TEST 4 - QUESTION: 1/50

You need to implement a model development strategy to determine a user's tendency to respond to an ad. Which technique should you use?

SCENARIO

Case study Overview You are a data scientist in a company that provides data science for professional sporting events. Models will use global and local market data to meet the following business goals: Understand sentiment of mobile device users at sporting events based on audio from crowd reactions. Assess a user's tendency to respond to an advertisement. Customize styles of ads served on mobile devices. Use video to detect penalty events Current environment Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared using social media. The images and videos will have varying sizes and formats. The data available for model building comprises of seven years of sporting event media. The sporting event media includes; recorded video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events. Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats. Penalty detection and sentiment Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection. Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation. Notebooks must execute with the same code on new Spark instances to recode only the source of the data. Global penalty detection models must be trained by using dynamic runtime graph computation during training. Local penalty detection models must be written by using BrainScript. Experiments for local crowd sentiment models must combine local penalty detection data. Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds. All shared features for local models are continuous variables. Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics available. Advertisements During the initial weeks in production, the following was observed: Ad response rated declined. Drops were not consistent across ad styles. The distribution of features across training and production data are not consistent Analysis shows that, of the 100 numeric features on user location and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue

is to engineer 10 linearly uncorrelated features. Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models. All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow. Audio samples show that the length of a catch phrase varies between 25%-47% depending on region The performance of the global penalty detection models shows lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases. Ad response models must be trained at the beginning of each event and applied during the sporting event. Market segmentation models must optimize for similar ad response history. Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. Ad response models must support non-linear boundaries of features. The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%. The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	2
	1	2	1

The ad propensity model uses proposed cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown

in the following diagram:

A

Use a Relative Expression Split module to partition the data based on distance travelled to the event.

B

Use a Relative Expression Split module to partition the data based on centroid distance.

C

Use a Split Rows module to partition the data based on distance travelled to the event.

D

Use a Split Rows module to partition the data based on centroid distance.

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: Split Data partitions the rows of a dataset into two distinct sets. The Relative Expression Split option in the Split Data module of Azure Machine Learning Studio is helpful when you need to divide a dataset into training and testing datasets using a numerical expression. Relative Expression Split: Use this option whenever you want to apply a condition to a number column. The number could be a date/time field, a column containing age or dollar amounts, or even a percentage. For example, you might want to divide your data set depending on the cost of the items, group people by age ranges, or separate data by a calendar date. Scenario: Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. The distribution of features across training and production data are not consistent Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/split-data

TEST 4 - QUESTION: 2/50

You use the Azure Machine Learning Python SDK to define a pipeline to train a model. The data used to train the model is read from a folder in a datastore. You need to ensure the pipeline runs automatically whenever the data in the folder changes. What should you do?

A

Create a ScheduleRecurrance object with a Frequency of auto . Use the object to create a Schedule for the pipeline

B

Create a PipelineParameter with a default value that references the location where the training data is stored

C

Set the regenerate_outputs property of the pipeline to True

D

Create a Schedule for the pipeline. Specify the datastore in the datastore property, and the folder containing the training data in the path_on_datastore property

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-trigger-published-pipeline

TEST 4 - QUESTION: 3/50

You create a Python script that runs a training experiment in Azure Machine Learning. The script uses the Azure Machine Learning SDK for Python. You must add a statement that retrieves the names of the logs and outputs generated by the script. You need to reference a Python class object from the SDK for the statement. Which class object should you use?

A

Experiment

B

ScriptRunConfig

C

Workspace

D

Run

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Explanation: A run represents a single trial of an experiment. Runs are used to monitor the asynchronous execution of a trial, log metrics and store output of the trial, and to analyze results and access artifacts generated by the trial. The run Class get_all_logs method downloads all logs for the run to a directory. Incorrect Answers: A: A run represents a single trial of an experiment. Runs are used to monitor the asynchronous execution of a trial, log metrics and store output of the trial, and to analyze results and access artifacts generated by the trial. B: A ScriptRunConfig packages together the configuration information needed to submit a run in Azure ML, including the script, compute target, environment, and any distributed job-specific configs. Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.run(class)

TEST 4 - QUESTION: 4/50 *SELECT MULTIPLE*

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.

SCENARIO

Case study This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided. To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study. At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section. To start the case study To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question. Overview You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules. Datasets There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

Column heading	Description
CapitaCrimeRate	per capita crime rate by town
Zoned	proportion of residential land zoned for lots over 25.000 square feet
NonRetailAcres	proportion of retail business acres per town
NextToRiver	proximity of a property to the river
NitrogenOxideConcentration	nitric oxides concentration (parts per 10 million)
AvgRoomsPerHouse	average number of rooms per dwelling
Age	proportion of owner-occupied units built prior to 1940
DistanceToEmploymentCenter	weighted distances to employment centers
AccessibilityToHighway	index of accessibility to radial highways to a value of two decimal places
Tax	full value property tax rate per $10,000
PupilTeacherRatio	pupil to teacher ratio by town
ProfessionalClass	professional class percentage
LowerStatus	percentage lower status of the population
MedianValue	median value of owner-occupied homes in $1000s

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format. Data issues Missing values The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values. Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail. Model fit The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting. Experiment requirements You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset. You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships. You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns. Model training Permutation Feature Importance Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model. Hyperparameters You must configure

hyperparameters in the model learning process to speed the learning phase . In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful. You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs. Testing You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio. Cross-validation You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city's main river. You must complete this task before the data goes through the sampling process. Linear regression module When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent. Data visualization You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results. You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another.

☐ A
Summarize Data

☐ B
Replace Discrete Values

☐ C
Build Counting Transform

☐ D
Clip Values

☐ E
Create Scatterplot

CORRECT ANSWERS: A,D,E

KEEP OPEN

EXPLANATION:

Explanation: B: To have a global view, the summarize data module can be used. Add the module and connect it to the data set that needs to be visualized. A: One way to quickly identify Outliers visually is to create scatter plots. C: The easiest way to treat the outliers in Azure ML is to use the Clip Values module. It can identify and optionally replace data values that are above or below a specified threshold. You can use the Clip Values module in Azure Machine Learning Studio, to identify and optionally replace data values that are above or below a specified threshold. This is useful when you want to remove outliers or replace them with a mean, a constant, or other substitute value. Reference: https://blogs.msdn.microsoft.com/azuredev/2017/05/27/data-cleansing-tools-in-azure-machine-learning/ https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/clip-values

TEST 4 - QUESTION: 5/50

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_output()[1]

☐ B

best_model = automl_run.get_metrics()

☐ C

best_model = automl_run.get_file_names()[1]

☐ D

best_model = automl_run.get_details()

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: The get_output method returns the best run and the fitted model.
Reference: https://notebooks.azure.com/azureml/projects/azureml-getting-started/html/how-to-use-azureml/automated-machine-learning/classification/auto-ml-classification.ipynb

TEST 4 - QUESTION: 6/50

HOTSPOT 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.

CHECK BELOW THE RIGHT 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)

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

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

TEST 4 - QUESTION: 7/50

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 have a Python script named train.py in a local folder named scripts. The script trains a regression model by using scikit-learn. The script includes code to load a training data file which is also located in the scripts folder. You must run the script as an Azure ML experiment on a compute cluster named aml-compute. You need to configure the run to ensure that the environment includes the required packages for model training. You have instantiated a variable named aml-compute that references the target compute cluster. Solution: Run the following code:

```
from azureml.train.sklearn import SKLearn
sk_est = SKLearn(source_directory='./scripts',
 compute_target=aml-compute,
 entry_script='train.py')
```


Does the solution meet the goal?

A

No

B

Yes

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: The scikit-learn estimator provides a simple way of launching a scikit-learn training job on a compute target. It is implemented through the SKLearn class, which can be used to support single-node CPU training. Example: from azureml.train.sklearn import SKLearn } estimator = SKLearn(source_directory=project_folder, compute_target=compute_target, entry_script='train_iris.py') Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-scikit-learn

TEST 4 - QUESTION: 8/50

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 have a Python script named train.py in a local folder named scripts. The script trains a regression model by using scikit-learn. The script includes code to load a training data file which is also located in the scripts folder. You must run the script as an Azure ML experiment on a compute cluster named aml-compute. You need to configure the run to ensure that the environment includes the required packages for model training. You have instantiated a variable named aml-compute that references the target compute cluster. Solution: Run the following code:

```
from azureml.train.estimator import Estimator
sk_est = Estimator(source_directory='./scripts',
 compute_target=aml-compute,
 entry_script='train.py',
 conda_packages=['scikit-learn'])
```


Does the solution meet the goal?

A

Yes

B

No

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: The scikit-learn estimator provides a simple way of launching a scikit-learn training job on a compute target. It is implemented through the SKLearn class, which can be used to support single-node CPU training. Example: from azureml.train.sklearn import SKLearn } estimator = SKLearn(source_directory=project_folder, compute_target=compute_target, entry_script='train_iris.py') Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-scikit-learn

TEST 4 - QUESTION: 9/50

You are creating a classification model for a banking company to identify possible instances of credit card fraud. You plan to create the model in Azure Machine Learning by using automated machine learning. The training dataset that you are using is highly unbalanced. You need to evaluate the classification model. Which primary metric should you use?

A
normalized_root_mean_squared_error

B
normalized_mean_absolute_error

C
spearman_correlation

D
AUC_weighted

E
accuracy

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Explanation: AUC_weighted is a Classification metric. Note: AUC is the Area under the Receiver Operating Characteristic Curve. Weighted is the arithmetic mean of the score for each class, weighted by the number of true instances in each class. Incorrect Answers: A: normalized_mean_absolute_error is a regression metric, not a classification metric. C: When comparing approaches to imbalanced classification problems, consider using metrics beyond accuracy such as recall, precision, and AUROC. It may be that switching the metric you optimize for during parameter selection or model selection is enough to provide desirable performance detecting the minority class. D: normalized_root_mean_squared_error is a regression metric, not a classification metric. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-understand-automated-ml

TEST 4 - QUESTION: 10/50

HOTSPOT 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.

