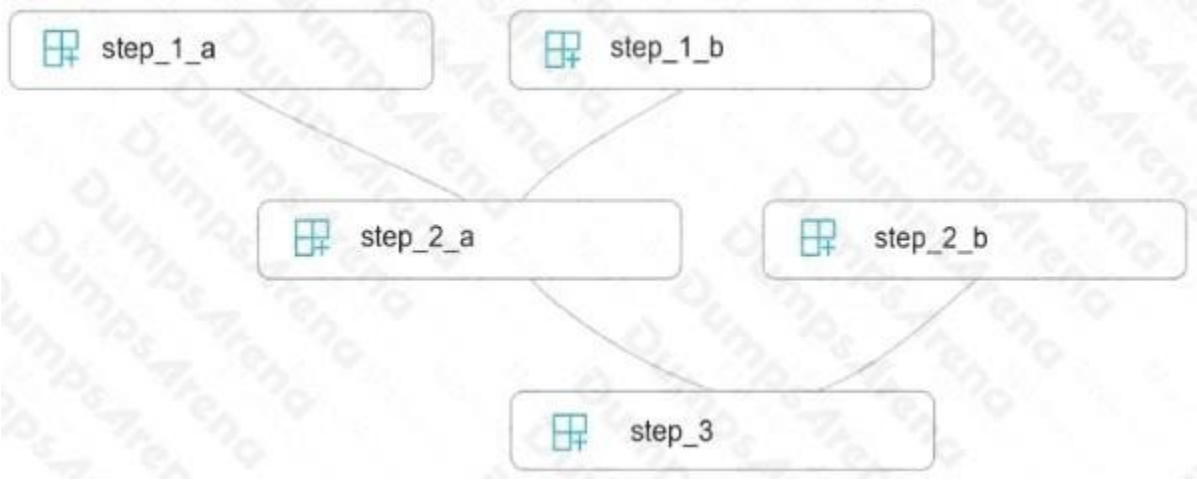
Windows 10 64-bit Professional is also correct for the classic Docker for Windows requirements because Docker Desktop on Windows was designed to use Hyper-V, which requires a 64-bit Windows edition that includes Hyper-V support, such as Windows 10 Professional, Enterprise, or Education. In a workshop setting, requiring Windows 10 64-bit Professional ensures attendees have an operating system edition capable of enabling the virtualization features Docker needs. Microsoft documents Hyper-V as requiring a 64-bit processor with Second Level Address Translation and VM Monitor Mode extensions, and Docker’s Windows installation guidance also identifies virtualization support as a key installation requirement. See Hyper-V requirements on Windows and Docker Desktop for Windows installation requirements.

QUESTION NO: 9 - (SIMULATION)

SIMULATION

You write five Python scripts that must be processed in the order specified in Exhibit A – which allows the same modules to run in parallel, but will wait for modules with dependencies.

You must create an Azure Machine Learning pipeline using the Python SDK, because you want to script to create the pipeline to be tracked in your version control system. You have created five PythonScriptSteps and have named the variables to match the module names.



You need to create the pipeline shown. Assume all relevant imports have been done.

Which Python code segment should you use?

A.
```
p = Pipeline(ws, steps=[[[[step_1_a, step_1_b], step_2_a],  step_2_b], step_3])
```

B.
```
pipeline_steps = {
    "Pipeline": {
        "run": step_3,
        "run_after": {[
            {"run": step_2_a,
                "run_after":
                    [{"run": step_1_a},
                     {"run": step_1_b}]
                },
            {"run": step_2_b}]
        }
    }
}
p = Pipeline(ws, steps=pipeline_steps)
```

C.
```
step_2_a.run_after(step_1_b)
step_2_a.run_after(step_1_a)
step_3.run_after(step_2_b)
step_3.run_after(step_2_a)
p = Pipeline(ws, steps=[step_3])
```

D.
```
p = Pipeline(ws, steps=[step_1_a, step_1_b, step_2_a, step_2_b, step_3])
```

ANSWER: See the explanation for the answer

Explanation:

The steps parameter is an array of steps. To build pipelines that have multiple steps, place the steps in order in this array.

QUESTION NO: 10

You need to resolve the local machine learning pipeline performance issue. What should you do?

A. Increase Graphic Processing Units (GPUs).

B. Increase the learning rate.

C. Increase the training iterations,

D. Increase Central Processing Units (CPUs).

ANSWER: A

Explanation:

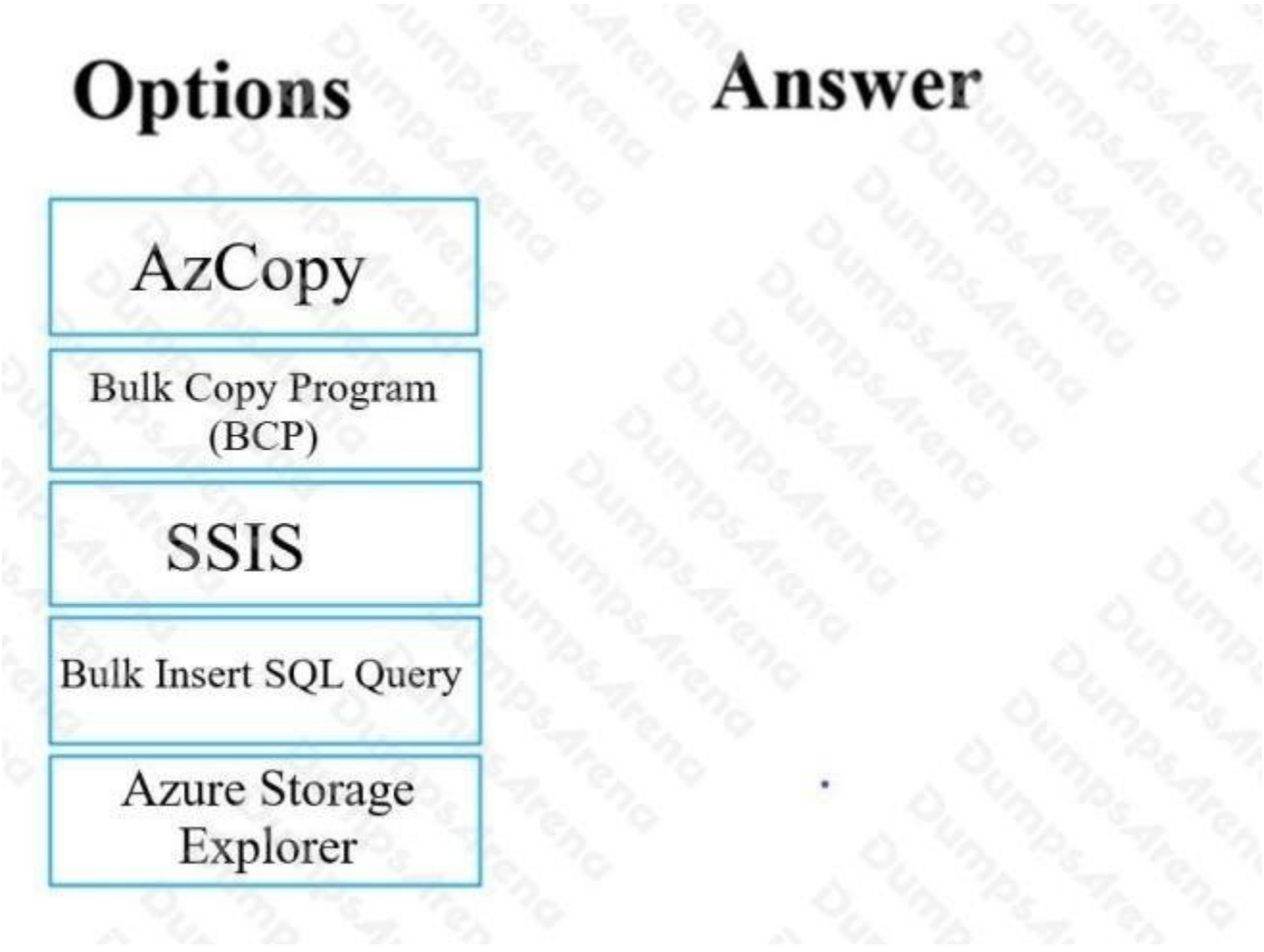
Increase Graphic Processing Units (GPUs) is correct because GPU acceleration is the appropriate way to improve performance for compute-intensive machine learning training pipelines, especially when the workload involves deep learning, tensor operations, or large-scale matrix calculations. GPUs are designed to perform many mathematical operations in parallel, which can significantly reduce training time compared with relying only on general-purpose processor resources. In Azure Machine Learning, Microsoft supports using GPU-enabled compute targets for training workloads that benefit from hardware acceleration, and this is a common remediation when local training performance is insufficient. Rather than changing model behavior, increasing GPU capacity addresses the underlying compute bottleneck and helps the pipeline complete faster while preserving the intended training logic. Microsoft's Azure Machine Learning documentation describes using GPU compute for training scenarios and selecting appropriate compute targets for machine learning workloads. See Train with GPUs in Azure Machine Learning and Azure Machine Learning compute targets.

