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

QUESTION 217
Hotspot Question
You use Azure Machine Learning to train and register a model.
You must deploy the model into production as a real-time web service to an inference cluster named service-compute that the IT department has created in the Azure Machine Learning workspace.
Client applications consuming the deployed web service must be authenticated based on their Azure Active Directory service principal.
You need to write a script that uses the Azure Machine Learning SDK to deploy the model. The necessary modules have been imported.
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

```
# Assume the necessary modules have been imported
deploy_target = [AksCompute | AmlCompute | RemoteCompute | BatchCompute](ws, "service-compute")
deployment_config = [AksWebservice | AciWebservice | LocalWebService].deploy_configuration(cpu_cores=1, memory_gb=1,
[token_auth_enabled=True | token_auth_enabled=False | auth_enabled=True | auth_enabled=False])
service = Model.deploy(ws, "ml-service",
 [model], inference_config, deployment_config, deploy_target)
service.wait_for_deployment(show_output = True)
```

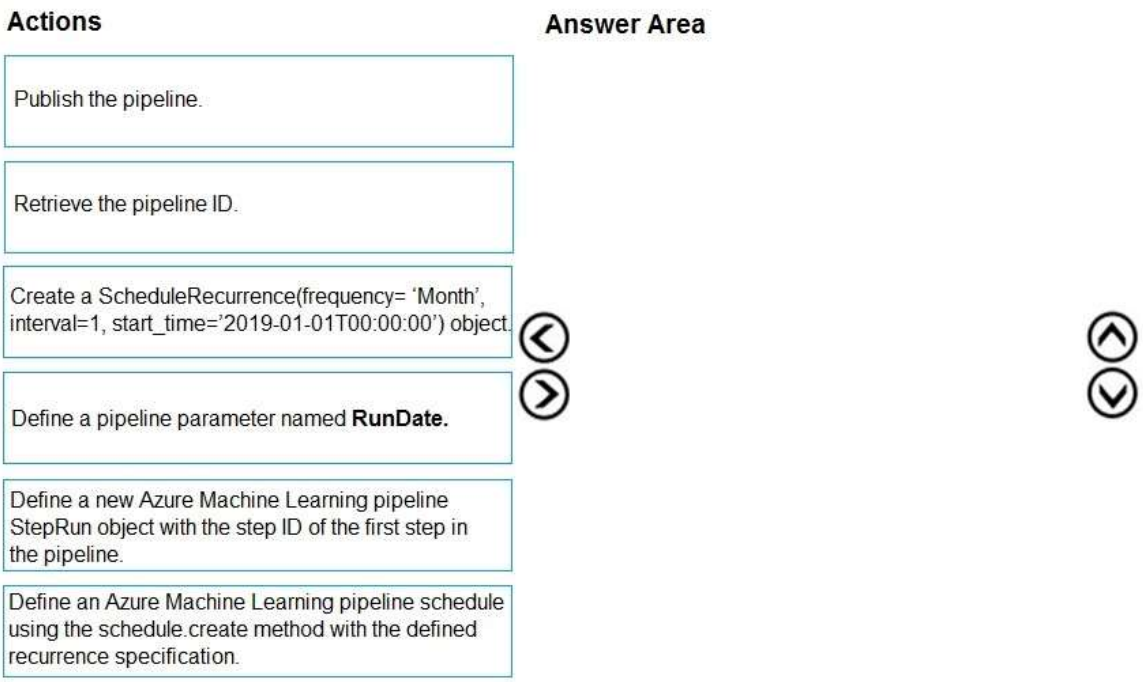
Answer:

Answer Area

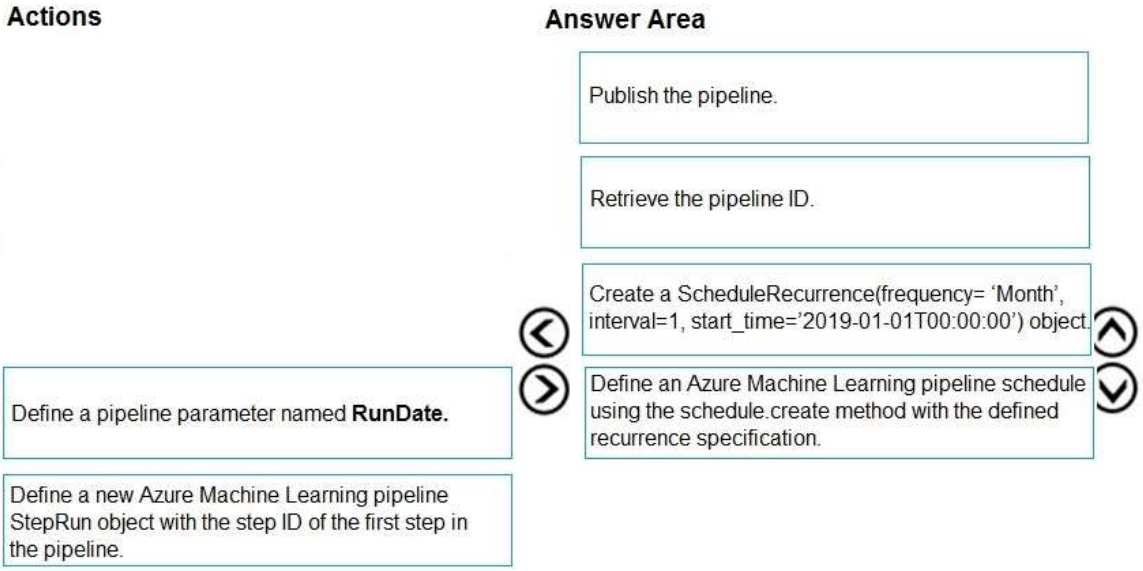
```
# Assume the necessary modules have been imported
deploy_target = [AksCompute | AmlCompute | RemoteCompute | BatchCompute](ws, "service-compute")
deployment_config = [AksWebservice | AciWebservice | LocalWebService].deploy_configuration(cpu_cores=1, memory_gb=1,
[token_auth_enabled=True | token_auth_enabled=False | auth_enabled=True | auth_enabled=False])
service = Model.deploy(ws, "ml-service",
 [model], inference_config, deployment_config, deploy_target)
service.wait_for_deployment(show_output = True)
```