CHECK BELOW THE RIGHT ANSWER

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

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

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

TEST 4 - QUESTION: 11/50

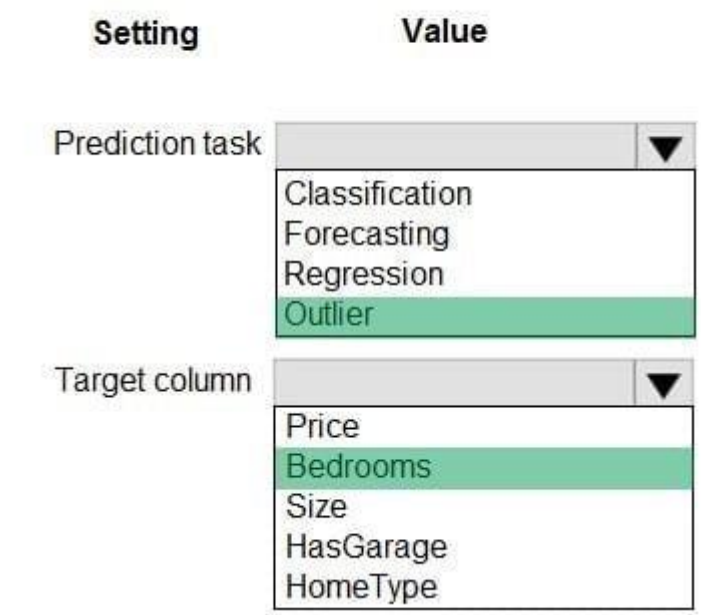
HOTSPOT You have a dataset that includes home sales data for a city. The dataset includes the following columns.

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.

CHECK BELOW THE RIGHT ANSWER

Answer Area



CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

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

TEST 4 - QUESTION: 12/50

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

Bayesian sampling

C

Random sampling

D

No sampling

CORRECT ANSWERS: B,C

KEEP OPEN

EXPLANATION:

C: Bayesian sampling is based on the Bayesian optimization algorithm and makes intelligent choices on the hyperparameter values to sample next. It picks the sample based on how the previous samples performed, such that the new sample improves the reported primary metric. Bayesian sampling does not support any early termination policy Example: from azureml.train.hyperdrive import BayesianParameterSampling from azureml.train.hyperdrive import uniform, choice param_sampling = BayesianParameterSampling({ "learning_rate": uniform(0.05, 0.1), "batch_size": choice(16, 32, 64, 128) }) D: In random sampling, hyperparameter values are randomly selected from the defined search space. Random sampling allows the search space to include both discrete and continuous hyperparameters. Incorrect Answers: B: Grid sampling can be used if your hyperparameter space can be defined as a choice among discrete values and if you have sufficient budget to exhaustively search over all values in the defined search space. Additionally, one can use automated early termination of poorly performing runs, which reduces wastage of resources. Example, the following space has a total of six samples: from azureml.train.hyperdrive import GridParameterSampling from

azureml.train.hyperdrive import choice param_sampling = GridParameterSampling({ "num_hidden_layers": choice(1, 2, 3), "batch_size": choice(16, 32) }) Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters

TEST 4 - QUESTION: 13/50

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 configure the Tune Model Hyperparameters module. Which two values should you use? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- [ ] A

Learning Rate

- [ ] B

Number of hidden nodes

- [ ] C

Number of learning iterations

- [ ] D

The type of the normalizer

- [ ] E

Hidden layer specification

CORRECT ANSWERS: C,E

KEEP OPEN

EXPLANATION:

Explanation: D: For Number of learning iterations, specify the maximum number of times the algorithm should process the training cases. E: For Hidden layer specification, select the type of network architecture to create. Between the input and output layers you can insert multiple hidden layers. Most predictive tasks can be accomplished easily with only one or a few hidden layers. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-neural-network

TEST 4 - QUESTION: 14/50

You plan to run a Python script as an Azure Machine Learning experiment. The script contains the following code:

```
import os, argparse, glob
from azureml.core import Run

parser = argparse.ArgumentParser()
parser.add_argument('--input-data', type=str, dest='data_folder')
args = parser.parse_args()
data_path = args.data_folder
file_paths = glob.glob(data_path + "/*.jpg")
```

You must specify a file dataset as an input to the script. The dataset consists of multiple large image files and must be streamed directly from its source. You need to write code to define a ScriptRunConfig object for the experiment and pass the ds dataset as an argument. Which code segment should you use?

A

arguments = ['--data-data', ds]

B

arguments = ['--input-data', ds.to_pandas_dataframe()]

C

arguments = ['--input-data', ds.as_mount()]

D

arguments = ['--input-data', ds.as_download()]

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: If you have structured data not yet registered as a dataset, create a TabularDataset and use it directly in your training script for your local or remote experiment. To load the TabularDataset to pandas DataFrame df = dataset.to_pandas_dataframe() Note: TabularDataset represents data in a tabular format created by parsing the provided file or list of files. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-with-datasets

TEST 4 - QUESTION: 15/50

You are building a recurrent neural network to perform a binary classification. You review the training loss, validation loss, training accuracy, and validation accuracy for each training epoch. You need to analyze model performance. You need to identify whether the classification model is overfitted. Which of the following is correct?

A

The training loss stays constant and the validation loss stays on a constant value and close to the training loss value when training the model.

B

The training loss decreases while the validation loss increases when training the model.

C

The training loss stays constant and the validation loss decreases when training the model.

D

The training loss increases while the validation loss decreases when training the model.

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: An overfit model is one where performance on the train set is good and continues to improve, whereas performance on the validation set improves to a point and then begins to degrade. Reference: https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/

TEST 4 - QUESTION: 16/50

HOTSPOT You create an Azure Databricks workspace and a linked Azure Machine Learning workspace. You have the following Python code segment in the Azure Machine Learning workspace: import mlflow import mlflow.azureml import azureml.mlflow import azureml.core from azureml.core import Workspace subscription_id = 'subscription_id' resourse_group = 'resource_group_name' workspace_name = 'workspace_name' ws = Workspace.get(name=workspace_name, subscription_id=subscription_id, resource_group=resource_group) experimentName = "/Users/{user_name}/{experiment_folder}/{experiment_name}" mlflow.set_experiment(experimentName) uri = ws.get_mlflow_tracking_uri() mlflow.set_tracking_uri(uri) Instructions: For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

CHECK BELOW THE RIGHT ANSWER

Answer Area

	Yes	No
A resource group and Azure Machine Learning workspace will be created.	○	○
An Azure Databricks experiment will be tracked only in the Azure Machine Learning workspace.	○	○
The epoch loss metric is set to be tracked.	○	○

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

Explanation: Box 1: No The Workspace.get method loads an existing workspace without using configuration files. ws = Workspace.get(name="myworkspace", subscription_id='<azure-subscription-id>', resource_group='myresourcegroup') Box 2: Yes MLflow Tracking with Azure Machine Learning lets you store the logged metrics and artifacts from your local runs into your Azure Machine Learning workspace. The get_mlflow_tracking_uri() method assigns a unique tracking URI address to the workspace, ws, and set_tracking_uri() points the MLflow tracking URI to that address. Box 3: Yes Note: In Deep Learning, epoch means the total dataset is passed forward and backward in a neural network once. Reference:

TEST 4 - QUESTION: 17/50

You create a script that trains a convolutional neural network model over multiple epochs and logs the validation loss after each epoch. The script includes arguments for batch size and learning rate. You identify a set of batch size and learning rate values that you want to try. You need to use Azure Machine Learning to find the combination of batch size and learning rate that results in the model with the lowest validation loss. What should you do?

A

Create a PythonScriptStep object for the script and run it in a pipeline

B

Run the script in an experiment based on an AutoMLConfig object

C

Run the script in an experiment based on a HyperDriveConfig object

D

Use the Automated Machine Learning interface in Azure Machine Learning studio

E

Run the script in an experiment based on a ScriptRunConfig object

CORRECT ANSWER: C

KEEP OPEN

EXPLANATION:

Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters

TEST 4 - QUESTION: 18/50

HOTSPOT You need to configure the Edit Metadata module so that the structure of the datasets match. Which configuration options should you select? To answer, select the appropriate options in the answer area. NOTE : Each correct selection is worth one point.

SCENARIO

Case study This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided. To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study. At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section. To start the case study To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question. Overview You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules. Datasets There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

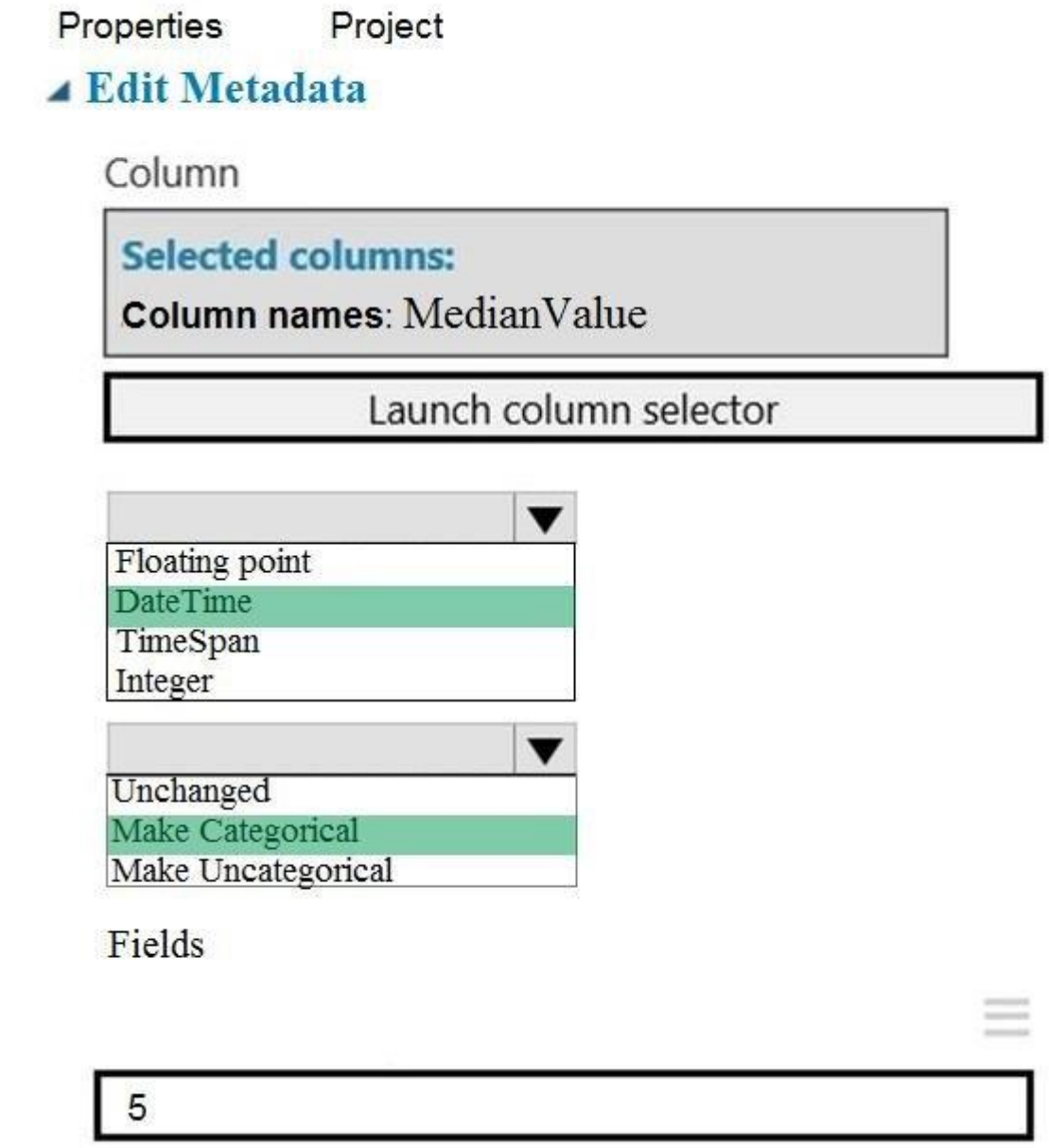
Column heading	Description
CapitaCrimeRate	per capita crime rate by town
Zoned	proportion of residential land zoned for lots over 25.000 square feet
NonRetailAcres	proportion of retail business acres per town
NextToRiver	proximity of a property to the river
NitrogenOxideConcentration	nitric oxides concentration (parts per 10 million)
AvgRoomsPerHouse	average number of rooms per dwelling
Age	proportion of owner-occupied units built prior to 1940
DistanceToEmploymentCenter	weighted distances to employment centers
AccessibilityToHighway	index of accessibility to radial highways to a value of two decimal places
Tax	full value property tax rate per $10,000
PupilTeacherRatio	pupil to teacher ratio by town
ProfessionalClass	professional class percentage
LowerStatus	percentage lower status of the population
MedianValue	median value of owner-occupied homes in $1000s