QUESTION NO: 11 - (DRAG DROP)

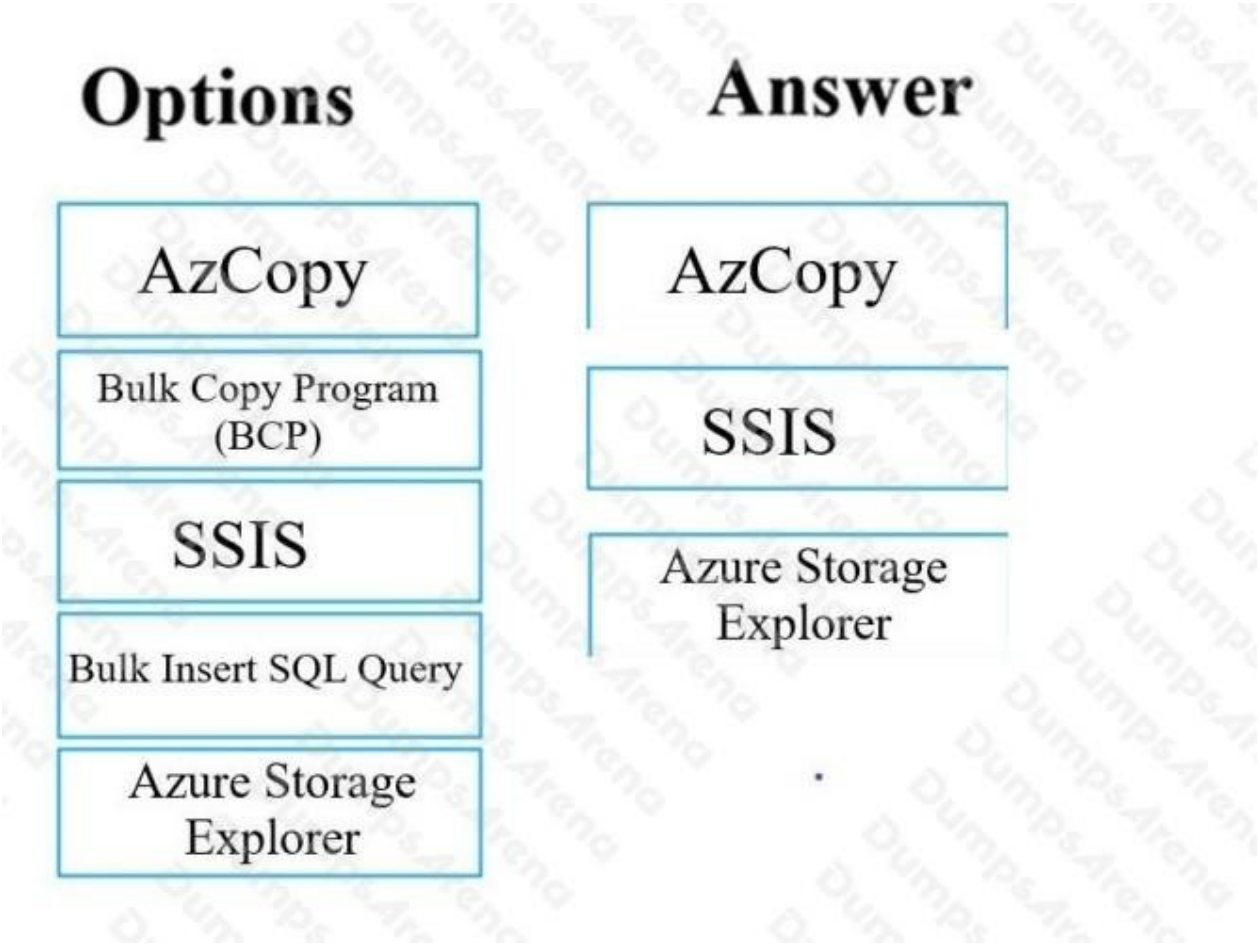
You have been tasked with moving data into Azure Blob Storage for the purpose of supporting Azure Machine Learning.

Which of the following can be used to complete your task? Answer by dragging the correct options from the list to the answer area.

Select and Place:



ANSWER:



Explanation:

The correct selections are AzCopy, SSIS, and Azure Storage Explorer because each can be used to move or upload data into Azure Blob Storage, which is commonly used as a storage location for Azure Machine Learning datasets and data assets. Azure Machine Learning can read data from Azure Storage accounts, including Blob Storage, through registered datastores or data assets, so the key requirement is to use a tool that can place files into a blob container reliably.

AzCopy is a Microsoft command-line utility built specifically for copying data to, from, and between Azure Storage locations. It is well suited for scripted or large-scale uploads because it supports high-performance transfers and automation. Microsoft documents AzCopy as a supported tool for copying blobs and files in Azure Storage: Get started with AzCopy.

Azure Storage Explorer is also appropriate because it provides a graphical interface for managing Azure Storage resources. It can upload local files and folders directly into Blob Storage containers, which makes it useful when users want an interactive tool rather than a command-line workflow. Microsoft describes this capability in the Azure Storage Explorer documentation: Manage Azure storage resources with Storage Explorer.

SSIS is valid as well because SQL Server Integration Services can move data as part of an ETL pipeline, and Microsoft provides Azure-related SSIS capabilities for working with Azure storage services. This makes it suitable when data movement into Blob Storage is part of a broader integration workflow. See Microsoft's Azure Feature Pack for Integration Services documentation: Azure Feature Pack for Integration Services.

QUESTION NO: 12

You plan to provision an Azure Machine Learning Basic edition workspace for a data science project.

You need to identify the tasks you will be able to perform in the workspace.

Which three tasks will you be able to perform? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

D

A. Create a Compute Instance and use it to run code in Jupyter notebooks.

B. Create an Azure Kubernetes Service (AKS) inference cluster.

C. Use the designer to train a model by dragging and dropping pre-defined modules.

D. Create a tabular dataset that supports versioning.

E. Use the Automated Machine Learning user interface to train a model.

ANSWER: A B D

Explanation:

In an Azure Machine Learning Basic edition workspace, you can use code-first capabilities for developing, running, and operationalizing machine learning solutions. Create a Compute Instance and use it to run code in Jupyter notebooks is correct because compute instances are managed cloud-based development environments integrated with Azure Machine Learning and intended for interactive notebook authoring and experimentation. Create an Azure Kubernetes Service (AKS) inference cluster is also correct because Azure Machine Learning supports deploying trained models to managed online inference targets such as AKS for production-scale inferencing. Create a tabular dataset that supports versioning is correct because Azure Machine Learning datasets can be registered in the workspace, referenced by name and version, and used consistently across experiments and training runs. These capabilities align with the Basic edition's focus on core workspace, compute, dataset, experiment, and deployment functionality, while more visual or no-code experiences were associated with higher-tier workspace features in the older Basic/Enterprise model. For more information, see the Azure Machine Learning pricing details at Azure Machine Learning pricing and Microsoft's dataset registration guidance at Create and register Azure Machine Learning datasets.

QUESTION NO: 13

You manage an Azure Machine Learning workspace. The Pylhon scrip! named scriptpy reads an argument named training_data. The trainlng.data argument specifies the path to the training data in a file named datasetl.csv.

You plan to run the scriptpy Python script as a command job that trains a machine learning model.

You need to provide the command to pass the path for the datasct as a parameter value when you submit the script as a training job.

Solution: python train.py --training_data training_data

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because the command shown passes the literal string `training_data` to the Python argument, rather than passing the resolved Azure Machine Learning input path. In an Azure Machine Learning command job, data inputs should be declared as job inputs and then referenced in the command by using the expression syntax, such as `${{inputs.training_data}}`. At runtime, Azure Machine Learning resolves that input to the actual mounted or downloaded path that the script can read. A valid command would typically look like `python train.py --training_data ${{inputs.training_data}}`, with `training_data` also defined in the job's `inputs` section as a URI file, URI folder, MLTable, or registered data asset. This is the documented pattern for passing data into command jobs in Azure Machine Learning v2. See Microsoft's documentation on training models with command jobs and reading and writing data in jobs.

QUESTION NO: 14

You plan to use the Hyperdrive feature of Azure Machine Learning to determine the optimal hyperparameter values when training a model.

You must use Hyperdrive to try combinations of the following hyperparameter values. You must not apply an early termination policy.

learning_rate: any value between 0.001 and 0.1

• batch_size: 16, 32, or 64

You need to configure the sampling method for the Hyperdrive experiment

Which two sampling methods can you use? Each correct answer is a complete solution.

NOTE: Each correct selection is worth one point.