Explanation:
Box 1: AksCompute
Example:
aks_target = AksCompute(ws,"myaks")
# If deploying to a cluster configured for dev/test, ensure that it was created with enough # cores and memory to handle this deployment configuration. Note that memory is also used by # things such as dependencies and AML components.
deployment_config = AksWebservice.deploy_configuration(cpu_cores = 1, memory_gb = 1) service = Model.deploy(ws, "myservice", [model], inference_config, deployment_config, aks_target)
Box 2: AksWebservice
Box 3: token_auth_enabled=Yes
Whether or not token auth is enabled for the Webservice.
Note: A Service principal defined in Azure Active Directory (Azure AD) can act as a principal on which authentication and authorization policies can be enforced in Azure Databricks.
The Azure Active Directory Authentication Library (ADAL) can be used to programmatically get an Azure AD access token for a user.
Incorrect Answers:
auth_enabled (bool): Whether or not to enable key auth for this Webservice. Defaults to True.
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-azure-kubernetes-service
https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/aad/service-prin-aad-token

QUESTION 218
Drag and Drop Question
You create a multi-class image classification deep learning model.
The model must be retrained monthly with the new image data fetched from a public web portal. You create an Azure Machine Learning pipeline to fetch new data, standardize the size of images, and retrain the model.
You need to use the Azure Machine Learning SDK to configure the schedule for the pipeline.
Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



Explanation:

Step 1: Publish the pipeline.

To schedule a pipeline, you'll need a reference to your workspace, the identifier of your published pipeline, and the name of the experiment in which you wish to create the schedule.

Step 2: Retrieve the pipeline ID.

Needed for the schedule.

Step 3: Create a ScheduleRecurrence..

To run a pipeline on a recurring basis, you'll create a schedule. A Schedule associates a pipeline, an experiment, and a trigger.

First create a schedule. Example: Create a Schedule that begins a run every 15 minutes:

recurrence = ScheduleRecurrence(frequency="Minute", interval=15)

Step 4: Define an Azure Machine Learning pipeline schedule..

Example, continued:

recurring_schedule = Schedule.create(ws, name="MyRecurringSchedule", description="Based on time",
pipeline_id=pipeline_id,
experiment_name=experiment_name,
recurrence=recurrence)
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-schedule-pipelines

QUESTION 219
Hotspot Question
You create a script for training a machine learning model in Azure Machine Learning service.
You create an estimator by running the following code:

```
from azureml.core import Workspace, Datastore
from azureml.core.compute import ComputeTarget
from azureml.train.estimator import Estimator
work_space = Workspace.from_config()
data_source = work_space.get_default_datastore()
train_cluster = ComputeTarget(workspace=work_space, name= 'train-cluster')
estimator = Estimator(source_directory =
 'training-experiment',
script_params = { ' --data-folder' : data_source.as_mount(), ' --regularization':0.8},
compute_target = train_cluster,
entry_script = 'train.py',
conda_packages = ['scikit-learn'])
```

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

Answer Area

	Yes	No
The estimator will look for the files it needs to run an experiment in the training-experiment directory of the local compute environment.	○	○
The estimator will mount the local data-folder folder and make it available to the script through a parameter.	○	○
The train.py script file will be created if it does not exist.	○	○
The estimator can run Scikit-learn experiments.	○	○

Answer:

Answer Area

	Yes	No
The estimator will look for the files it needs to run an experiment in the training-experiment directory of the local compute environment.	○	○
The estimator will mount the local data-folder folder and make it available to the script through a parameter.	○	○
The train.py script file will be created if it does not exist.	○	○
The estimator can run Scikit-learn experiments.	○	○

Explanation:
Box 1: Yes
Parameter source_directory is a local directory containing experiment configuration and code files needed for a training job.
Box 2: Yes
script_params is a dictionary of command-line arguments to pass to the training script specified in entry_script.
Box 3: No
Box 4: Yes
The conda_packages parameter is a list of strings representing conda packages to be added to the Python environment for the experiment.

QUESTION 220
Hotspot Question
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,
- X
- Y
- X_valid
- Y_valid
- training_data

[▼] =split2_ds
- X
- Y
- X_valid
- Y_valid
- validation_data
- training_data

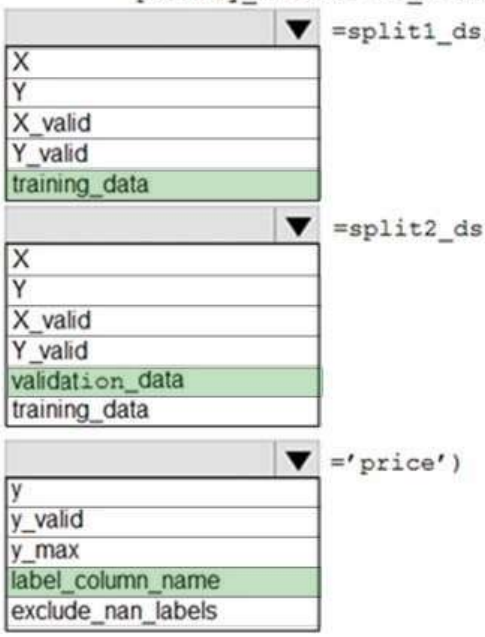
[▼] ='price')
- y
- y_valid
- y_max
- label_column_name
- exclude_nan_labels