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format. Data issues Missing values The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values. Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail. Model fit The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting. Experiment requirements You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset. You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships. You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns. Model training Permutation Feature Importance Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model. Hyperparameters You must configure

hyperparameters in the model learning process to speed the learning phase . In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful. You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs. Testing You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio. Cross-validation You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city's main river. You must complete this task before the data goes through the sampling process. Linear regression module When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent. Data visualization You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results. You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another.

CHECK BELOW THE RIGHT ANSWER

Answer Area



CORRECT ANSWER:
KEEP OPEN

EXPLANATION:

Explanation: Box 1: Floating point Need floating point for Median values. Scenario: An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format. Box 2: Unchanged Note: Select the Categorical option to specify that the values in the selected columns should

be treated as categories. For example, you might have a column that contains the numbers 0,1 and 2, but know that the numbers actually mean "Smoker", "Non smoker" and "Unknown". In that case, by flagging the column as categorical you can ensure that the values are not used in numeric calculations, only to group data.

TEST 4 - QUESTION: 19/50

HOTSPOT You register the following versions of a model.

Model name	Model version	Tags	Properties
healthcare_model	3	'Training context':'CPU Compute'	value:87.43
healthcare_model	2	'Training context':'CPU Compute'	value:54.98
healthcare_model	1	'Training context':'CPU Compute'	value:23.56

You use the Azure ML Python SDK to run a training experiment. You use a variable named run to reference the experiment run. After the run has been submitted and completed, you run the following code:

```
run.register_model(model_path='outputs/model.pkl',
 model_name='healthcare_model',
 tags={'Training context':'CPU Compute'} )
```

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

CHECK BELOW THE RIGHT ANSWER

Answer Area

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

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where

TEST 4 - QUESTION: 20/50

You are building a machine learning model for translating English language textual content into French language textual content. You need to build and train the machine learning model to learn the sequence of the textual content. Which type of neural network should you use?

A

Generative Adversarial Networks (GANs)

B

Convolutional Neural Networks (CNNs)

C

Multilayer Perceptions (MLPs)

D

Recurrent Neural Networks (RNNs)

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Explanation: To translate a corpus of English text to French, we need to build a recurrent neural network (RNN). Note: RNNs are designed to take sequences of text as inputs or return sequences of text as outputs, or both. They're called recurrent because the network's hidden layers have a loop in which the output and cell state from each time step become inputs at the next time step. This recurrence serves as a form of memory. It allows contextual information to flow through the network so that relevant outputs from previous time steps can be applied to network operations at the current time step. Reference: https://towardsdatascience.com/language-translation-with-rnns-d84d43b40571

TEST 4 - QUESTION: 21/50

You need to implement a feature engineering strategy for the crowd sentiment local models. What should you do?

SCENARIO

Case study Overview You are a data scientist in a company that provides data science for professional sporting events. Models will use global and local market data to meet the following business goals: Understand sentiment of mobile device users at sporting events based on audio from crowd reactions. Assess a user's tendency to respond to an advertisement. Customize styles of ads served on mobile devices. Use video to detect penalty events Current environment Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared using social media. The images and videos will have varying sizes and formats. The data available for model building comprises of seven years of sporting event media. The sporting event media includes; recorded video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events. Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats. Penalty detection and sentiment Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection. Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation. Notebooks must execute with the same code on new Spark instances to recode only the source of the data. Global penalty detection models must be trained by using dynamic runtime graph computation during training. Local penalty detection models must be written by using BrainScript. Experiments for local crowd sentiment models must combine local penalty detection data. Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds. All shared features for local models are continuous variables. Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics available. Advertisements During the initial weeks in production, the following was observed: Ad response rated declined. Drops were not consistent across ad styles. The distribution of features across training and production data are not consistent Analysis shows that, of the 100 numeric features on user location and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue

is to engineer 10 linearly uncorrelated features. Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models. All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow. Audio samples show that the length of a catch phrase varies between 25%-47% depending on region The performance of the global penalty detection models shows lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases. Ad response models must be trained at the beginning of each event and applied during the sporting event. Market segmentation models must optimize for similar ad response history. Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. Ad response models must support non-linear boundaries of features. The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%. The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	2
	1	2	1

The ad propensity model uses proposed cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown

in the following diagram:

A

Apply a linear discriminant analysis.

B

Apply a Spearman correlation coefficient.

C

Apply an analysis of variance (ANOVA).

D

Apply a Pearson correlation coefficient.

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: The linear discriminant analysis method works only on continuous variables, not categorical or ordinal variables. Linear discriminant analysis is similar to analysis of variance (ANOVA) in that it works by comparing the means of the variables. Scenario: Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Experiments for local crowd sentiment models must combine local penalty detection data. All shared features for local models are continuous variables. Incorrect Answers: B: The Pearson correlation coefficient, sometimes called Pearson's R test, is a statistical value that measures the linear relationship between two variables. By examining the coefficient values, you can infer something about the strength of the relationship between the two variables, and whether they are positively correlated or negatively correlated. C: Spearman's correlation coefficient is designed for use with non-parametric and non-normally distributed data. Spearman's coefficient is a nonparametric measure of statistical dependence between two variables, and is sometimes denoted by the Greek letter rho. The

Spearman's coefficient expresses the degree to which two variables are monotonically related. It is also called Spearman rank correlation, because it can be used with ordinal variables. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/fisher-linear-discriminant-analysis https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/compute-linear-correlation

TEST 4 - QUESTION: 22/50

You need to implement a new cost factor scenario for the ad response models as illustrated in the performance curve exhibit. Which technique should you use?

SCENARIO

Case study Overview You are a data scientist in a company that provides data science for professional sporting events. Models will use global and local market data to meet the following business goals: Understand sentiment of mobile device users at sporting events based on audio from crowd reactions. Assess a user's tendency to respond to an advertisement. Customize styles of ads served on mobile devices. Use video to detect penalty events Current environment Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared using social media. The images and videos will have varying sizes and formats. The data available for model building comprises of seven years of sporting event media. The sporting event media includes; recorded video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events. Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats. Penalty detection and sentiment Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection. Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation. Notebooks must execute with the same code on new Spark instances to recode only the source of the data. Global penalty detection models must be trained by using dynamic runtime graph computation during training. Local penalty detection models must be written by using BrainScript. Experiments for local crowd sentiment models must combine local penalty detection data. Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds. All shared features for local models are continuous variables. Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics available. Advertisements During the initial weeks in production, the following was observed: Ad response rated declined. Drops were not consistent across ad styles. The distribution of features across training and production data are not consistent Analysis shows that, of the 100 numeric features on user location and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue

is to engineer 10 linearly uncorrelated features. Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models. All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow. Audio samples show that the length of a catch phrase varies between 25%-47% depending on region The performance of the global penalty detection models shows lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases. Ad response models must be trained at the beginning of each event and applied during the sporting event. Market segmentation models must optimize for similar ad response history. Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. Ad response models must support non-linear boundaries of features. The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%. The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	2
	1	2	1

The ad propensity model uses proposed cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown

in the following diagram:

A

Set the threshold to 0.5 and retrain if weighted Kappa deviates +/- 5% from 0.45.

B

Set the threshold to 0.05 and retrain if weighted Kappa deviates +/- 5% from 0.5.

C

Set the threshold to 0.2 and retrain if weighted Kappa deviates +/- 5% from 0.6.

D

Set the threshold to 0.75 and retrain if weighted Kappa deviates +/- 5% from 0.15.

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: Scenario: Performance curves of current and proposed cost factor scenarios are shown in the following diagram:

The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%.

TEST 4 - QUESTION: 23/50

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 create a model to forecast weather conditions based on historical data. You need to create a pipeline that runs a processing script to load data from a datastore and pass the processed data to a machine learning model training script. Solution: Run the following code:

```
data_store = Datastore.get(ws, "ml-data")
data_input = DataReference(
    datastore = data_store,
    data_reference_name = "training_data",
    path_on_datastore = "train/data.txt")
data_output = PipelineData("processed_data", datastore=datastore)
process_step = PythonScriptStep(script_name= "process.py",
    arguments=[ "- -data", data_input], outputs=[data_output],
    compute_target=aml_compute, source_directory=process_directory)
train_step = PythonScriptStep(script_name= "train.py",
      arguments=["- -data", data_output], inputs=[data_output],
      compute_target=aml_compute, source_directory=train_directory)
pipeline = Pipeline(workspace=ws, steps = [process_step, train_step])
```

Does the solution meet the goal?

A

Yes

B

No

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: The two steps are present: process_step and train_step Data_input correctly references the data in the data store. Note: Data used in pipeline can be produced by one step and consumed in another step by providing a PipelineData object as an output of one step and an input of one or more subsequent steps. PipelineData objects are also used when constructing Pipelines to describe step dependencies. To specify that a step requires the output of another step as input, use a PipelineData object in the constructor of

both steps. For example, the pipeline train step depends on the process_step_output output of the pipeline process step: from azureml.pipeline.core import Pipeline, PipelineData from azureml.pipeline.steps import PythonScriptStep datastore = ws.get_default_datastore() process_step_output = PipelineData("processed_data", datastore=datastore) process_step = PythonScriptStep(script_name="process.py", arguments=["--data_for_train", process_step_output], outputs=[process_step_output], compute_target=aml_compute, source_directory=process_directory) train_step = PythonScriptStep(script_name="train.py", arguments=["--data_for_train", process_step_output], inputs=[process_step_output], compute_target=aml_compute, source_directory=train_directory) pipeline = Pipeline(workspace=ws, steps=[process_step, train_step]) Reference: https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.pipelinedata?view=azure-ml-py

TEST 4 - QUESTION: 24/50

HOTSPOT You need to use the Python language to build a sampling strategy for the global penalty detection models. 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.