A. Grid sampling

B. No sampling

C. Bayesian sampling

D. Random sampling

ANSWER: C D

Explanation:

Bayesian sampling and Random sampling are correct because both can be used with a mixed hyperparameter search space that includes a continuous value for learning_rate and a discrete set of values for batch_size. In Azure Machine Learning HyperDrive, Random sampling selects hyperparameter values randomly from the defined search space and supports both continuous distributions, such as uniform ranges, and discrete choices. This fits a learning_rate defined as any value between 0.001 and 0.1 together with batch_size values of 16, 32, or 64.

Bayesian sampling is also valid because it can sample from supported continuous and discrete parameter expressions and uses previous trial results to choose promising values for subsequent runs. It is especially useful when the goal is to optimize a primary metric efficiently rather than exhaustively evaluate every possible value. The requirement not to apply an early termination policy is compatible with Bayesian sampling, since Azure Machine Learning does not support early termination policies with Bayesian sampling. Microsoft documents the available HyperDrive sampling methods and their supported parameter expressions in the Azure Machine Learning hyperparameter tuning guidance: Tune hyperparameters and the SDK reference for azureml.train.hyperdrive.

QUESTION NO: 15

You manage an Azure Machine Learning workspace.

An MLflow model is already registered. You plan to customize how the deployment does inference. You need to deploy the MLflow model to a batch endpoint for batch inferencing. What should you create first?

A. scoring script

B. deployment

C. environment

D. deployment definition

ANSWER: A

Explanation:

A scoring script is correct because customizing inference behavior for an Azure Machine Learning batch deployment requires you to provide custom inference logic. For batch endpoints, the scoring script is where you define how the

registered model is loaded and how each mini-batch of input data is processed. In Azure Machine Learning, this is typically done by implementing the required `init()` function for initialization and the `run(mini_batch)` function for batch inference. When an MLflow model is deployed without customization, Azure Machine Learning can often generate the inference behavior automatically from the MLflow model flavor and dependencies. However, the scenario explicitly says you plan to customize how the deployment does inference, so the first artifact to create is the scoring script that contains that custom logic. After that script exists, it can be referenced when configuring the batch deployment. Microsoft documents this customization pattern for batch deployments and MLflow models in Azure Machine Learning. See Use batch model deployments and Deploy MLflow models to batch endpoints.

QUESTION NO: 16

You are analyzing a dataset by using Azure Machine Learning Studio.

You need to generate a statistical summary that contains the p-value and the unique count for each feature column.

Which two modules can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Computer Linear Correlation

B. Export Count Table

C. Execute Python Script

D. Convert to Indicator Values

E. Summarize Data

ANSWER: B E

Explanation:

Export Count Table is correct because it is designed to convert a count table into a tabular dataset that can be inspected or used downstream. Count tables produced by count-based feature engineering modules include statistics about feature values, including measures such as p-values and unique counts, so exporting the count table provides the required statistical summary in a readable table format. Microsoft documents this module as the way to view or export count-table information generated by count featurization workflows: Export Count Table.

Summarize Data is also correct because it directly computes descriptive statistics for dataset columns in Azure Machine Learning Studio. It returns a row of summary statistics for each variable or feature column, including useful profiling metrics such as missing-value counts, unique-value counts, and other statistical scores used to understand the distribution and characteristics of the dataset. This makes it a direct fit when the goal is to generate a statistical summary for each feature column. Microsoft describes this profiling behavior in the module reference: Summarize Data.

QUESTION NO: 17

You are solving a classification task.

You must evaluate your model on a limited data sample by using k-fold cross-validation. You start by configuring a k parameter as the number of splits.

You need to configure the k parameter for the cross-validation.

Which value should you use?

A. k=1

B. k=10

C. k=0.5

D. k=0.9

ANSWER: B

Explanation:

k=10 is correct because the k parameter in k-fold cross-validation represents the number of folds, or splits, into which the dataset is divided for repeated training and validation. With limited data, 10-fold cross-validation is a widely used practical choice because it lets each observation be used for validation once while using most of the available data for training in each iteration. This usually gives a more reliable estimate of model performance than a single train/test split, while avoiding the computational overhead and higher variance that can come with leave-one-out cross-validation on every individual record.

Microsoft's Azure Machine Learning documentation for the Cross Validate Model component describes configuring cross-validation by specifying the number of folds, with 10 folds commonly used as the default setting for this type of evaluation. This aligns with standard machine learning practice for classification model evaluation when the available sample is limited. See Microsoft's Cross Validate Model component documentation and the Azure Machine Learning overview of designer components and pipelines.

QUESTION NO: 18

You run an experiment that uses an AutoMLConfig class to define an automated machine learning task with a maximum of ten model training iterations. The task will attempt to find the best performing model based on a metric named accuracy.

You submit the experiment with the following code:

```
from azureml.core.experiment import Experiment
automl_experiment = Experiment(ws, 'automl_experiment')
automl_run = automl_experiment.submit(automl_config, show_output=True)
```

You need to create Python code that returns the best model that is generated by the automated machine learning task.

Which code segment should you use?

A. best_model = automl_run.get_details()

B. best_model = automl_run.get_metrics()

C. best_model = automl_run.get_file_names()[1]

D. best_model = automl_run.get_output()[1]

ANSWER: D

Explanation:

The correct code is best_model = automl_run.get_output()[1] because an Azure Machine Learning automated ML experiment run exposes the get_output method to retrieve the best child run and the fitted model produced by the AutoML job. When called without specifying a particular iteration or metric, get_output returns the best run according to the primary metric configured for the AutoML task, which in this scenario is accuracy. The method returns a tuple: the first item is the Run object for the selected best run, and the second item is the trained model object. Since the requirement is to return the best model itself, indexing the returned tuple with [1] selects the fitted model rather than the run metadata. This is the standard SDK pattern used after submitting and completing an AutoML run in Azure Machine Learning SDK v1. Microsoft's SDK documentation for AutoMLRun.get_output describes this behavior and the returned values. You can review the method in the AutoMLRun class documentation and the AutoML configuration workflow in Microsoft Learn.

QUESTION NO: 19

You have an Azure Machine Learning workspace. You build a deep learning model.

You need to publish a GPU-enabled model as a web service.

Which two compute targets can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Azure Kubernetes Service (AKS)

B. Azure Container Instances (ACI)

C. Local web service

D. Azure Machine Learning compute clusters

ANSWER: A C

Explanation:

For a GPU-backed deployment in Azure Machine Learning, Azure Kubernetes Service (AKS) is the supported Azure-hosted compute target for production-style real-time inference. AKS can be configured with GPU-capable node pools, allowing a deep learning scoring container to use GPU resources when serving requests. This makes Azure Kubernetes Service (AKS) the appropriate scalable web-service target for a GPU-enabled model. Local web service is also a valid compute target for this scenario when you need to run and validate the GPU-enabled scoring service locally, assuming the local Docker environment is configured with the required GPU support. Azure Machine Learning's local deployment workflow creates a Docker-based web service on the local machine, which is commonly used to test the inference script, model files, dependencies, and container behavior before moving the service to an Azure-hosted target. These deployment patterns are covered in Microsoft's Azure Machine Learning deployment guidance and local container deployment documentation: Deploy models and choose compute targets and Deploy a model locally.

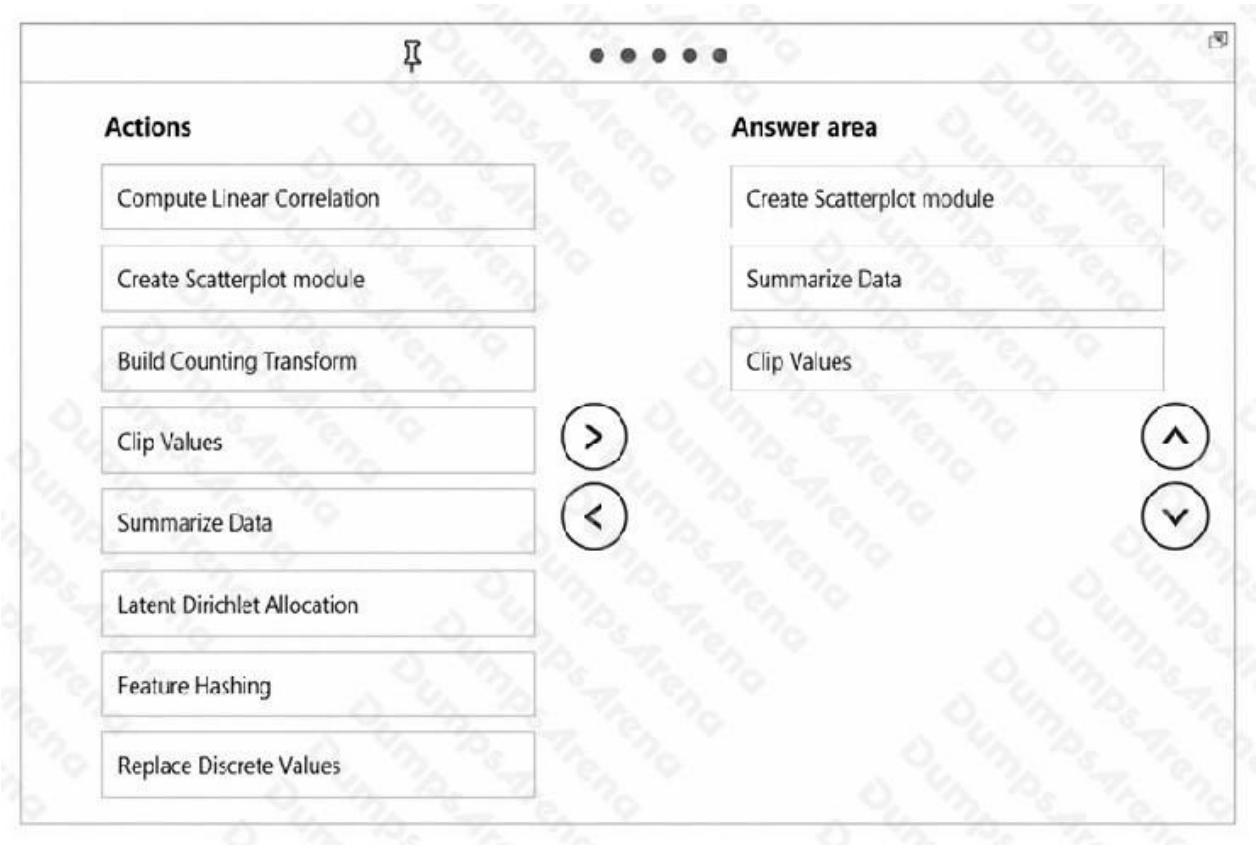
QUESTION NO: 20 - (DRAG DROP)