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',
```



Explanation:

Box 1: training_data

The training data to be used within the experiment. It should contain both training features and a label column (optionally a sample weights column). If training_data is specified, then the label_column_name parameter must also be specified.

Box 2: validation_data

Provide validation data: In this case, you can either start with a single data file and split it into training and validation sets or you can provide a separate data file for the validation set. Either way, the validation_data parameter in your AutoMLConfig object assigns which data to use as your validation set.

Example, the following code example explicitly defines which portion of the provided data in dataset to use for training and validation.

```
dataset = Dataset.Tabular.from_delimited_files(data)
training_data, validation_data = dataset.random_split(percentage=0.8, seed=1)
automl_config = AutoMLConfig(compute_target = aml_remote_compute, task = 'classification',
primary_metric = 'AUC_weighted',
training_data = training_data,
validation_data = validation_data,
label_column_name = 'Class'
)
```

Box 3: label_column_name

label_column_name:

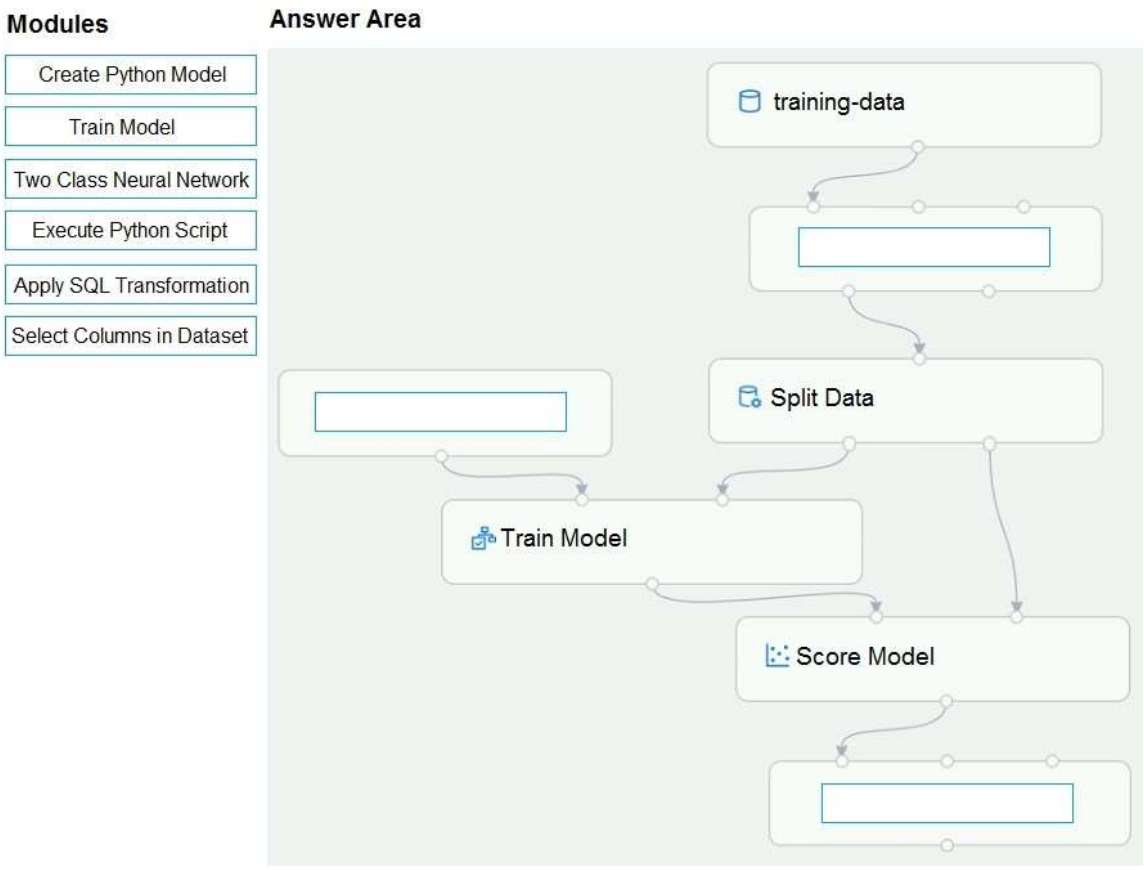
The name of the label column. If the input data is from a pandas.DataFrame which doesn't have column names,

column indices can be used instead, expressed as integers.
This parameter is applicable to training_data and validation_data parameters.
Incorrect Answers:
X: The training features to use when fitting pipelines during an experiment. This setting is being deprecated. Please use training_data and label_column_name instead.
Y: The training labels to use when fitting pipelines during an experiment. This is the value your model will predict. This setting is being deprecated. Please use training_data and label_column_name instead.
X_valid: Validation features to use when fitting pipelines during an experiment. If specified, then y_valid or sample_weight_valid must also be specified.
Y_valid: Validation labels to use when fitting pipelines during an experiment.
Both X_valid and y_valid must be specified together.
exclude_nan_labels: Whether to exclude rows with NaN values in the label. The default is True.
y_max: y_max (float)
Maximum value of y for a regression experiment. The combination of y_min and y_max are used to normalize test set metrics based on the input data range. If not specified, the maximum value is inferred from the data.
Reference:
https://docs.microsoft.com/en-us/python/api/azureml-train-automl-client/azureml.train.automl.automlconfig.automlconfig?view=azure-ml-py

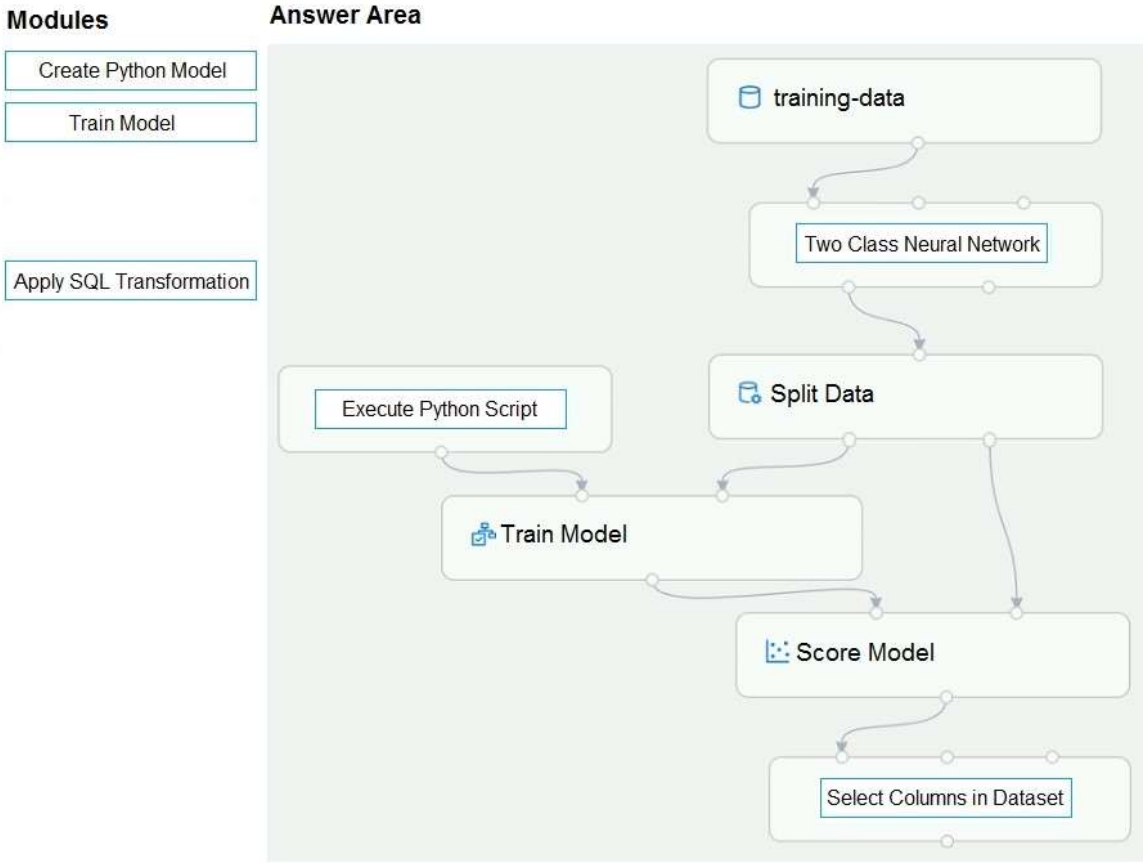
QUESTION 221
Drag and Drop Question
You create a training pipeline using the Azure Machine Learning designer. You upload a CSV file that contains the data from which you want to train your model.
You need to use the designer to create a pipeline that includes steps to perform the following tasks:

```
- Select the training features using the pandas filter method.
- Train a model based on the naive_bayes.GaussianNB algorithm.
- Return only the Scored Labels column by using the query SELECT [Scored Labels] FROM t1;
```

Which modules should you use? To answer, drag the appropriate modules to the appropriate locations. Each module name may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:

Box 1: Two-Class Neural Network
The Two-Class Neural Network creates a binary classifier using a neural network algorithm. Train a model based on the naive_bayes.GaussianNB algorithm.
Box 2: Execute python script
Select the training features using the pandas filter method
Box 3: Select Columns in DataSet
Return only the Scored Labels column by using the query SELECT [Scored Labels] FROM t1;
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-neural-network

QUESTION 222
Hotspot Question
You collect data from a nearby weather station.
You have a pandas dataframe named weather_df that includes the following data:

Temperature	Observation_time	Humidity	Pressure	Visibility	Days_since_last observation
74	2019/10/2 00:00	0.62	29.87	3	0.5
89	2019/10/2 12:00	0.70	28.88	10	0.5
72	2019/10/3 00:00	0.64	30.00	8	0.5
80	2019/10/3 12:00	0.66	29.75	7	0.5

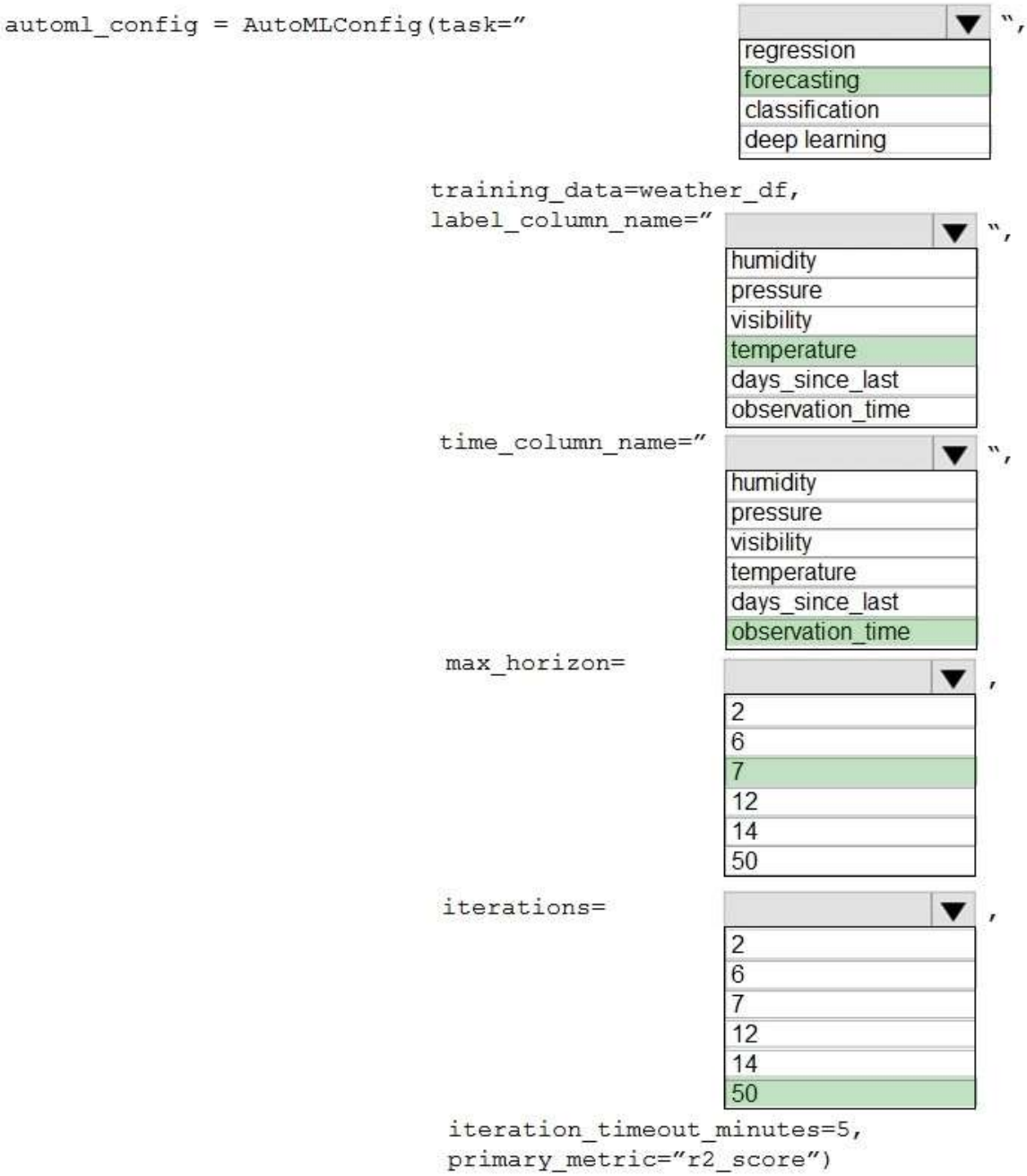
The data is collected every 12 hours: noon and midnight.
You plan to use automated machine learning to create a time-series model that predicts temperature over the next seven days. For the initial round of training, you want to train a maximum of 50 different models.
You must use the Azure Machine Learning SDK to run an automated machine learning experiment to train these models.
You need to configure the automated machine learning run.
How should you complete the AutoMLConfig definition? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