SCENARIO

Case study Overview You are a data scientist in a company that provides data science for professional sporting events. Models will use global and local market data to meet the following business goals: Understand sentiment of mobile device users at sporting events based on audio from crowd reactions. Assess a user's tendency to respond to an advertisement. Customize styles of ads served on mobile devices. Use video to detect penalty events Current environment Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared using social media. The images and videos will have varying sizes and formats. The data available for model building comprises of seven years of sporting event media. The sporting event media includes; recorded video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events. Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats. Penalty detection and sentiment Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection. Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation. Notebooks must execute with the same code on new Spark instances to recode only the source of the data. Global penalty detection models must be trained by using dynamic runtime graph computation during training. Local penalty detection models must be written by using BrainScript. Experiments for local crowd sentiment models must combine local penalty detection data. Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds. All shared features for local models are continuous variables. Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics available. Advertisements During the initial weeks in production, the following was observed: Ad response rated declined. Drops were not consistent across ad styles. The distribution of features across training and production data are not consistent Analysis shows that, of the 100 numeric features on user location

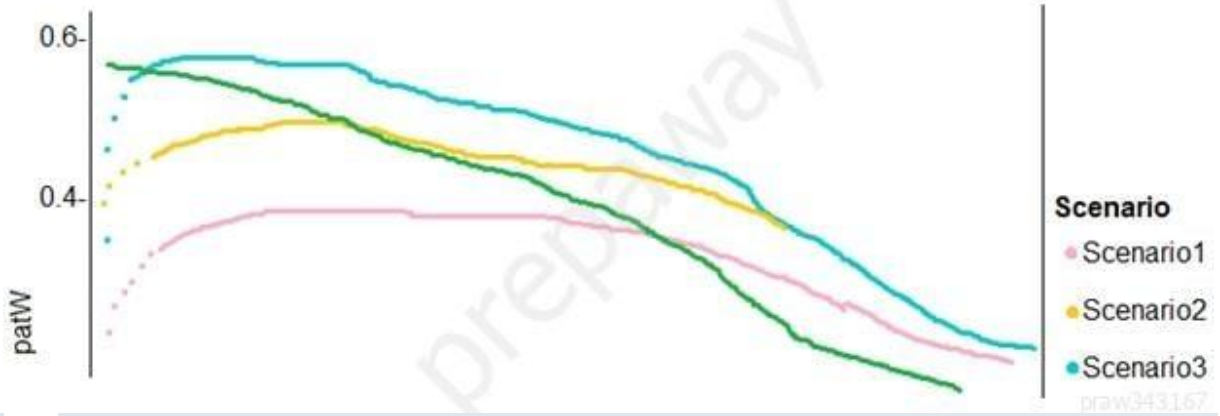
and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue is to engineer 10 linearly uncorrelated features. Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models. All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow. Audio samples show that the length of a catch phrase varies between 25%-47% depending on region The performance of the global penalty detection models shows lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases. Ad response models must be trained at the beginning of each event and applied during the sporting event. Market segmentation models must optimize for similar ad response history. Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. Ad response models must support non-linear boundaries of features. The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%. The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	2
	1	2	1

The ad propensity model uses proposed cost factors shown in the following diagram:

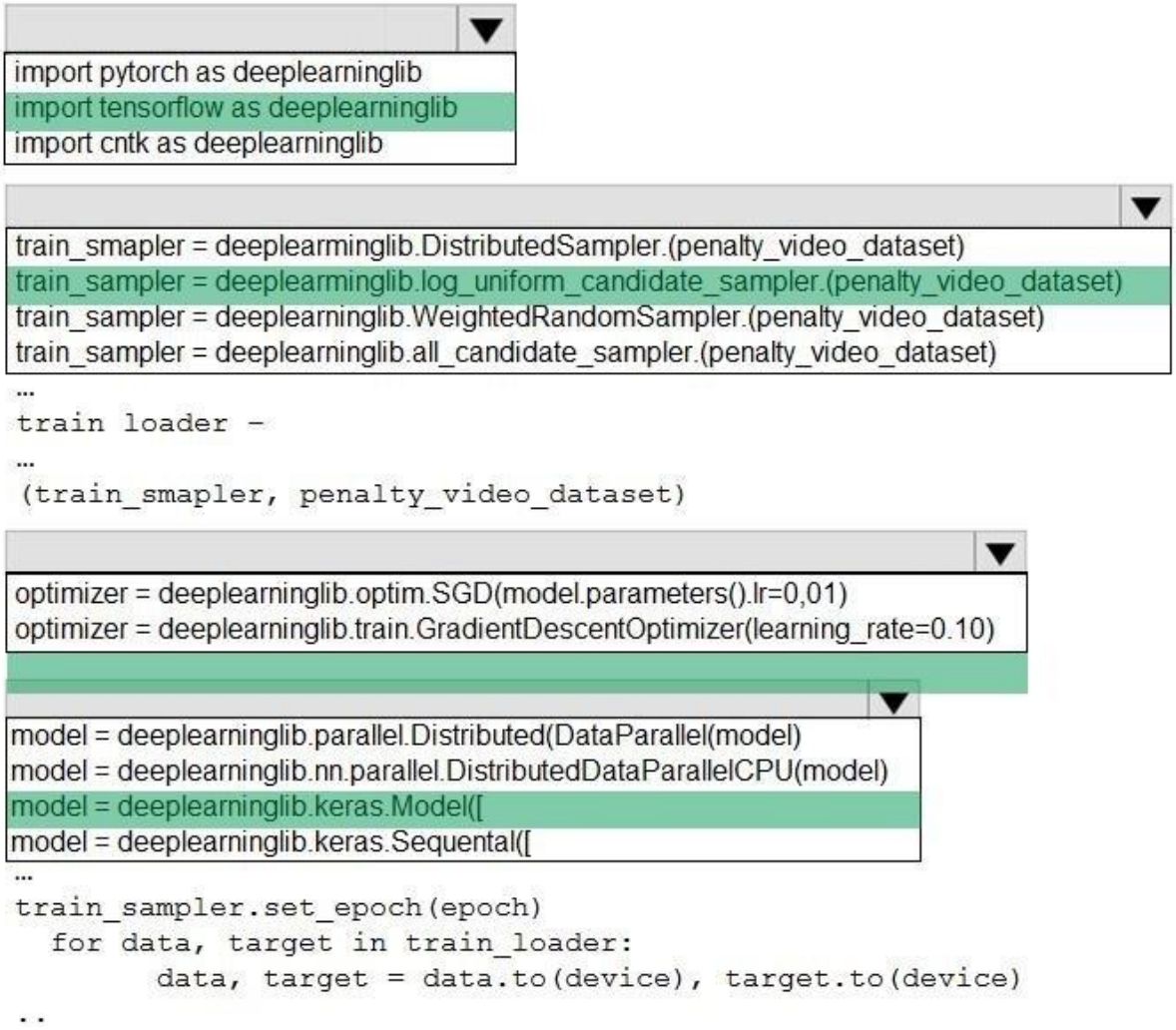
		Actual	
		1	0
Predicted	0	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown in the following diagram:



CHECK BELOW THE RIGHT ANSWER

Answer Area



CORRECT ANSWER:
KEEP OPEN

EXPLANATION:

Explanation: Box 1: import pytorch as deeplearninglib Box 2: ..DistributedSampler(Sampler).. DistributedSampler(Sampler): Sampler that restricts data loading to a subset of the dataset. It is especially useful in conjunction with class:`torch.nn.parallel.DistributedDataParallel`. In such case, each process can pass a DistributedSampler instance as a DataLoader sampler, and load a subset of the original dataset that is exclusive to it. Scenario: Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Box 3: optimizer = deeplearninglib.train. GradientDescentOptimizer(learning_rate=0.10) Incorrect Answers: ..SGD.. Scenario: All penalty detection models show inference phases

using a Stochastic Gradient Descent (SGD) are running too slow. Box 4: .. nn.parallel.DistributedDataParallel.. DistributedSampler(Sampler): The sampler that restricts data loading to a subset of the dataset. It is especially useful in conjunction with :class:`torch.nn.parallel.DistributedDataParallel`. Reference: https://github.com/pytorch/pytorch/blob/master/torch/utils/data/distributed.py

TEST 4 - QUESTION: 25/50

HOTSPOT You create a Python script named train.py and save it in a folder named scripts. The script uses the scikit-learn framework to train a machine learning model. You must run the script as an Azure Machine Learning experiment on your local workstation. You need to write Python code to initiate an experiment that runs the train.py script. 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.

CHECK BELOW THE RIGHT ANSWER

Answer Area

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

ws = Workspace.from_config()
py_sk = Environment('sklearn-training')
pkgs = CondaDependencies.create(pip_packages=['scikit-learn', 'azureml-defaults'])
py_sk.python.conda_dependencies = pkgs
script_config = ScriptRunConfig (
```

[▼] = 'scripts',
- script
- source_directory
- resume_from
- arguments

[▼] = 'train.py',
- script
- arguments
- environment
- compute_target

[▼] =py_sk)
- arguments
- resume_from
- environment
- compute_target

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

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

Explanation: Box 1: source_directory source_directory: A local directory containing code files needed for a run. Box 2: script Script: The file path relative to the source_directory of the script to be run. Box 3: environment Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.scriptrunconfig

TEST 4 - QUESTION: 26/50

HOTSPOT You use an Azure Machine Learning workspace. You create the following Python code:

```
from azureml.core import ScriptRunConfig
src = ScriptRunConfig(source_directory=project_folder,
                      script='train.py'
                      environment=myenv)
```

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

CHECK BELOW THE RIGHT ANSWER

Answer Area

Statements	Yes	No
The default environment will be created	○	○
The training script will run on local compute	○	○
A script run configuration runs a training script named `train.py` located in a directory defined by the `project_folder` variable	○	○

CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

Explanation: Box 1: No Environment is a required parameter. The environment to use for the run. If no environment is specified, azureml.core.runconfig.DEFAULT_CPU_IMAGE will be used as the Docker image for the run. The following example shows how to instantiate a new environment. from azureml.core import Environment myenv = Environment(name="myenv") Box 2: Yes Parameter compute_target: The compute target where training will happen. This can either be a ComputeTarget object, the name of an existing ComputeTarget, or the string "local". If no compute target is specified, your local machine will be used. Box 3: Yes Parameter source_directory. A local directory containing code files needed for a run. Parameter script. The file path relative to the source_directory of the script to be run. Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.scriptrunconfig https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.environment.environment

TEST 4 - QUESTION: 27/50

You create a multi-class image classification deep learning model that uses a set of labeled images. You create a script file named train.py that uses the PyTorch 1.3 framework to train the model. You must run the script by using an estimator. The code must not require any additional Python libraries to be installed in the environment for the estimator. The time required for model training must be minimized. You need to define the estimator that will be used to run the script. Which estimator type should you use?

A

SKLearn

B

PyTorch

C

Estimator

D

TensorFlow

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: For PyTorch, TensorFlow and Chainer tasks, Azure Machine Learning provides respective PyTorch, TensorFlow, and Chainer estimators to simplify using these frameworks. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-ml-models

TEST 4 - QUESTION: 28/50

HOTSPOT You need to replace the missing data in the AccessibilityToHighway columns. How should you configure the Clean Missing Data module? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

SCENARIO

Case study This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided. To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study. At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section. To start the case study To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question. Overview You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules. Datasets There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

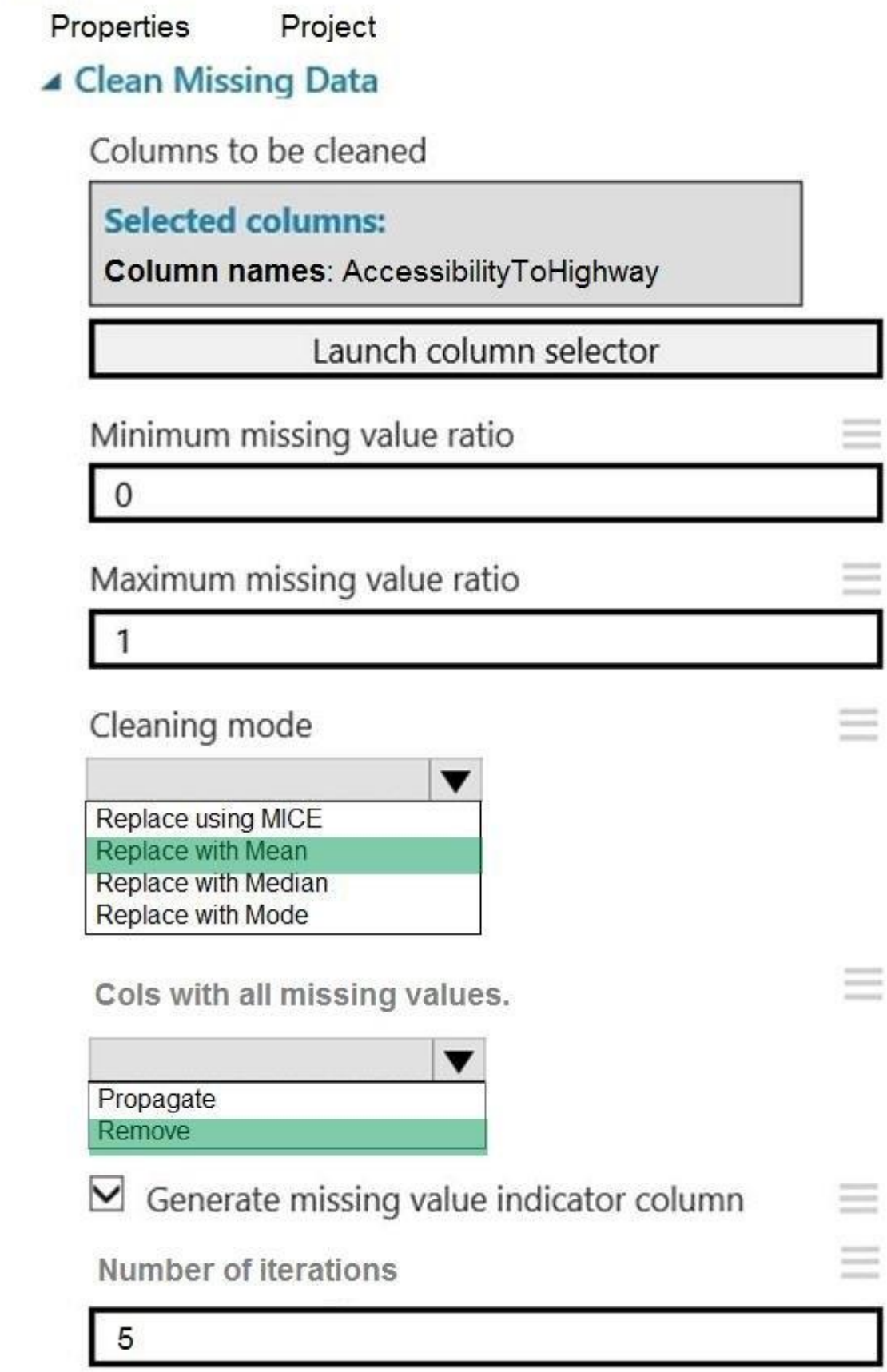
Column heading	Description
CapitaCrimeRate	per capita crime rate by town
Zoned	proportion of residential land zoned for lots over 25.000 square feet
NonRetailAcres	proportion of retail business acres per town
NextToRiver	proximity of a property to the river
NitrogenOxideConcentration	nitric oxides concentration (parts per 10 million)
AvgRoomsPerHouse	average number of rooms per dwelling
Age	proportion of owner-occupied units built prior to 1940
DistanceToEmploymentCenter	weighted distances to employment centers
AccessibilityToHighway	index of accessibility to radial highways to a value of two decimal places
Tax	full value property tax rate per $10,000
PupilTeacherRatio	pupil to teacher ratio by town
ProfessionalClass	professional class percentage
LowerStatus	percentage lower status of the population
MedianValue	median value of owner-occupied homes in $1000s

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format. Data issues Missing values The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values. Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail. Model fit The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting. Experiment requirements You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset. You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships. You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns. Model training Permutation Feature Importance Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model. Hyperparameters You must configure

hyperparameters in the model learning process to speed the learning phase . In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful. You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs. Testing You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio. Cross-validation You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city's main river. You must complete this task before the data goes through the sampling process. Linear regression module When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent. Data visualization You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results. You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another.

CHECK BELOW THE RIGHT ANSWER

Answer Area


Properties
Project
Clean Missing Data
Columns to be cleaned
Selected columns:
Column names: AccessibilityToHighway
Launch column selector
Minimum missing value ratio
0
Maximum missing value ratio
1
Cleaning mode
Replace using MICE
Replace with Mean
Replace with Median
Replace with Mode
Cols with all missing values.
Propagate
Remove
Generate missing value indicator column
Number of iterations
5

CORRECT ANSWER:
KEEP OPEN

EXPLANATION:

Explanation: Box 1: Replace using MICE Replace using MICE: For each missing value, this option assigns a new value, which is calculated by using a method described in the statistical literature as "Multivariate Imputation using Chained Equations" or "Multiple Imputation by Chained Equations". With a multiple imputation method, each variable with missing data is modeled conditionally using the other variables in the data before filling in the missing values. Scenario: The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values. Box 2: Propagate Cols with all missing values indicate if columns of all missing values should be preserved in the output. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/clean-missing-data

TEST 4 - QUESTION: 29/50

You create a binary classification model by using Azure Machine Learning Studio. You must tune hyperparameters by performing a parameter sweep of the model. The parameter sweep must meet the following requirements: iterate all possible combinations of hyperparameters minimize computing resources required to perform the sweep You need to perform a parameter sweep of the model. Which parameter sweep mode should you use?

A
Entire grid

B
Sweep clustering

C
Random grid

D
Random sweep

CORRECT ANSWER: C

KEEP OPEN

EXPLANATION:

Explanation: Maximum number of runs on random grid: This option also controls the number of iterations over a random sampling of parameter values, but the values are not generated randomly from the specified range; instead, a matrix is created of all possible combinations of parameter values and a random sampling is taken over the matrix. This method is more efficient and less prone to regional oversampling or undersampling. If you are training a model that supports an integrated parameter sweep, you can also set a range of seed values to use and iterate over the random seeds as well. This is optional, but can be useful for avoiding bias introduced by seed selection. Incorrect Answers: B: If you are building a clustering model, use Sweep Clustering to automatically determine the optimum number of clusters and other parameters. C: Entire grid: When you select this option, the module loops over a grid predefined by the system, to try different combinations and identify the best learner. This option is useful for cases where you don't know what the best parameter settings might be and want to try all possible combination of values. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/tune-model-hyperparameters

TEST 4 - QUESTION: 30/50

You have the following code. The code prepares an experiment to run a script:

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

ws = Workspace.from_config()
script_config = ScriptRunConfig(source_directory='experiment_files',
                                script='experiment.py')

script_experiment = Experiment(workspace=ws, name='script-experiment')
```

The experiment must be run on local computer using the default environment. You need to add code to start the experiment and run the script. Which code segment should you use?

A

run = script_experiment.start_logging()

B

run = script_experiment.submit(config=script_config)

C

ws.get_run(run_id=experiment.id)

D

run = Run(experiment=script_experiment)

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: The experiment class submit method submits an experiment and return the active created run. Syntax: submit(config, tags=None, **kwargs) Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.experiment.experiment

TEST 4 - QUESTION: 31/50

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 create a model to forecast weather conditions based on historical data. You need to create a pipeline that runs a processing script to load data from a datastore and pass the processed data to a machine learning model training script. Solution: Run the following code:

```
datastore = ws.get_default_datastore()
data_input = PipelineData("raw_data", datastore=rawdatastore)
data_output = PipelineData("processed_data", datastore=datastore)
process_step = PythonScriptStep(script_name="process.py",
 arguments=["--data_for_train", data_input],
 outputs=[data_output], compute_target=aml_compute,
 source_directory=process_directory)
train_step = PythonScriptStep(script_name="train.py",
 arguments=["--data_for_train", data_input], inputs=[data_output],
 compute_target=aml_compute, source_directory=train_directory)
pipeline = Pipeline(workspace=ws, steps=[process_step, train_step])
```

Does the solution meet the goal?

A

Yes

B

No

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: Note: Data used in pipeline can be produced by one step and consumed in another step by providing a PipelineData object as an output of one step and an input of one or more subsequent steps. Compare with this example, the pipeline train step depends on the process_step_output output of the pipeline process step: from azureml.pipeline.core import Pipeline, PipelineData from azureml.pipeline.steps import PythonScriptStep datastore = ws.get_default_datastore() process_step_output = PipelineData("processed_data", datastore=datastore) process_step =

PythonScriptStep(script_name="process.py", arguments=["--data_for_train", process_step_output], outputs=[process_step_output], compute_target=aml_compute, source_directory=process_directory) train_step = PythonScriptStep(script_name="train.py", arguments=["--data_for_train", process_step_output], inputs=[process_step_output], compute_target=aml_compute, source_directory=train_directory) pipeline = Pipeline(workspace=ws, steps=[process_step, train_step]) Reference: https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.pipelinedata?view=azure-ml-py

TEST 4 - QUESTION: 32/50 *SELECT MULTIPLE*

You run a script as an experiment in Azure Machine Learning. You have a Run object named run that references the experiment run. You must review the log files that were generated during the experiment run. You need to download the log files to a local folder for review. Which two code segments can you run to achieve this goal? Each correct answer presents a complete solution. NOTE: Each correct selection is worth one point.