DRAG DROP

You need to visually identify whether outliers exist in the Age column and quantify the outliers before the outliers are removed.

Which three Azure Machine Learning Studio modules should you use in sequence? To answer, move the appropriate modules from the list of modules to the answer area and arrange them in the correct order.

ANSWER:



Explanation:

The correct sequence is to use **Create Scatterplot module**, then **Summarize Data**, and then **Clip Values**. The goal starts with visual identification, so a scatterplot is the right first step because it lets you inspect the Age column graphically and

quickly see whether unusual values appear far away from the main concentration of records. This is the most direct way to visually detect potential outliers before making any changes to the dataset.

After the visual check, **Summarize Data** is the appropriate next step because it produces descriptive statistics for the selected column. Those statistics help quantify the Age distribution before any values are altered. This is important because once outliers are clipped or replaced, the original minimum, maximum, mean, standard deviation, and related distribution information may change. Capturing the summary first preserves the evidence needed to understand how extreme the values were.

Finally, **Clip Values** is used once the outliers have been identified and quantified. In Azure Machine Learning Studio, this module is intended to handle values outside a defined threshold by clipping or replacing them, which is a common preprocessing step for outlier treatment. Microsoft documents these classic Azure ML Studio modules in the module reference, including Create Scatterplot, Summarize Data, and Clip Values. This order preserves the original data for inspection and measurement before applying the transformation that removes or caps the outliers.

QUESTION NO: 21

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are using Azure Machine Learning to run an experiment that trains a classification model.

You want to use Hyperdrive to find parameters that optimize the AUC metric for the model. You configure a HyperDriveConfig for the experiment by running the following code:

```
hyperdrive = HyperDriveConfig(estimator=your_estimator,
 hyperparameter_sampling=your_params,
 policy=policy,
 primary_metric_name='AUC',
 primary_metric_goal=PrimaryMetricGoal.MAXIMIZE,
 max_total_runs=6,
 max_concurrent_runs=4)
```

variable named y_test variable, and the predicted probabilities from the model are stored in a variable named y_predicted. You need to add logging to the script to allow Hyperdrive to optimize hyperparameters for the AUC metric. Solution: Run the following code:

```
from sklearn.metrics import roc_auc_score
import logging
# code to train model omitted
auc = roc_auc_score(y_test, y_predicted)
logging.info("AUC: " + str(auc))
```

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because HyperDrive can optimize only a metric that is logged as an Azure Machine Learning run metric with the same name specified in the HyperDriveConfig primary_metric_name, such as AUC. Writing the AUC value with ordinary

Python logging, console output, or informational log messages does not create a tracked run metric that HyperDrive can evaluate when comparing child runs. The training script should compute the AUC, for example by using roc_auc_score against y_test and y_predicted, and then log it to the run history, such as with Run.get_context().log("AUC", auc) in the Azure ML SDK, or with an equivalent MLflow metric logging call in supported workflows. HyperDrive reads those recorded metrics from each child run and uses the configured primary metric goal, typically maximize for AUC, to select the best hyperparameter configuration. Microsoft's HyperDrive guidance states that the primary metric must be logged by the training script, and Azure ML run logging documentation describes using run.log to record scalar metrics for experiments. See Tune hyperparameters for your model with Azure Machine Learning and Run.log method.

QUESTION NO: 22

You plan to run a script as an experiment using a Script Run Configuration. The script uses modules from the scipy library as well as several Python packages that are not typically installed in a default conda environment

You plan to run the experiment on your local workstation for small datasets and scale out the experiment by running it on more powerful remote compute clusters for larger datasets.

You need to ensure that the experiment runs successfully on local and remote compute with the least administrative effort.

What should you do?

A. Create and register an Environment that includes the required packages. Use this Environment for all experiment runs.

B. Always run the experiment with an Estimator by using the default packages.

C. Do not specify an environment in the run configuration for the experiment. Run the experiment by using the default environment.

D. Create a config.yaml file defining the conda packages that are required and save the file in the experiment folder.

E. Create a virtual machine (VM) with the required Python configuration and attach the VM as a compute target. Use this compute target for all experiment runs.

ANSWER: A

Explanation:

Create and register an Environment that includes the required packages. Use this Environment for all experiment runs. is correct because an Azure Machine Learning environment is designed to capture the Python dependencies, Conda packages, pip packages, and related runtime settings needed by a training script. By defining scipy and the additional required packages in a single Azure ML Environment, you make the run configuration portable across compute targets. The same environment definition can be used when running locally for small datasets and when submitting the experiment to a managed remote compute cluster for larger workloads, reducing manual setup and avoiding package mismatch issues.

Registering the environment in the workspace also provides reuse and versioning. Once registered, the environment can be referenced consistently by script runs, shared across experiments, and automatically prepared on supported compute targets by Azure Machine Learning. This is the recommended approach for reproducible training and the least administrative effort because you do not need to manually configure every machine or rely on whatever packages happen to exist in a default environment. See Microsoft documentation on managing Azure Machine Learning environments and Azure Machine Learning environment concepts.

QUESTION NO: 23

You create an Azure Machine Learning compute resource to train models. The compute resource is configured as follows:

You must decrease the minimum number of nodes and increase the maximum number of nodes to the following values:

You need to reconfigure the compute resource.

What are three possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Use the Azure Machine Learning studio.

B. Run the update method of the AmlCompute class in the Python SDK.

C. Use the Azure portal.

D. Use the Azure Machine Learning designer.

E. Run the refresh_state() method of the BatchCompute class in the Python SDK

ANSWER: A B C

Explanation:

The correct ways to reconfigure the Azure Machine Learning compute resource are to use the Azure Machine Learning studio, run the update method of the AmlCompute class in the Python SDK, or use the Azure portal. Azure Machine Learning compute clusters support scale configuration changes after creation, including changing the minimum and maximum node counts used for autoscaling. In Azure Machine Learning studio, compute resources are managed from the Compute area, where a cluster can be selected and edited to update its scale settings. Programmatically, the Azure ML Python SDK v1 AmlCompute class exposes an update method that accepts properties such as min_nodes and max_nodes, which directly supports this scenario. The Azure portal can also be used to manage Azure Machine Learning workspace resources and access compute configuration for supported workspace compute operations. These methods are complete administrative paths for changing cluster scale settings without recreating the training compute resource. Microsoft documents compute management in studio at Create and manage compute resources in Azure Machine Learning studio and the SDK update capability at AmlCompute class documentation.

QUESTION NO: 24

You create an Azure Machine Learning compute resource to train models. The compute resource is configured as follows:

- Minimum nodes: 2
- Maximum nodes: 4

You must decrease the minimum number of nodes and increase the maximum number of nodes to the following values:

- Minimum nodes: 0
- Maximum nodes: 8

You need to reconfigure the compute resource.

What are three possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Use the Azure Machine Learning studio.

B. Run the update method of the AmlCompute class in the Python SDK.

C. Use the Azure portal.

D. Use the Azure Machine Learning designer.

E. Run the refresh_state() method of the BatchCompute class in the Python SDK.

ANSWER: A B C

Explanation:

Use the Azure Machine Learning studio, Run the update method of the AmlCompute class in the Python SDK, and Use the Azure portal are correct ways to reconfigure the scale settings of an Azure Machine Learning compute cluster. Azure Machine Learning compute clusters support autoscaling by defining a minimum node count and a maximum node count.