```
automl_config = AutoMLConfig(task="  [ regression | forecasting | classification | deep learning ] ",
                             training_data=weather_df,
                             label_column_name="  [ humidity | pressure | visibility | temperature | days_since_last | observation_time ] ",
                             time_column_name="  [ humidity | pressure | visibility | temperature | days_since_last | observation_time ] ",
                             max_horizon=  [ 2 | 6 | 7 | 12 | 14 | 50 ] ,
                             iterations=  [ 2 | 6 | 7 | 12 | 14 | 50 ] ,
                             iteration_timeout_minutes=5,
                             primary_metric="r2_score")
```

Answer:

Answer Area



Explanation:
Box 1: forcasting
Task: The type of task to run. Values can be 'classification', 'regression', or 'forecasting' depending on the type of automated ML problem to solve.
Box 2: temperature
The training data to be used within the experiment. It should contain both training features and a label column (optionally a sample weights column).
Box 3: observation_time
time_column_name: The name of the time column. This parameter is required when forecasting to specify the datetime column in the input data used for building the time series and inferring its frequency. This setting is being deprecated. Please use forecasting_parameters instead.
Box 4: 7

"predicts temperature over the next seven days"
max_horizon: The desired maximum forecast horizon in units of time-series frequency. The default value is 1.
Units are based on the time interval of your training data, e.g., monthly, weekly that the forecaster should predict out.
When task type is forecasting, this parameter is required.
Box 5: 50
"For the initial round of training, you want to train a maximum of 50 different models."
Iterations: The total number of different algorithm and parameter combinations to test during an automated ML experiment.
Reference:
https://docs.microsoft.com/en-us/python/api/azureml-train-automl-client/azureml.train.automl.automlconfig.automlconfig

QUESTION 223
Hotspot Question
You train a classification model by using a decision tree algorithm.
You create an estimator by running the following Python code. The variable feature_names is a list of all feature names, and class_names is a list of all class names.