☐ A

run.get_metrics()

☐ B

run.get_all_logs(destination='./runlogs')

☐ C

run.download_files(output_directory='./runfiles')

☐ D

run.get_file_names()

☐ E

run.get_details()

CORRECT ANSWERS: B,E

KEEP OPEN

EXPLANATION:

Explanation: The run Class get_all_logs method downloads all logs for the run to a directory. The run Class get_details gets the definition, status information, current log files, and other details of the run. Incorrect Answers: B: The run get_file_names list the files that are stored in association with the run. Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.run(class)

TEST 4 - QUESTION: 33/50

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 have a Python script named train.py in a local folder named scripts. The script trains a regression model by using scikit-learn. The script includes code to load a training data file which is also located in the scripts folder. You must run the script as an Azure ML experiment on a compute cluster named aml-compute. You need to configure the run to ensure that the environment includes the required packages for model training. You have instantiated a variable named aml-compute that references the target compute cluster. Solution: Run the following code:

```
from azureml.train.dnn import TensorFlow
sk_est = TensorFlow(source_directory='./scripts',
 compute_target=aml-compute,
 entry_script='train.py')
```


Does the solution meet the goal?

A

No

B

Yes

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: The scikit-learn estimator provides a simple way of launching a scikit-learn training job on a compute target. It is implemented through the SKLearn class, which can be used to support single-node CPU training. Example: from azureml.train.sklearn import SKLearn } estimator = SKLearn(source_directory=project_folder, compute_target=compute_target, entry_script='train_iris.py') Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-train-scikit-learn

TEST 4 - QUESTION: 34/50 *SELECT MULTIPLE*

You use the following code to define the steps for a pipeline: from azureml.core import Workspace, Experiment, Run from azureml.pipeline.core import Pipeline from azureml.pipeline.steps import PythonScriptStep ws = Workspace.from_config() . . . step1 = PythonScriptStep(name="step1", ...) step2 = PythonScriptsStep(name="step2", ...) pipeline_steps = [step1, step2] You need to add code to run the steps. Which two code segments can you use to achieve this goal? Each correct answer presents a complete solution. NOTE: Each correct selection is worth one point.

☐ A

pipeline = Pipeline(workspace=ws, steps=pipeline_steps) run = pipeline.submit(experiment_name='pipeline-experiment')

☐ B

pipeline = Pipeline(workspace=ws, steps=pipeline_steps) experiment = Experiment(workspace=ws, name='pipeline-experiment') run = experiment.submit(pipeline)

☐ C

run = Run(pipeline_steps)

☐ D

experiment = Experiment(workspace=ws, name='pipeline-experiment') run = experiment.submit(config=pipeline_steps)

CORRECT ANSWERS: A,B

KEEP OPEN

EXPLANATION:

Explanation: After you define your steps, you build the pipeline by using some or all of those steps. # Build the pipeline. Example: pipeline1 = Pipeline(workspace=ws, steps=[compare_models]) # Submit the pipeline to be run pipeline_run1 = Experiment(ws, 'Compare_Models_Exp').submit(pipeline1) Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-machine-learning-pipelines

TEST 4 - QUESTION: 35/50

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 create a model to forecast weather conditions based on historical data. You need to create a pipeline that runs a processing script to load data from a datastore and pass the processed data to a machine learning model training script. Solution: Run the following code:

```
datastore = ws.get_default_datastore()
data_output = PipelineData("processed_data", datastore=datastore)
process_step = PythonScriptStep(script_name="process.py",
 arguments=["--data_for_train", data_output],
 outputs=[data_output], compute_target=aml_compute,
 source_directory=process_directory)
pipeline = Pipeline(workspace=ws, steps=[process_step])
```


Does the solution meet the goal?

A

No

B

Yes

CORRECT ANSWER: A

KEEP OPEN

EXPLANATION:

Explanation: train_step is missing. Reference: https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.pipelinedata?view=azure-ml-py

TEST 4 - QUESTION: 36/50 *SELECT MULTIPLE*

You are training machine learning models in Azure Machine Learning. You use Hyperdrive to tune the hyperparameters. In previous model training and tuning runs, many models showed similar performance. You need to select an early termination policy that meets the following requirements: accounts for the performance of all previous runs when evaluating the current run avoids comparing the current run with only the best performing run to date Which two early termination policies should you use? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

☐ A

Default

☐ B

Median stopping

☐ C

Bandit

☐ D

Truncation selection

CORRECT ANSWERS: A,B

KEEP OPEN

EXPLANATION:

Explanation: The Median Stopping policy computes running averages across all runs and cancels runs whose best performance is worse than the median of the running averages. If no policy is specified, the hyperparameter tuning service will let all training runs execute to completion. Incorrect Answers: B: BanditPolicy defines an early termination policy based on slack criteria, and a frequency and delay interval for evaluation. The Bandit policy takes the following configuration parameters: slack_factor: The amount of slack allowed with respect to the best performing training run. This factor specifies the slack as a ratio. D: The Truncation selection policy periodically cancels the given percentage of runs that rank the lowest for their performance on the primary metric. The policy strives for fairness in ranking the runs by accounting for improving model performance with training time. When ranking a relatively young run, the policy uses the corresponding (and earlier) performance of older runs for comparison. Therefore, runs aren't terminated for having a lower performance because they have run for less time than other runs. Reference: https://docs.microsoft.com/en-us/python/api/azureml-train-core/azureml.train.hyperdrive.medianstoppingpolicy

TEST 4 - QUESTION: 37/50

You need to implement a scaling strategy for the local penalty detection data. Which normalization type should you use?

SCENARIO

Case study Overview You are a data scientist in a company that provides data science for professional sporting events. Models will use global and local market data to meet the following business goals: Understand sentiment of mobile device users at sporting events based on audio from crowd reactions. Assess a user's tendency to respond to an advertisement. Customize styles of ads served on mobile devices. Use video to detect penalty events Current environment Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared using social media. The images and videos will have varying sizes and formats. The data available for model building comprises of seven years of sporting event media. The sporting event media includes; recorded video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events. Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats. Penalty detection and sentiment Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection. Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines. Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation. Notebooks must execute with the same code on new Spark instances to recode only the source of the data. Global penalty detection models must be trained by using dynamic runtime graph computation during training. Local penalty detection models must be written by using BrainScript. Experiments for local crowd sentiment models must combine local penalty detection data. Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds. All shared features for local models are continuous variables. Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics available. Advertisements During the initial weeks in production, the following was observed: Ad response rated declined. Drops were not consistent across ad styles. The distribution of features across training and production data are not consistent Analysis shows that, of the 100 numeric features on user location and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue

is to engineer 10 linearly uncorrelated features. Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models. All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow. Audio samples show that the length of a catch phrase varies between 25%-47% depending on region The performance of the global penalty detection models shows lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases. Ad response models must be trained at the beginning of each event and applied during the sporting event. Market segmentation models must optimize for similar ad response history. Sampling must guarantee mutual and collective exclusively between local and global segmentation models that share the same features. Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement. Ad response models must support non-linear boundaries of features. The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%. The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	2
	1	2	1

The ad propensity model uses proposed cost factors shown in the following diagram:

		Actual	
		1	**0**
Predicted	**0**	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown

in the following diagram:

A
Cosine

B
Streaming

C
Weight

D
Batch

CORRECT ANSWER: D
KEEP OPEN

EXPLANATION:

Explanation: Post batch normalization statistics (PBN) is the Microsoft Cognitive Toolkit (CNTK) version of how to evaluate the population mean and variance of Batch Normalization which could be used in inference Original Paper. In CNTK, custom networks are defined using the BrainScriptNetworkBuilder and described in the CNTK network description language "BrainScript." Scenario: Local penalty detection models must be written by using BrainScript. Reference: https://docs.microsoft.com/en-us/cognitive-toolkit/post-batch-normalization-statistics

TEST 4 - QUESTION: 38/50

HOTSPOT You need to identify the methods for dividing the data according to the testing requirements. Which properties should you select? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

SCENARIO

Case study This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided. To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study. At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section. To start the case study To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question. Overview You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules. Datasets There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

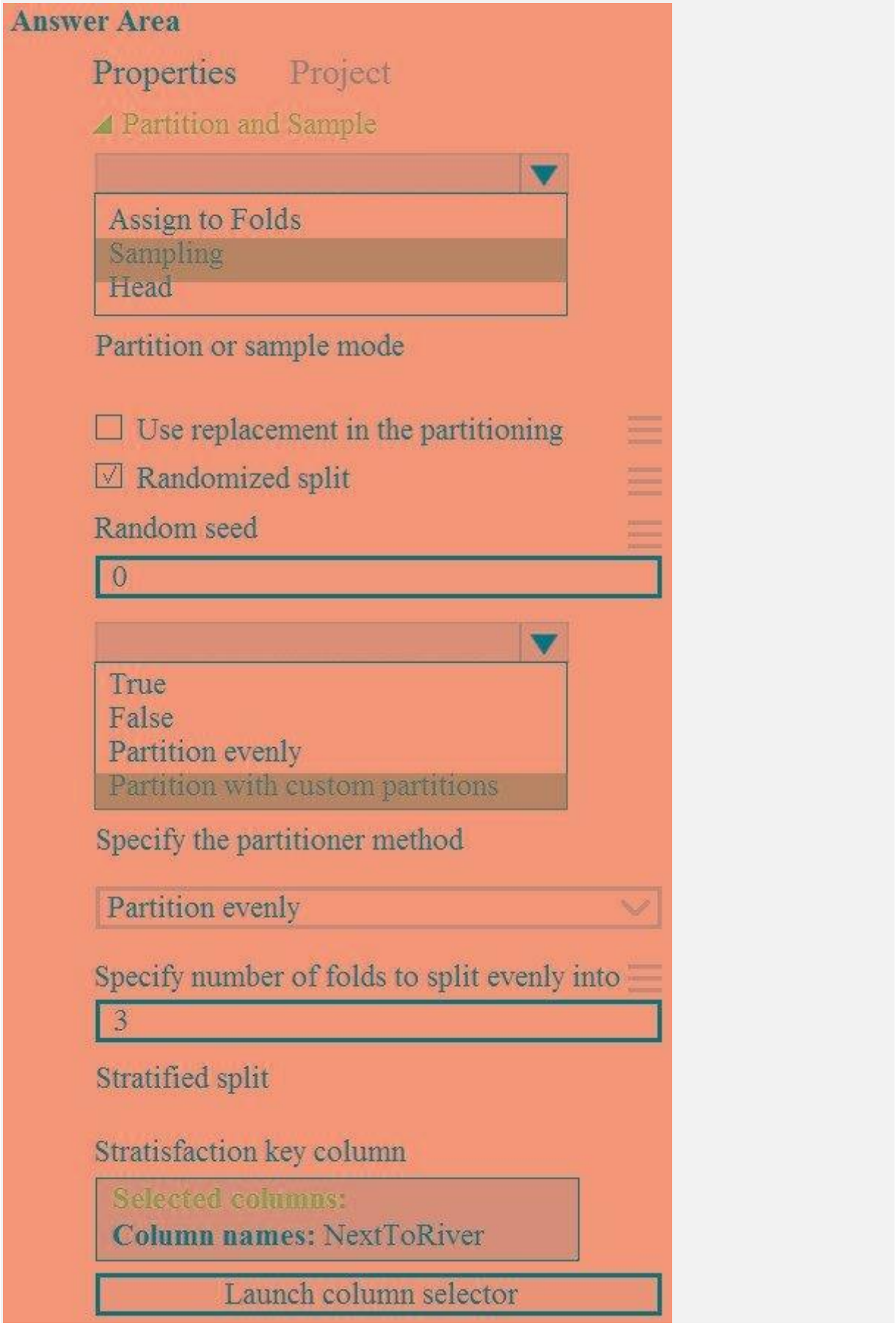
Column heading	Description
CapitaCrimeRate	per capita crime rate by town
Zoned	proportion of residential land zoned for lots over 25.000 square feet
NonRetailAcres	proportion of retail business acres per town
NextToRiver	proximity of a property to the river
NitrogenOxideConcentration	nitric oxides concentration (parts per 10 million)
AvgRoomsPerHouse	average number of rooms per dwelling
Age	proportion of owner-occupied units built prior to 1940
DistanceToEmploymentCenter	weighted distances to employment centers
AccessibilityToHighway	index of accessibility to radial highways to a value of two decimal places
Tax	full value property tax rate per $10,000
PupilTeacherRatio	pupil to teacher ratio by town
ProfessionalClass	professional class percentage
LowerStatus	percentage lower status of the population
MedianValue	median value of owner-occupied homes in $1000s