Setting the minimum nodes to 0 allows the cluster to scale down completely when idle, while increasing the maximum nodes to 8 allows more parallel capacity when training jobs require it. In Azure Machine Learning studio, compute resources can be managed from the Compute area, where cluster settings such as scale limits can be edited. Programmatically, the Azure ML Python SDK exposes the AmlCompute class, whose update method can change min_nodes and max_nodes for an existing compute target. The Azure portal can also be used to access and manage Azure Machine Learning workspace resources and compute-related configuration. Microsoft documents the AmlCompute update method in the SDK reference at AmlCompute class, and provides guidance for managing Azure Machine Learning compute resources at Create an Azure Machine Learning compute cluster.

QUESTION NO: 25

You manage an Azure Machine Learning workspace.

You must provide explanations for the behavior of the models with feature importance measures.

You need to configure a Responsible AI dashboard in Azure Machine Learning.

Which dashboard component should you configure?

A. Fairness assessment

B. Counterfactual what-if

C. Interpretability

D. Casual inference

ANSWER: C

Explanation:

Interpretability is the correct dashboard component because it is designed to help users understand how a machine learning model makes predictions, including which input features are most influential. In Azure Machine Learning's Responsible AI dashboard, the interpretability capability surfaces model explanations such as global feature importance, which summarizes the overall impact of features across the model, and local feature importance, which explains the factors that contributed to individual predictions. These explanations are specifically intended to make model behavior more transparent and easier to evaluate by data scientists, ML engineers, and responsible AI reviewers. When the requirement is to provide explanations for model behavior using feature importance measures, configuring Interpretability directly matches that goal. Microsoft describes model interpretability as a key part of Responsible AI tooling in Azure Machine Learning, enabling users to understand feature contributions and debug model decisions. For more details, see Microsoft's Responsible AI dashboard documentation at Responsible AI dashboard and the model interpretability guidance at Model interpretability in Azure Machine Learning.

QUESTION NO: 26

You manage an Azure At Foundry project.

You are implementing a RAG solution. The documents contain tables and images that must be broken into semantically relevant chunks.

You need to generate textual representations of images and tables to be used as chunks.

Which two chunking approaches should you use? Each correct answer presents a complete solution. Choose two.

NOTE: Each correct selection is worth one point.

A. Sentence-based parsing

B. Document layout analysis

C. Large language model augmentation

D. Fixed-size parsing

ANSWER: B C

Explanation:

Document layout analysis is correct because it uses document structure, such as headings, sections, tables, captions, and other layout elements, to produce chunks that preserve semantic context instead of splitting content arbitrarily. In Azure, Azure AI Document Intelligence layout analysis can extract structured content from documents, including tables and layout relationships, which is especially useful when tabular content needs to be represented as text or Markdown for retrieval. Large language model augmentation is also correct because an LLM can be used to create natural-language descriptions or summaries of non-textual or semi-structured content, such as images and complex tables, so those generated textual representations can be embedded and retrieved in a RAG pipeline. This approach is commonly used when raw extracted text is not enough and the solution needs richer semantic chunks for downstream grounding. Microsoft documents the Layout model capabilities in Azure AI Document Intelligence layout analysis, and Azure AI Search supports generative enrichment patterns such as prompt-based skill processing through the GenAI Prompt skill.

QUESTION NO: 27

You have a Jupyter Notebook that contains Python code that is used to train a model.

You must create a Python script for the production deployment. The solution must minimize code maintenance.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Refactor the Jupyter Notebook code into functions

B. Save each function to a separate Python file

C. Define a main() function in the Python script

D. Remove all comments and functions from the Python script

ANSWER: A C

Explanation:

“Refactor the Jupyter Notebook code into functions” and “Define a main() function in the Python script” are the correct actions because they turn exploratory notebook code into a maintainable, reusable training script without duplicating logic. In Azure Machine Learning production workflows, training code is typically submitted as a script or command job, with parameters, data access, model training, and model registration handled in a repeatable execution flow. Refactoring notebook cells into functions makes the code modular and easier to test, update, and reuse across local development, pipeline jobs, and deployment automation. Defining a main() function provides a clear entry point for the script, so orchestration logic can call the reusable functions in a predictable order while keeping importable code separate from executable code. This is especially useful when the same script is run repeatedly in Azure Machine Learning jobs or integrated into CI/CD processes. Microsoft’s Azure Machine Learning guidance shows model training being performed through scripts submitted as jobs, reinforcing the need for clean, script-based training code rather than notebook-only execution. See Train models with Azure Machine Learning and Run Jupyter notebooks in Azure Machine Learning.

QUESTION NO: 28

You create an Azure Machine Learning workspace. You are preparing a local Python environment on a laptop computer. You want to use the laptop to connect to the workspace and run experiments.

You create the following config.json file.

{

"workspace_name" : "ml-workspace" }

You must use the Azure Machine Learning SDK to interact with data and experiments in the workspace.

You need to configure the config.json file to connect to the workspace from the Python environment.

Which two additional parameters must you add to the config.json file in order to connect to the workspace? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. login

B. resource_group

C. subscription_id

D. key

E. region

ANSWER: B C

Explanation:

The correct additional parameters are **resource_group** and **subscription_id**. When using the Azure Machine Learning SDK from a local Python environment, the workspace configuration file is used by methods such as `Workspace.from_config()` to locate an existing Azure Machine Learning workspace. The minimum identifying values in the configuration are the workspace name, the Azure subscription ID that contains the workspace, and the resource group where the workspace resource is deployed. Since the provided file already includes `workspace_name`, adding `subscription_id` and `resource_group` completes the required workspace reference so the SDK can authenticate to Azure and resolve the correct workspace resource.

This is the standard pattern used when saving or manually creating an Azure Machine Learning workspace configuration file for reuse across local development environments, notebooks, and scripts. Microsoft documentation shows that a workspace can be loaded from configuration once these workspace identifiers are present, allowing the SDK to interact with experiments, datasets, compute targets, and other workspace assets. See Workspace class documentation and Azure Machine Learning environment configuration guidance.

QUESTION NO: 29

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are creating a new experiment in Azure Machine Learning Studio.

One class has a much smaller number of observations than the other classes in the training set.

You need to select an appropriate data sampling strategy to compensate for the class imbalance.

Solution: You use the Scale and Reduce sampling mode.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because the stated goal is to compensate for a class imbalance in the training data, where one class has far fewer observations than the others. In Azure Machine Learning, the appropriate sampling-oriented technique for this situation is typically Synthetic Minority Oversampling Technique, or SMOTE. SMOTE increases the representation of the minority class by generating synthetic examples from existing minority-class records, rather than merely duplicating rows. This can help a model learn the decision boundary for the underrepresented class more effectively and reduce bias toward the majority class.

The proposed solution does not meet the goal because class imbalance requires a resampling strategy that changes the class distribution or otherwise addresses minority-class underrepresentation. Microsoft's Azure Machine Learning component documentation describes SMOTE as a component used specifically to increase the number of cases in a dataset that are underrepresented. For more detail, see the Microsoft documentation for SMOTE and Azure Machine Learning component reference.

QUESTION NO: 30

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it as a result, these questions will not appear in the review screen.

You use Azure Machine Learning designer to load the following datasets into an experiment:

Dataset1

Age	Length	Width
3	22	13
7	11	96
18	32	85

Dataset2

Age	Length	Width
11	101	65
6	98	23
33	22	54
17	52	12

You need to create a dataset that has the same columns and header row as the input datasets and contains all rows from both input datasets.

Solution: Use the Apply Transformation module.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

The correct response is **No**. The requirement is to create one dataset that preserves the same schema and header row while appending all records from both input datasets. In Azure Machine Learning designer, that is a row-combination operation, typically performed with the **Add Rows** component when the datasets have compatible columns. The **Apply Transformation** component is used to apply a previously learned or saved transformation to a dataset, such as applying preprocessing logic that was produced by another transformation component. It does not concatenate two datasets or append the rows of one dataset to another. Therefore, using Apply Transformation would not produce a single dataset containing all rows from both source datasets. Microsoft's component reference describes Add Rows as the component for combining two datasets by appending rows, while Apply Transformation is documented for applying a transformation object to input data. See the Microsoft references for Add Rows and Apply Transformation.

QUESTION NO: 31

You are moving a large dataset from Azure Machine Learning Studio to a Weka environment.

You need to format the data for the Weka environment.

Which module should you use?