```
from interpret.ext.blackbox import TabularExplainer
explainer = TabularExplainer(model,
                             x_train,
                             features=feature_names,
                             classes=classe_names)
```

You need to explain the predictions made by the model for all classes by determining the importance of all features.
For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
The SHAP TreeExplainer will be used to interpret the model.	○	○
If you omit the features and classes parameters in the TabularExplainer instantiation, the explainer still works as expected.	○	○
You could interpret the model by using a MimicExplainer instead of a TabularExplainer.	○	○

Answer:

Answer Area

	Yes	No
The SHAP TreeExplainer will be used to interpret the model.	○	○
If you omit the features and classes parameters in the TabularExplainer instantiation, the explainer still works as expected.	○	○
You could interpret the model by using a MimicExplainer instead of a TabularExplainer.	○	○

Explanation:
Box 1: Yes
TabularExplainer calls one of the three SHAP explainers underneath (TreeExplainer, DeepExplainer, or KernelExplainer).
Box 2: Yes
To make your explanations and visualizations more informative, you can choose to pass in feature names and output class names if doing classification.

Box 3: No
TabularExplainer automatically selects the most appropriate one for your use case, but you can call each of its three underlying explainers underneath (TreeExplainer, DeepExplainer, or KernelExplainer) directly.
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability-aml

QUESTION 224
Hotspot Question
You are hired as a data scientist at a winery. The previous data scientist used Azure Machine Learning.
You need to review the models and explain how each model makes decisions.
Which explainer modules should you use? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

Model type	Explainer
A random forest model for predicting the alcohol content in wine given a set of covariates	Tabular / HAN / Text / Image
A natural language processing model for analyzing field reports	Tree / HAN / Text / Image
An image classifier that determines the quality of the grape based upon its physical characteristics.	Kernel / HAN / Text / Image

Answer:

Answer Area

Model type	Explainer
A random forest model for predicting the alcohol content in wine given a set of covariates	Tabular
A natural language processing model for analyzing field reports	Text
An image classifier that determines the quality of the grape based upon its physical characteristics.	Image

Explanation:
Meta explainers automatically select a suitable direct explainer and generate the best explanation info based on the given model and data sets. The meta explainers leverage all the libraries (SHAP, LIME, Mimic, etc.) that we have integrated or developed. The following are the meta explainers available in the SDK:
Tabular Explainer: Used with tabular datasets.
Text Explainer: Used with text datasets.
Image Explainer: Used with image datasets.
Box 1: Tabular
Box 2: Text
Box 3: Image
Incorrect Answers:
Hierarchical Attention Network (HAN)
HAN was proposed by Yang et al. in 2016. Key features of HAN that differentiates itself from existing approaches to document classification are (1) it exploits the hierarchical nature of text data and (2) attention mechanism is adapted for document classification.
Reference:
https://medium.com/microsoftazure/automated-and-interpretable-machine-learning-d07975741298

QUESTION 225
Hotspot Question
You have a dataset that includes home sales data for a city. The dataset includes the following columns.

Name	Description
Price	The sales price for the house.
Bedrooms	The number of bedrooms in the house.
Size	The size of the house in square feet.
HasGarage	A binary value indicating whether or not the house has a garage.
HomeType	The category of home, for example, apartment, townhouse, single-family home.

Each row in the dataset corresponds to an individual home sales transaction.
You need to use automated machine learning to generate the best model for predicting the sales price based on the

features of the house.
Which values should you use? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

Setting	Value
Prediction task	Classification Forecasting Regression Outlier
Target column	Price Bedrooms Size HasGarage HomeType

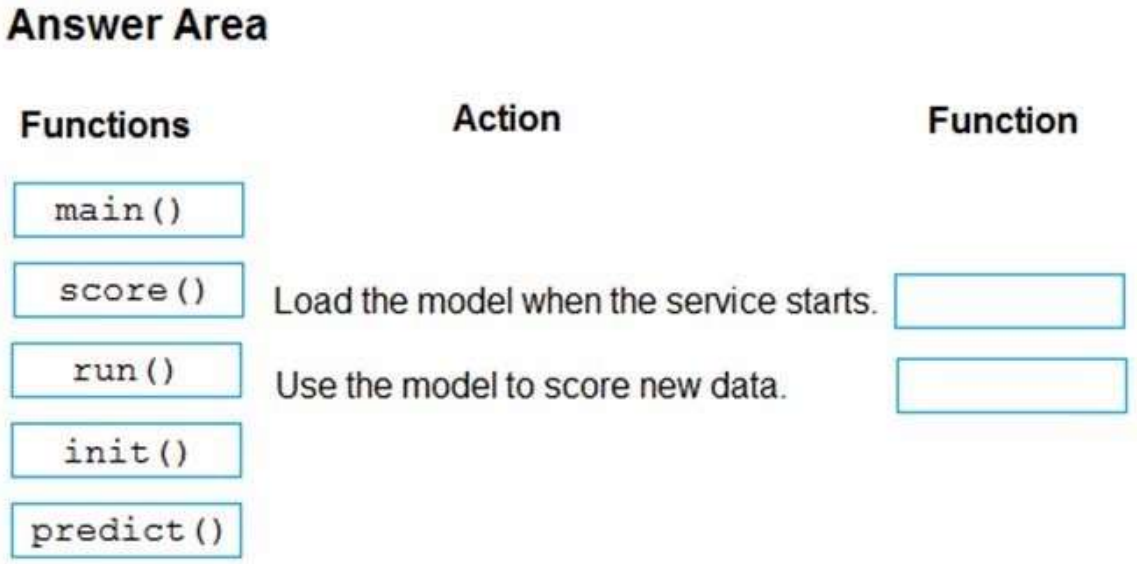
Answer:

Answer Area

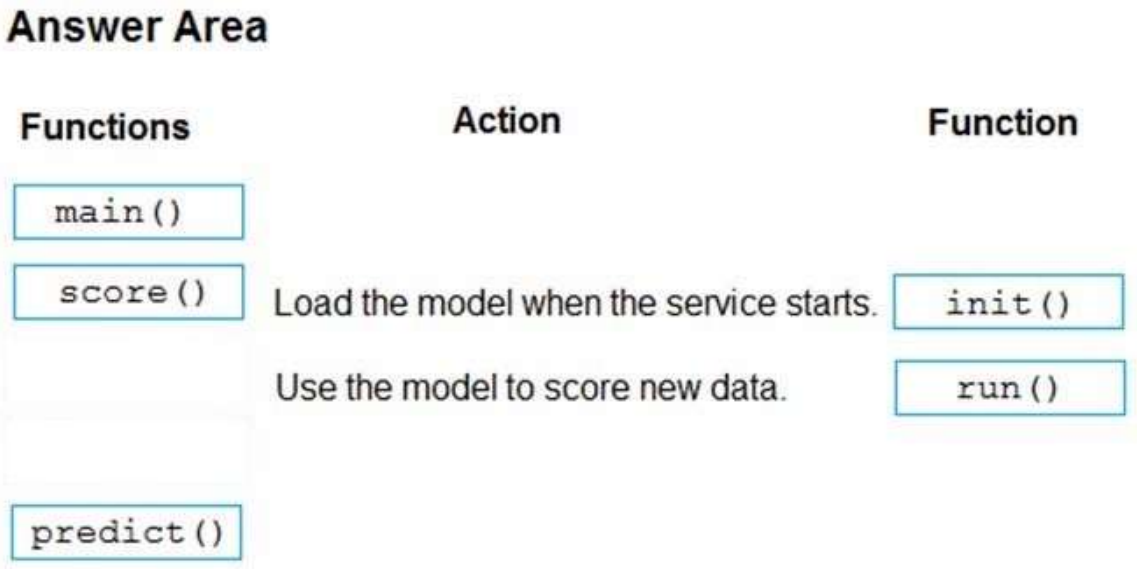
Setting	Value
Prediction task	Classification Forecasting **Regression** Outlier
Target column	**Price** Bedrooms Size HasGarage HomeType

Explanation:
Box 1: Regression
Regression is a supervised machine learning technique used to predict numeric values.
Box 2: Price
Reference:
https://docs.microsoft.com/en-us/learn/modules/create-regression-model-azure-machine-learning-designer

QUESTION 226
Drag and Drop Question
You use Azure Machine Learning to deploy a model as a real-time web service.
You need to create an entry script for the service that ensures that the model is loaded when the service starts and is used to score new data as it is received.
Which functions should you include in the script? To answer, drag the appropriate functions to the correct actions. Each function may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:
Box 1: init()
The entry script has only two required functions, init() and run(data). These functions are used to initialize the service at startup and run the model using request data passed in by a client. The rest of the script handles loading and running the model(s).
Box 2: run()
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-existing-model

QUESTION 227

Hotspot Question
You create an Azure Machine Learning workspace and set up a development environment. You plan to train a deep neural network (DNN) by using the Tensorflow framework and by using estimators to submit training scripts.
You must optimize computation speed for training runs.
You need to choose the appropriate estimator to use as well as the appropriate training compute target configuration.
Which values should you use? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

Parameter	Value
Estimator	Estimator / SKLearn / PyTorch / Tensorflow / Chainer
Training compute	12 vCPU, 48 GB memory, 96 GB SSD / 12 vCPU, 112 GB memory, 680 GB SSD, 2 GPU, 24 GB GPU memory / 16 vCPU, 128 GB memory, 160 GB HDD, 80 GB NVME disk (4000 MBps) / 44 vCPU, 352 GB memory, 3.4 GHz CPU frequency all cores

Answer:

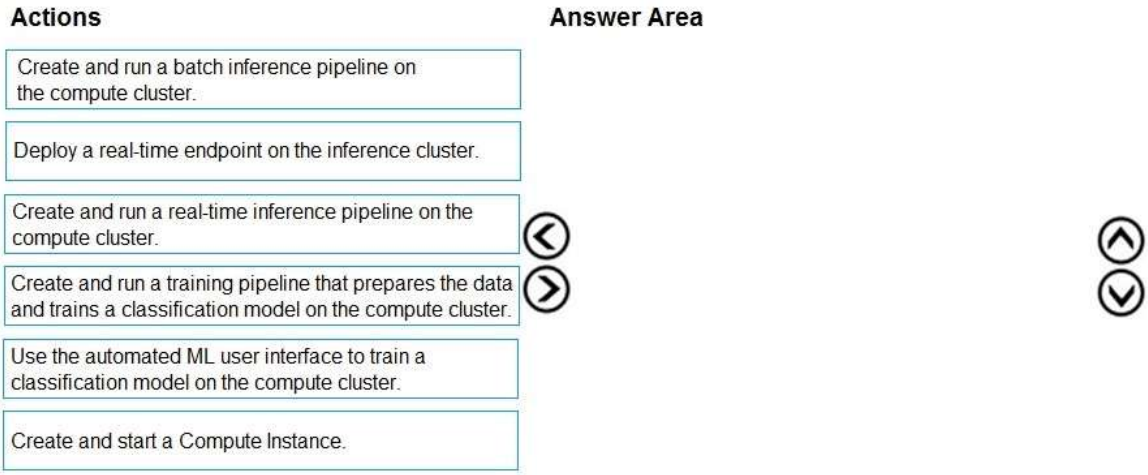
Answer Area

Parameter	Value
Estimator	Estimator / SKLearn / PyTorch / **Tensorflow** / Chainer
Training compute	12 vCPU, 48 GB memory, 96 GB SSD / **12 vCPU, 112 GB memory, 680 GB SSD, 2 GPU, 24 GB GPU memory** / 16 vCPU, 128 GB memory, 160 GB HDD, 80 GB NVME disk (4000 MBps) / 44 vCPU, 352 GB memory, 3.4 GHz CPU frequency all cores

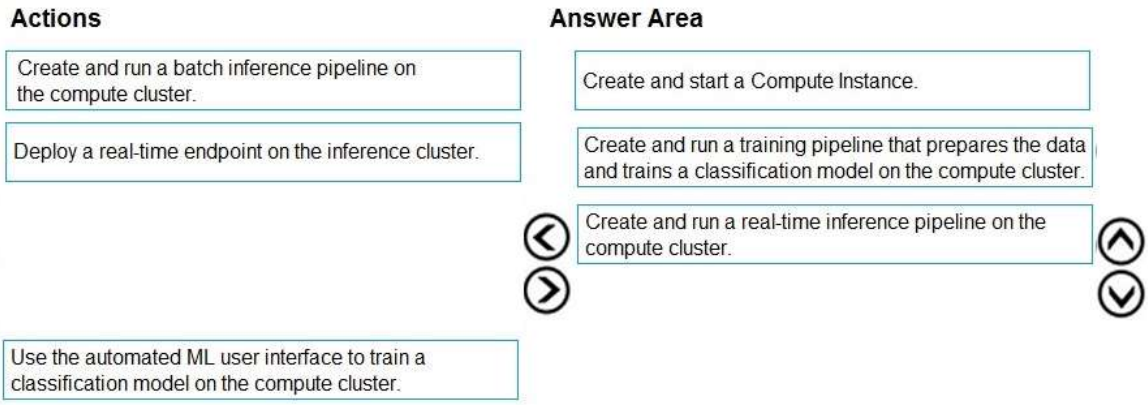
Explanation:
Box 1: Tensorflow
TensorFlow represents an estimator for training in TensorFlow experiments.
Box 2: 12 vCPU, 112 GB memory..,2 GPU,..
Use GPUs for the deep neural network.
Reference:
https://docs.microsoft.com/en-us/python/api/azureml-train-core/azureml.train.dnn

QUESTION 228
Drag and Drop Question
You have an Azure Machine Learning workspace that contains a CPU-based compute cluster and an Azure Kubernetes Services (AKS) inference cluster. You create a tabular dataset containing data that you plan to use to

create a classification model.
You need to use the Azure Machine Learning designer to create a web service through which client applications can consume the classification model by submitting new data and getting an immediate prediction as a response.
Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



Explanation:
Step 1: Create and start a Compute Instance
To train and deploy models using Azure Machine Learning designer, you need compute on which to run the training process, test the model, and host the model in a deployed service.
There are four kinds of compute resource you can create:
Compute Instances: Development workstations that data scientists can use to work with data and models. Compute Clusters: Scalable clusters of virtual machines for on-demand processing of experiment code. Inference Clusters: Deployment targets for predictive services that use your trained models. Attached Compute: Links to existing Azure compute resources, such as Virtual Machines or Azure Databricks clusters.
Step 2: Create and run a training pipeline..
After you've used data transformations to prepare the data, you can use it to train a machine learning model. Create and run a training pipeline
Step 3: Create and run a real-time inference pipeline
After creating and running a pipeline to train the model, you need a second pipeline that performs the same data transformations for new data, and then uses the trained model to inference (in other words, predict) label values based on its features. This pipeline will form the basis for a predictive service that you can publish for applications to use.
Reference:
https://docs.microsoft.com/en-us/learn/modules/create-classification-model-azure-machine-learning-designer/

QUESTION 229
Hotspot Question

You are running a training experiment on remote compute in Azure Machine Learning.
The experiment is configured to use a conda environment that includes the mlflow and azureml-contrib-run packages.
You must use MLflow as the logging package for tracking metrics generated in the experiment.
You need to complete the script for the experiment.
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