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format. Data issues Missing values The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values. Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail. Model fit The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting. Experiment requirements You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset. You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships. You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns. Model training Permutation Feature Importance Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model. Hyperparameters You must configure

hyperparameters in the model learning process to speed the learning phase . In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful. You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs. Testing You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio. Cross-validation You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city's main river. You must complete this task before the data goes through the sampling process. Linear regression module When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent. Data visualization You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results. You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another.

CHECK BELOW THE RIGHT ANSWER

Answer Area


Properties
Project
Partition and Sample
Assign to Folds
Sampling
Head
Partition or sample mode
Use replacement in the partitioning
Randomized split
Random seed
0
True
False
Partition evenly
Partition with custom partitions
Specify the partitioner method
Partition evenly
Specify number of folds to split evenly into
3
Stratified split
Stratisfaction key column
Selected columns:
Column names: NextToRiver
Launch column selector

CORRECT ANSWER:
KEEP OPEN

EXPLANATION:

Explanation: Scenario: Testing You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio. Box 1: Assign to folds Use Assign to folds option when you want to divide the dataset into subsets of the data. This option is also useful when you want to create a custom number of folds for cross-validation, or to split rows into several groups. Not Head: Use Head mode to get only the first n rows. This option is useful if you want to test a pipeline on a small number of rows, and don't need the data to be balanced or sampled in any way. Not Sampling: The Sampling option supports simple random sampling or stratified random sampling. This is useful if you want to create a smaller representative sample dataset for testing. Box 2: Partition evenly Specify the partitioner method: Indicate how you want data to be apportioned to each partition, using these options: Partition evenly: Use this option to place an equal number of rows in each partition. To specify the number of output partitions, type a whole number in the Specify number of folds to split evenly into text box. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/partition-and-sample

TEST 4 - QUESTION: 39/50

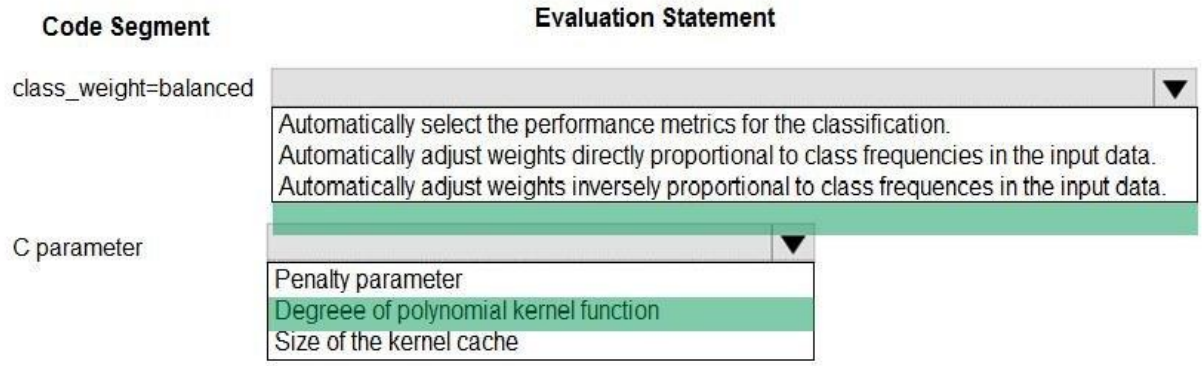
HOTSPOT You are using C-Support Vector classification to do a multi-class classification with an unbalanced training dataset. The C-Support Vector classification using Python code shown below:

```
from sklearn.svm import svc
import numpy as np
svc = SVC(kernel= 'linear', class_weight= 'balanced', C-1.0, random_state-0)
model1 = svc.fit(X_train, y)
```

You need to evaluate the C-Support Vector classification code. Which evaluation statement should you use? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

CHECK BELOW THE RIGHT ANSWER

Answer Area



CORRECT ANSWER:

KEEP OPEN

EXPLANATION:

Explanation: Box 1: Automatically adjust weights inversely proportional to class frequencies in the input data The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y)). Box 2: Penalty parameter Parameter: C : float, optional (default=1.0) Penalty parameter C of the error term. Reference: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

TEST 4 - QUESTION: 40/50 *SELECT MULTIPLE*

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.

CORRECT ANSWERS: A, C, D

KEEP OPEN

EXPLANATION:

Explanation: AD: The algorithm terminates when the centroids stabilize or when a specified number of iterations are completed. C: A measure of how well the centroids represent the members of their clusters is the residual sum of squares or RSS, the squared distance of each vector from its centroid summed over all vectors. RSS is the objective function and our goal is to minimize it. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/k-means-clustering https://nlp.stanford.edu/IR-book/html/htmledition/k-means-1.html

TEST 4 - QUESTION: 41/50 *SELECT MULTIPLE*

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

Save each function to a separate Python file

☐ B

Define a main() function in the Python script

☐ C

Refactor the Jupyter Notebook code into functions

☐ D

Remove all comments and functions from the Python script

CORRECT ANSWERS: B,C

KEEP OPEN

EXPLANATION:

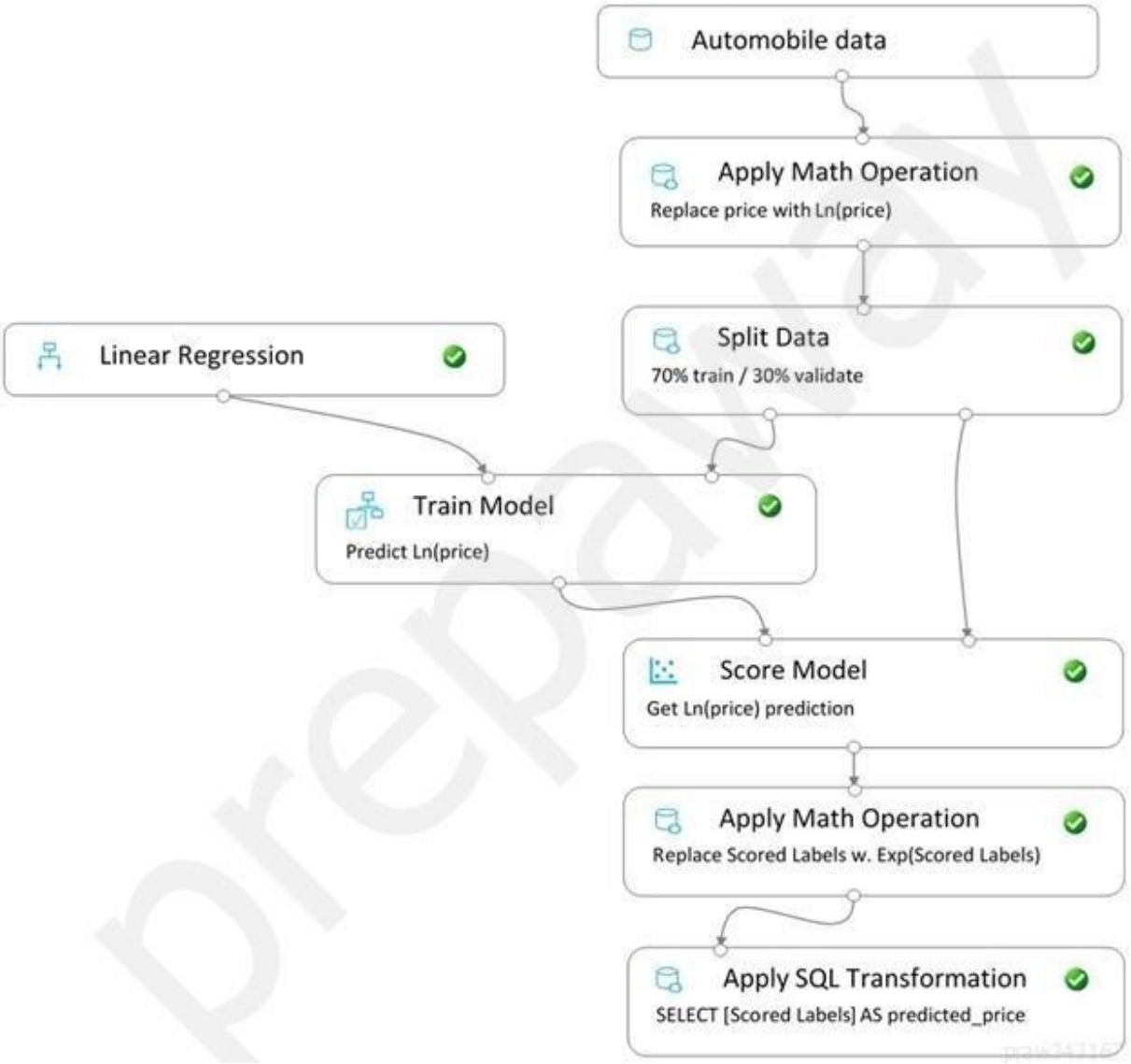
Explanation: C: Python main function is a starting point of any program. When the program is run, the python interpreter runs the code sequentially. Main function is executed only when it is run as a Python program. A: Refactoring, code style and testing The first step is to modularise the notebook into a reasonable folder structure, this effectively means to convert files from .ipynb format to .py format, ensure each script has a clear distinct purpose and organise these files in a coherent way.

```
├── src
│   ├── conf           # stores project configurations is json format.
│   ├── main           # main logic for training, predicting and visualisation.
│   ├── resources      # storage of resources such as trained models.
│   ├── template_app   # contains all logic for the flask application.
│   └── utils          # helper functions.
├── tests              # contains projects testing suite.
├── docker-compose.yml # Docker configurations.
├── Dockerfile         # machine instructions to setup the application and run inside D
├── logs.log           # log files storage.
├── Readme.md
├── requirements.txt   # Python dependancies for installation with pip.
├── run_app.py         # entry point of the project for the Flask application.
└── run.py             # entry point of the project for local usage.
```