A. Convert to CSV

B. Convert to Dataset

C. Convert to ARFF

D. Convert to SVMLight

ANSWER: C

Explanation:

The correct module is Convert to ARFF because Weka's native interchange format is Attribute-Relation File Format (ARFF). When moving data out of Azure Machine Learning Studio for use in Weka, the dataset must be serialized with relation metadata, attribute definitions, and the data section in the structure Weka expects. The Convert to ARFF module is specifically designed for this purpose: it takes an Azure Machine Learning dataset and writes it in ARFF format so it can be downloaded or passed to tooling that understands Weka datasets. This preserves the tabular schema in a format suitable for Weka workflows such as preprocessing, classification, and feature selection. Microsoft's module reference identifies Convert to ARFF as the Azure Machine Learning Studio module for converting datasets and results to the ARFF format used by the Weka toolset: Microsoft Learn: Convert to ARFF. Weka's own documentation also describes ARFF as the standard file format for Weka datasets: Weka ARFF documentation.

QUESTION NO: 32

A set of CSV files contains sales records. All the CSV files have the same data schema.

Each CSV file contains the sales record for a particular month and has the filename sales.csv. Each file in stored in a folder that indicates the month and year when the data was recorded. The folders are in an Azure blob container for which a datastore has been defined in an Azure Machine Learning workspace. The folders are organized in a parent folder named sales to create the following hierarchical structure:

At the end of each month, a new folder with that month's sales file is added to the sales folder.

You plan to use the sales data to train a machine learning model based on the following requirements:

You need to register the sales data as a dataset in Azure Machine Learning service workspace.

What should you do?

A. Create a tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/ sales.csv' file every month. Register the dataset with the name sales_dataset each month, replacing the existing dataset and specifying a tag named month indicating the month and year it was registered. Use this dataset for all experiments.

B. Create a tabular dataset that references the datastore and specifies the path 'sales/*/sales.csv', register the dataset with the name sales_dataset and a tag named month indicating the month and year it was registered, and use this dataset for all experiments.

C. Create a new tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/ sales.csv' file every month. Register the dataset with the name sales_dataset_MM-YYYY each month with appropriate MM and YYYY values for the month and year. Use the appropriate month-specific dataset for experiments.

D. Create a tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/ sales.csv' file. Register the dataset with the name sales_dataset each month as a new version and with a tag named month indicating the month and year it was registered. Use this dataset for all experiments, identifying the version to be used based on the month tag as necessary.

ANSWER: B

Explanation:

Create a tabular dataset that references the datastore and specifies the path 'sales/*/sales.csv', register the dataset with the name sales_dataset and a tag named month indicating the month and year it was registered, and use this dataset for all experiments is correct because Azure Machine Learning tabular datasets can be created from delimited files in a datastore by supplying datastore paths, including wildcard path patterns. Since every monthly file has the same schema and is stored under the same parent folder, the wildcard path matches each monthly sales.csv file without requiring the dataset definition to list each file individually. Registering the dataset in the workspace provides a reusable, named dataset asset for experiments and pipelines while preserving the datastore-backed reference to the source data. This is the most maintainable pattern for a growing folder hierarchy where new month folders are added under the same structure. Microsoft's Azure ML v1 guidance shows creating TabularDataset objects from delimited files using datastore paths and wildcards, then registering them for reuse in a workspace. See Create and register Azure Machine Learning datasets and the TabularDatasetFactory.from_delimited_files API reference.

QUESTION NO: 33

You train and register a model in your Azure Machine Learning workspace.

You must publish a pipeline that enables client applications to use the model for batch inferencing. You must use a pipeline with a single ParallelRunStep step that runs a Python inferencing script to get predictions from the input data.

You need to create the inferencing script for the ParallelRunStep pipeline step.

Which two functions should you include? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. run(mini_batch)
D

B. main()

C. batch()

D. init()

E. score(mini_batch)

ANSWER: A D

Explanation:

For an Azure Machine Learning ParallelRunStep entry script, the correct functions to include are init() and a run function that accepts mini_batch. The init() function is called once when the worker process starts, so it is the right place to load the registered model, initialize reusable objects, and prepare any resources needed for scoring. The run(mini_batch) function is then called repeatedly by the ParallelRunStep runtime for each mini-batch of input data assigned to that worker. It contains the batch inference logic: read or transform the mini-batch input, call the model to generate predictions, and return or write the results in the expected format. Microsoft's guidance for ParallelRunStep entry scripts describes this same pattern: initialization logic goes in init(), and per-mini-batch processing goes in run(mini_batch). This design lets Azure Machine Learning parallelize inference across nodes and processes while avoiding repeated model loading for every mini-batch. See the Microsoft documentation for using ParallelRunStep and the ParallelRunStep Python SDK reference.

QUESTION NO: 34

You develop and train a machine learning model to predict fraudulent transactions for a hotel booking website.

Traffic to the site varies considerably. The site experiences heavy traffic on Monday and Friday and much lower traffic on other days. Holidays are also high web traffic days. You need to deploy the model as an Azure Machine Learning real-time web service endpoint on compute that can dynamically scale up and down to support demand. Which deployment compute option should you use?

A. attached Azure Databricks cluster

B. Azure Container Instance (ACI)

C. Azure Kubernetes Service (AKS) inference cluster

D. Azure Machine Learning Compute Instance

E. attached virtual machine in a different region

ANSWER: C

Explanation:

Azure Kubernetes Service (AKS) inference cluster is the correct deployment compute option for a production Azure Machine Learning real-time web service that must handle variable demand. AKS is designed for scalable, production-grade container orchestration and is a supported target for deploying Azure Machine Learning models as real-time endpoints. In this scenario, the fraud detection service must respond to unpredictable and uneven web traffic patterns, including high-demand days such as Mondays, Fridays, and holidays. AKS can scale the underlying cluster and application capacity to support these changing workloads, making it appropriate for a real-time inference service that needs elasticity and reliability.

Microsoft documentation describes AKS as a suitable production deployment target for Azure Machine Learning model inference, including scenarios that require high scale and operational control. AKS also supports autoscaling capabilities

through Kubernetes mechanisms and Azure AKS cluster autoscaler features, allowing compute resources to expand or contract based on demand. For more information, see Deploy a model to an Azure Kubernetes Service cluster and Automatically scale a cluster to meet application demands on AKS.

QUESTION NO: 35

You are building a regression model for estimating the number of calls during an event.

You need to determine whether the feature values achieve the conditions to build a Poisson regression model.

Which two conditions must the feature set contain? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. The label data must be a negative value.

B. The label data must be whole numbers.

C. The label data must be non-discrete.

D. The label data must be a positive value.

E. The label data can be positive or negative.

ANSWER: B D

Explanation:

Poisson regression is appropriate when the value being predicted is count data, such as the number of calls during an event. Count-based target values are modeled as outcomes from a Poisson distribution, so the label data must be whole numbers because a count of calls cannot be fractional. In addition, the label data must be constrained to positive, non-negative count values because a real-world event cannot have a negative number of calls. These requirements align with the Azure Machine Learning Poisson Regression guidance, which states that the response variable should follow a Poisson distribution, that counts cannot be negative, and that the method is not meaningful for non-whole numbers. Therefore, “The label data must be whole numbers” and “The label data must be a positive value” are the conditions that match the intended use of Poisson regression for predicting event call counts. See the Microsoft documentation for the Poisson Regression component and the general Azure Machine Learning algorithm and component guidance for related model-selection context.

QUESTION NO: 36

You use the Azure Machine Learning service to create a tabular dataset named training.data. You plan to use this dataset in a training script.

You create a variable that references the dataset using the following code:

training_ds = workspace.datasets.get("training_data")

You define an estimator to run the script.

You need to set the correct property of the estimator to ensure that your script can access the training.data dataset

Which property should you set?