```
import numpy as np
# Import library to log metrics
```

[▼]
- from azureml.core import Run
- import mlflow
- import logging

```
# Start logging for this run
```

[▼]
- run = Run.get_context()
- mlflow.start_run()
- logger = logging.getLogger('Run')

```
reg_rate = 0.01
# Log the reg_rate metric
```

[▼]
- run.log('reg_rate', np.float(reg_rate))
- mlflow.log_metric('reg_rate', np.float(reg_rate))
- logger.info(np.float(reg_rate))

```
# Stop logging for this run
```

[▼]
- run.complete()
- mlflow.end_run()
- logger.setLevel(logging.INFO)

Answer:

Answer Area

```
import numpy as np
# Import library to log metrics
```

[from azureml.core import Run | import mlflow | import logging]

```
# Start logging for this run
```

[run = Run.get_context() | mlflow.start_run() | logger = logging.getLogger('Run')]

```
reg_rate = 0.01
# Log the reg_rate metric
```

[run.log('reg_rate', np.float(reg_rate)) | mlflow.log_metric('reg_rate', np.float(reg_rate)) | logger.info(np.float(reg_rate))]

```
# Stop logging for this run
```

[run.complete() | mlflow.end_run() | logger.setLevel(logging.INFO)]

Explanation:
Box 1: import mlflow
Import the mlflow and Workspace classes to access MLflow's tracking URI and configure your workspace.
Box 2: mlflow.start_run()
Set the MLflow experiment name with set_experiment() and start your training run with start_run().
Box 3: mlflow.log_metric(' ..')
Use log_metric() to activate the MLflow logging API and begin logging your training run metrics.
Box 4: mlflow.end_run()
Close the run:
run.endRun()
Reference:
https://docs.microsoft.com/en-us/azure/machine-learning/how-to-use-mlflow