A)

```
inputs = [training_ds.as_named_input('training_ds')]
```

B)

```
script_params = {"--training_ds":training_ds}
```

C)

```
environment_definition = {"training_data":training_ds}
```

D)

```
source_directory = training_ds
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A

Explanation:

The correct estimator property is the **inputs** property, using the dataset as a named input, for example `inputs=[training_ds.as_named_input('training_data')]`. In the Azure Machine Learning SDK v1 estimator pattern, datasets are made available to a training run by adding them to the estimator's inputs collection. Calling `as_named_input()` creates a named dataset consumption configuration, which lets the training script retrieve the dataset by name through the run context, such as with `Run.get_context().input_datasets['training_data']`, or consume it according to the configured mode. This is the supported way to pass a registered tabular dataset into a remote training job when defining an estimator. Microsoft's documentation for training with datasets shows datasets being supplied to training runs as named inputs, and the Python SDK reference documents `as_named_input` as the method used to create a named dataset input for a run. References: Train with datasets in Azure Machine Learning and AbstractDataset.as_named_input.

QUESTION NO: 37

A set of CSV files contains sales records. All the CSV files have the same data schema.

Each CSV file contains the sales record for a particular month and has the filename sales.csv. Each file is stored in a folder that indicates the month and year when the data was recorded. The folders are in an Azure blob container for which a datastore has been defined in an Azure Machine Learning workspace. The folders are organized in a parent folder named sales to create the following hierarchical structure:

At the end of each month, a new folder with that month's sales file is added to the sales folder.

You plan to use the sales data to train a machine learning model based on the following requirements:

- You must define a dataset that loads all of the sales data to date into a structure that can be easily converted to a dataframe.

- You must be able to create experiments that use only data that was created before a specific previous month, ignoring any data that was added after that month. - You must register the minimum number of datasets possible.

You need to register the sales data as a dataset in Azure Machine Learning service workspace.

What should you do?

A. Create a tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/sales.csv' file every month. Register the dataset with the name sales_dataset each month, replacing the existing dataset and specifying a tag named month indicating the month and year it was registered. Use this dataset for all experiments.

B. Create a tabular dataset that references the datastore and specifies the path 'sales/*/sales.csv', register the dataset with the name sales_dataset and a tag named month indicating the month and year it was registered, and use this dataset for all experiments.

C. Create a new tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/sales.csv' file every month. Register the dataset with the name sales_dataset_MM-YYYY each month with appropriate MM and YYYY values for the month and year. Use the appropriate month-specific dataset for experiments.

D. Create a tabular dataset that references the datastore and explicitly specifies each 'sales/mm-yyyy/sales.csv' file. Register the dataset with the name sales_dataset each month as a new version and with a tag named month indicating the month and year it was registered. Use this dataset for all experiments, identifying the version to be used based on the month tag as necessary.

ANSWER: D

Explanation:

Creating a tabular dataset that explicitly specifies each monthly CSV file, then registering it under the same dataset name as a new version each month, is the correct approach. A TabularDataset is appropriate because the CSV files share the same schema and can be loaded into a tabular structure that is easily converted to a pandas dataframe. Registering the dataset repeatedly with the same name creates versioned dataset definitions, so experiments can deliberately retrieve the version that corresponds to the desired historical cutoff month. That satisfies the requirement to run experiments using only data available before a specific previous month, because each version can reference only the files that existed up to that month. Using one registered dataset name with multiple versions also satisfies the requirement to register the minimum number of datasets possible, while still preserving reproducibility for older experiments. Microsoft documents that Azure Machine Learning datasets can be registered and versioned, and that tabular datasets can be created from delimited files in datastores. See Azure ML Dataset class and Azure ML TabularDataset.

QUESTION NO: 38 - (HOTSPOT)

HOTSPOT

Your Azure Machine Learning workspace has a dataset named real_estate_data. A sample of the data in the dataset follows.

postal_code	num_bedrooms	sq_feet	garage	price
12345	3	1300	0	23,9000
54321	1	950	0	11,0000
12346	2	1200	1	15,0000

You want to use automated machine learning to find the best regression model for predicting the price column.

You need to configure an automated machine learning experiment using the Azure Machine Learning SDK.

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

```
from azureml.core import Workspace
from azureml.core.compute import ComputeTarget
from azureml.core.runconfig import RunConfiguration
from azureml.train.automl import AutoMLConfig

ws = Workspace.from_config()
training_cluster = ComputeTarget(workspace=ws, name= 'aml-cluster1')
real_estate_ds = ws.datasets.get('real_estate_data')
split1_ds, split2_ds = real_estate_ds.random_split(percentage=0.7, seed=123)
automl_run_config = RunConfiguration(framework= "python")
automl_config = AutoMLConfig(          task= 'regression',
                                       compute_target= training_cluster,
                                       run_configuration=automl_run_config,
                                       primary_metric='r2_score',
                                                                  =split1_ds,
                                                                  =split2_ds
                                                                  ='price')
```

- X
- Y
- X_valid
- Y_valid
- training_data

- X
- Y
- X_valid
- Y_valid
- validation_data
- training_data

- y
- y_valid
- y_max
- label_column_name
- exclude_nan_labels

ANSWER:

Answer Area

```
from azureml.core import Workspace
from azureml.core.compute import ComputeTarget
from azureml.core.runconfig import RunConfiguration
from azureml.train.automl import AutoMLConfig

ws = Workspace.from_config()
training_cluster = ComputeTarget(workspace=ws, name= 'aml-cluster1')
real_estate_ds = ws.datasets.get('real_estate_data')
split1_ds, split2_ds = real_estate_ds.random_split(percentage=0.7, seed=123)
automl_run_config = RunConfiguration(framework= "python")
automl_config = AutoMLConfig(          task= 'regression',
                                       compute_target= training_cluster,
                                       run_configuration=automl_run_config,
                                       primary_metric='r2_score',
                                       [▼]=split1_ds,
                                       [▼]=split2_ds
                                       [▼]='price')
```

[▼] options: X, Y, X_valid, Y_valid, training_data

[▼] options: X, Y, X_valid, Y_valid, validation_data, training_data

[▼] options: y, y_valid, y_max, label_column_name, exclude_nan_labels

Explanation:

The correct configuration uses **training_data** for the first split, **validation_data** for the second split, and **label_column_name** for the **price** column. The dataset shown contains feature columns such as postal_code, num_bedrooms, sq_feet, and garage, plus the target value that the regression model must learn to predict. In Azure Machine Learning automated ML, the training dataset must contain both the input features and the target column, so the first result of the random split can be passed directly to the **training_data** argument. Since the code explicitly creates a second dataset from the split, that second dataset should be supplied through **validation_data** so automated ML can evaluate candidate regression models on held-out data during model selection.

The target column is identified separately by setting **label_column_name** to **'price'**. This tells automated ML which column represents the value to predict and prevents the price values from being treated as ordinary input features. Because the task is already set to **'regression'** and the primary metric is **'r2_score'**, this setup is consistent with finding the best regression model for predicting real estate prices. Microsoft's Azure ML SDK documentation for AutoMLConfig describes these parameters: **training_data** supplies the data for model training, **validation_data** supplies validation data, and **label_column_name** identifies the label column for supervised learning. This matches the structure of the provided code and dataset exactly.

QUESTION NO: 39

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are a data scientist using Azure Machine Learning Studio.

You need to normalize values to produce an output column into bins to predict a target column.

Solution: Apply a Quantiles normalization with a QuantileIndex normalization.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

The correct answer is No. The stated goal is not just to split numeric values into quantile-based buckets; it is to create binned output that can be used to predict a target column. In Azure Machine Learning Studio, this is a supervised binning requirement because the bin boundaries should be chosen with awareness of the target variable. The appropriate approach is to use the Group Data into Bins module with the Entropy MDL binning mode, which is designed for predictive scenarios and requires a target column when determining the binning strategy. Entropy MDL uses the relationship between the input column and the target column to create bins that are more meaningful for classification or prediction tasks. Quantiles normalization with QuantileIndex assigns values to quantile-based indexes based on the distribution of the input values, but it does not use the target column to optimize the bins for prediction. Therefore, it does not meet the stated requirement. See Microsoft's documentation for the Group Data into Bins component and the broader Azure Machine Learning designer component reference.

QUESTION NO: 40

You are performing clustering by using the K-means algorithm.

You need to define the possible termination conditions.

Which three conditions can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Centroids do not change between iterations.

B. The residual sum of squares (RSS) rises above a threshold.

C. The residual sum of squares (RSS) falls below a threshold.

D. A fixed number of iterations is executed.

E. The sum of distances between centroids reaches a maximum.

ANSWER: A C D

Explanation:

Centroids do not change between iterations is a valid K-means termination condition because the algorithm has converged when reassigning points and recomputing cluster centers no longer changes the cluster centroids. At that point, additional iterations would produce the same clustering result. The residual sum of squares (RSS) falls below a threshold is also a valid stopping condition because K-means minimizes the within-cluster squared distance objective; if the objective is already below an acceptable threshold, the clustering quality target has been met. A fixed number of iterations is executed is another common termination condition, especially in practical implementations, because it prevents indefinite execution and provides a deterministic upper bound on training time. Microsoft's K-Means Clustering component documentation describes iterative training behavior and the use of a maximum number of iterations, while the standard K-means objective is based on

minimizing squared distances within clusters. See Microsoft Learn: K-Means Clustering component and Stanford IR book: K-means.

QUESTION NO: 41

You need to visually identify whether outliers exist in the Age column and quantify the outliers before the outliers are removed.

Which three Azure Machine Learning Studio modules should you use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Create Scatterplot

B. Summarize Data

C. Clip Values

D. Replace Discrete Values

E. Build Counting Transform

ANSWER: A B C

Explanation:

Create Scatterplot, Summarize Data, and Clip Values are the appropriate Azure Machine Learning Studio modules for this task. Create Scatterplot supports the visual inspection requirement: by plotting the Age column against another numeric field or record index, you can quickly see unusually high or low values that stand apart from the rest of the distribution. Summarize Data supports the quantification requirement because it generates descriptive statistics such as count, minimum, maximum, mean, standard deviation, and percentile-style summaries that help determine the scale and frequency of unusual Age values before any cleansing action is applied. Clip Values is the module used when the identified outliers need to be handled according to a defined threshold; it can detect values above or below specified limits and optionally replace them with boundary values, means, constants, or missing values. Together, these modules provide a practical workflow: inspect the column visually, summarize the column statistically, and then apply a controlled outlier treatment step. Microsoft's module documentation describes Clip Values as a way to identify and optionally replace values outside specified thresholds, and the classic Studio module reference includes visualization and summarization modules used for exploratory data analysis. See Clip Values and Summarize Data.

QUESTION NO: 42

You create a binary classification model. You use the Fairlearn package to assess model fairness. You must eliminate the need to retrain the model. You need to implement the Fair learn package. Which algorithm should you use?

A. fairlearn.reductions.ExponentiatedGradient

B. fatrlearn.reductions.GridSearch

C. fair Icarn.postprocessing.ThresholdOplimizer

D. fairlearn.preprocessing.CorrelationRemover

ANSWER: C

Explanation:

The correct choice is fair Icarn.postprocessing.ThresholdOplimizer, which refers to Fairlearn's postprocessing ThresholdOptimizer. ThresholdOptimizer is designed for binary classification scenarios where you already have a trained model and want to mitigate fairness issues by adjusting the decision thresholds applied to the model's scores or predictions. This is the key requirement in the question: eliminating the need to retrain the original model. As a postprocessing mitigation technique, it learns group-specific thresholding rules after the model has been trained, so the underlying classifier does not

need to be rebuilt or retrained. This makes it appropriate when a model is already in production or when retraining is expensive, impractical, or not allowed. Fairlearn documents ThresholdOptimizer as a postprocessing algorithm for binary classification that can optimize predictions under fairness constraints such as demographic parity or equalized odds. Microsoft's responsible AI guidance also references Fairlearn as a toolkit for assessing and mitigating unfairness in machine learning workflows. For more detail, see the Fairlearn documentation for ThresholdOptimizer and Microsoft Learn's guidance on model fairness in Azure Machine Learning.

QUESTION NO: 43

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You train and register a machine learning model.

You plan to deploy the model as a real-time web service. Applications must use key-based authentication to use the model.

You need to deploy the web service.

Solution:

Create an AksWebservice instance.

Set the value of the auth_enabled property to False.

Set the value of the token_auth_enabled property to True.

Deploy the model to the service.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

The correct response is No. The requirement is that applications must use key-based authentication when calling the deployed real-time web service. In Azure Machine Learning SDK v1 deployments, key-based authentication is controlled by the auth_enabled setting on the web service deployment configuration. For an AKS web service, enabling key-based authentication means the service exposes primary and secondary keys that clients include when invoking the scoring endpoint. Setting auth_enabled to False does not enable that key-based access pattern.

The solution instead sets token_auth_enabled to True. Token authentication is a different authentication mechanism from key-based authentication; it requires callers to obtain and present an authentication token rather than using the service's generated keys. Since the stated goal specifically requires key-based authentication, the proposed configuration does not meet the requirement. The deployment should use key authentication by enabling auth_enabled for the AKS web service configuration. Microsoft documents these authentication-related properties for AKS web service deployments in the Azure Machine Learning SDK, including auth_enabled and token_auth_enabled, in the AksWebservice class documentation and related AKS deployment configuration documentation.

QUESTION NO: 44

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You train and register a machine learning model.

You plan to deploy the model as a real-time web service. Applications must use key-based authentication to use the model.

You need to deploy the web service.

Solution:

Create an AciWebservice instance.

Set the value of the ssl_enabled property to True.

Deploy the model to the service.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

No is correct because enabling SSL on an Azure Container Instances web service only configures HTTPS/TLS for the scoring endpoint; it does not enable key-based authentication. The requirement is that client applications must authenticate by using keys before they can call the deployed model. For Azure Machine Learning SDK v1 ACI deployments, key-based authentication is controlled by the `auth_enabled` setting in the ACI deployment configuration. For ACI web services, authentication is not enabled by default, so the deployment configuration must explicitly set `auth_enabled=True`. The `ssl_enabled=True` setting can be useful when securing traffic to the endpoint, but it addresses encryption in transit rather than access control. Therefore, the proposed solution does not meet the stated goal because it omits the required key-based authentication configuration. Microsoft's API reference for `AciWebservice.deploy_configuration` documents both `auth_enabled` and `ssl_enabled` as separate settings, and the Azure Machine Learning web service authentication guidance describes using authentication keys for secured service calls. See AciWebservice class and Authenticate clients for Azure Machine Learning web services.

QUESTION NO: 45

You must store data in Azure Blob Storage to support Azure Machine Learning.

You need to transfer the data into Azure Blob Storage.

What are three possible ways to achieve the goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Bulk Insert SQL Query

B. AzCopy

C. Python script

D. Azure Storage Explorer

E. Bulk Copy Program (BCP)

ANSWER: B C D

Explanation:

AzCopy, Python script, and Azure Storage Explorer are correct ways to transfer data into Azure Blob Storage for use with Azure Machine Learning. AzCopy is Microsoft's command-line utility designed specifically for copying blobs and files to, from, and between Azure Storage accounts, making it well suited for repeatable or large-scale uploads. Azure Storage Explorer provides a graphical interface for browsing storage accounts and uploading files or folders into Blob containers, which is useful for interactive data management without writing code. A Python script is also a complete solution because Azure provides Blob Storage client libraries for Python that can authenticate to a storage account and upload data programmatically; this approach is commonly used in data science workflows and automation pipelines. These methods are aligned with Microsoft guidance for moving data into Azure Blob Storage and can support datasets that are later registered or consumed by Azure Machine Learning. For details, see Microsoft's guidance on moving data to and from Azure Blob Storage at Move data to and from Azure Blob Storage and the AzCopy documentation at Get started with AzCopy.

QUESTION NO: 46

You manage an Azure Machine Learning workspace.

You build an image recognition training pipeline, which includes hyperparameter tuning. For each epoch run, you plan to log the following metrics:

• the transformed images used for training in an existing folder

• a description to explain the hyperparameter changes

You need to configure logging for the experiment.

Which two functions should you use? Each correct answer presents part of the solution. Choose two. NOTE: Each correct selection is worth one point.

A. mlflow.log_artifact()

B. mlflow.log_artifacts()

C. mlflow.logjnetric()

D. mlflow.log_param()

ANSWER: B D

Explanation:

The correct functions are **mlflow.log_artifacts()** and **mlflow.log_param()**. In Azure Machine Learning, MLflow is the recommended tracking API for logging experiment information such as parameters, metrics, and artifacts. Because the transformed training images are already stored in an existing folder, **mlflow.log_artifacts()** is the appropriate function: it logs all files from a local directory as artifacts for the current run, making the image outputs available for later inspection in the run record. For documenting the hyperparameter changes, **mlflow.log_param()** is appropriate because MLflow parameters are key-value inputs or configuration values associated with a run. A descriptive value can be logged as a parameter, such as a key named "change_description" with text explaining what was modified for that epoch or run. This aligns with Azure ML experiment tracking practices, where artifacts capture files and folders produced or used during experimentation, while parameters capture configuration details used to reproduce and compare runs. See Microsoft's guidance on logging and viewing metrics and artifacts in Azure Machine Learning at Microsoft Learn, and MLflow tracking support in Azure Machine Learning at Microsoft Learn.

QUESTION NO: 47 - (SIMULATION)

You create an Azure Data Lake Storage Gen2 stowage account named storage1 containing a file system named fsi and a folder named folder1.

The contents of folder1 must be accessible from jobs on compute targets in the Azure Machine Learning workspace.

You need to construct a URI to reference folder1.

How should you construct the URI? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

ANSWER: See the explanation for the answer

Explanation:

See below image

QUESTION NO: 48

You create a machine learning model by using the Azure Machine Learning designer. You publish the model as a real-time service on an Azure Kubernetes Service (AKS) inference compute cluster. You make no changes to the deployed endpoint configuration.

You need to provide application developers with the information they need to consume the endpoint.

Which two values should you provide to application developers? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. The name of the AKS cluster where the endpoint is hosted.

B. The name of the inference pipeline for the endpoint.

C. The URL of the endpoint.

D. The run ID of the inference pipeline experiment for the endpoint.

E. The key for the endpoint.

ANSWER: C E

Explanation:

When an Azure Machine Learning designer model is published as a real-time service on Azure Kubernetes Service, consuming applications call it through its scoring REST endpoint. Therefore, application developers need **The URL of the endpoint.** as the target address for prediction requests. Because the deployed endpoint configuration is left unchanged, the real-time endpoint is protected by authentication, so developers also need **The key for the endpoint.**
to authorize calls to the service. In practice, the endpoint URL is used as the scoring URI in the HTTP request, and the key is sent in the request headers, typically as a bearer token or authorization value depending on the endpoint type and SDK/client pattern. Microsoft documentation for consuming Azure Machine Learning web services describes retrieving the scoring URI and, when authentication is enabled, obtaining the service keys or tokens required to submit requests successfully. See Microsoft Learn: Consume an Azure Machine Learning model deployed as a web service and Microsoft Learn: Deploy and score a machine learning model by using an online endpoint.