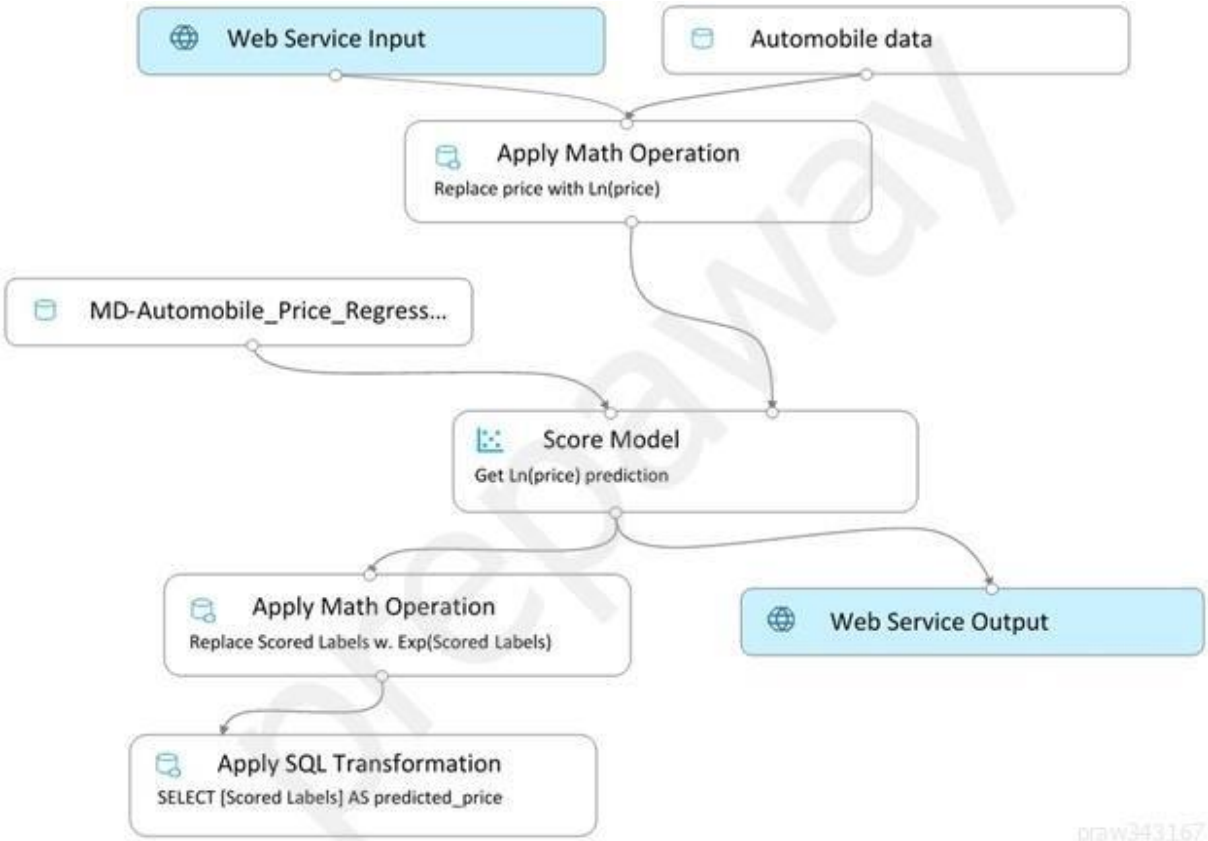
Once the project is nicely structured we can tidy up or refactor the code.

TEST 4 - QUESTION: 42/50 *SELECT MULTIPLE*

You create a pipeline in designer to train a model that predicts automobile prices. Because of non-linear relationships in the data, the pipeline calculates the natural log (Ln) of the prices in the training data, trains a model to predict this natural log of price value, and then calculates the exponential of the scored label to get the predicted price. The training pipeline is shown in the exhibit. (Click the Training pipeline tab.) Training pipeline



You create a real-time inference pipeline from the training pipeline, as shown in the exhibit. (Click the Real-time pipeline tab.) Real-time pipeline



You need to modify the inference pipeline to ensure that the web service returns the exponential of the scored label as the predicted automobile price and that client applications are not required to include a price value in the input values. Which three modifications must you make to the inference pipeline? Each correct answer presents part of the solution. NOTE : Each correct selection is worth one point.

☐ A

Replace the training dataset module with a data input that does not include the price column.

☐ B

Remove the Apply Math Operation module that replaces price with its natural log from the data flow.

☐ C

Remove the Apply SQL Transformation module from the data flow.

☐ D

Add a Select Columns module before the Score Model module to select all columns other than price.

☐ E

Connect the output of the Apply SQL Transformation to the Web Service Output module.

☐ F

Replace the Web Service Input module with a data input that does not include the price column.

CORRECT ANSWERS: B,D,E
KEEP OPEN

EXPLANATION:

As you can see at the moment our simulator just shows to you the right answer. We are trying our best to add also a reasonable explanation for the provided answer. We are sorry for the inconvenience.

TEST 4 - QUESTION: 43/50 *SELECT MULTIPLE*

You create a binary classification model. You need to evaluate the model performance. Which two metrics can you use? Each correct answer presents a complete solution. NOTE: Each correct selection is worth one point.

☐ A
relative absolute error

☐ B
mean absolute error

☐ C
coefficient of determination

☐ D
precision

☐ E
accuracy

CORRECT ANSWERS: D,E

KEEP OPEN

EXPLANATION:

Explanation: The evaluation metrics available for binary classification models are: Accuracy, Precision, Recall, F1 Score, and AUC. Note: A very natural question is: 'Out of the individuals whom the model, how many were classified correctly (TP)?' This question can be answered by looking at the Precision of the model, which is the proportion of positives that are classified correctly. Reference: https://docs.microsoft.com/en-us/azure/machine-learning/studio/evaluate-model-performance

TEST 4 - QUESTION: 44/50 *SELECT MULTIPLE*

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 URL of the endpoint.

☐ B

The key for the endpoint.

☐ C

The name of the inference pipeline for the endpoint.

☐ D

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

☐ E

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

CORRECT ANSWERS: A,B

KEEP OPEN

EXPLANATION:

Explanation: Deploying an Azure Machine Learning model as a web service creates a REST API endpoint. You can send data to this endpoint and receive the prediction returned by the model. You create a web service when you deploy a model to your local environment, Azure Container Instances, Azure Kubernetes Service, or field-programmable gate arrays (FPGA). You retrieve the URI used to access the web service by using the Azure Machine Learning SDK. If authentication is enabled, you can also use the SDK to get the authentication keys or tokens. Example: # URL for the web service scoring_uri = '<your web service URI>' # If the service is authenticated, set the key or token key = '<your key or token>' Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-consume-web-service

TEST 4 - QUESTION: 45/50

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.log_list('Label Values', label_vals) Does the solution meet the goal?

A

No

B

Yes

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: run.log_list log a list of values to the run with the given name using log_list. Example: run.log_list("accuracies", [0.6, 0.7, 0.87]) Note: Data= pd.read_csv('data.csv') Data is read into a pandas.DataFrame, which is a two-dimensional, size-mutable, potentially heterogeneous tabular data. label_vals =data['label'].unique label_vals contains a list of unique label values. Reference:

TEST 4 - QUESTION: 46/50

You create and register a model in an Azure Machine Learning workspace. You must use the Azure Machine Learning SDK to implement a batch inference pipeline that uses a ParallelRunStep to score input data using the model. You must specify a value for the ParallelRunConfig compute_target setting of the pipeline step. You need to create the compute target. Which class should you use?

A
BatchCompute

B
AdlaCompute

C
AksCompute

D
AmlCompute

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Explanation: Compute target to use for ParallelRunStep. This parameter may be specified as a compute target object or the string name of a compute target in the workspace. The compute_target target is of AmlCompute or string. Note: An Azure Machine Learning Compute (AmlCompute) is a managed-compute infrastructure that allows you to easily create a single or multi-node compute. The compute is created within your workspace region as a resource that can be shared with other users Reference: https://docs.microsoft.com/en-us/python/api/azureml-contrib-pipeline-steps/azureml.contrib.pipeline.steps.parallelrunconfig https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.compute.amlcompute(class)

TEST 4 - QUESTION: 47/50

You plan to run a Python script as an Azure Machine Learning experiment. The script must read files from a hierarchy of folders. The files will be passed to the script as a dataset argument. You must specify an appropriate mode for the dataset argument. Which two modes can you use? Each correct answer presents a complete solution. NOTE: Each correct selection is worth one point.

- A as_mount()
- B as_upload()
- C to_pandas_dataframe()
- D as_download()

CORRECT ANSWER: D

KEEP OPEN

EXPLANATION:

Reference: https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.data.filedataset?view=azure-ml-py

TEST 4 - QUESTION: 48/50 *SELECT MULTIPLE*

You use the Azure Machine Learning SDK in a notebook to run an experiment using a script file in an experiment folder. The experiment fails. You need to troubleshoot the failed experiment. What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

☐ A

Use the get_output() method of the run object to retrieve the experiment run logs.

☐ B

Use the get_metrics() method of the run object to retrieve the experiment run logs.

☐ C

View the logs for the experiment run in Azure Machine Learning studio.

☐ D

Use the get_details_with_logs() method of the run object to display the experiment run logs.

☐ E

View the log files for the experiment run in the experiment folder.

CORRECT ANSWERS: C,D

KEEP OPEN

EXPLANATION:

Explanation: Use get_details_with_logs() to fetch the run details and logs created by the run. You can monitor Azure Machine Learning runs and view their logs with the Azure Machine Learning studio. Incorrect Answers: A: You can view the metrics of a trained model using run.get_metrics(). E: get_output() gets the output of the step as PipelineData. Reference: https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.steprun https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-view-training-logs

TEST 4 - QUESTION: 49/50

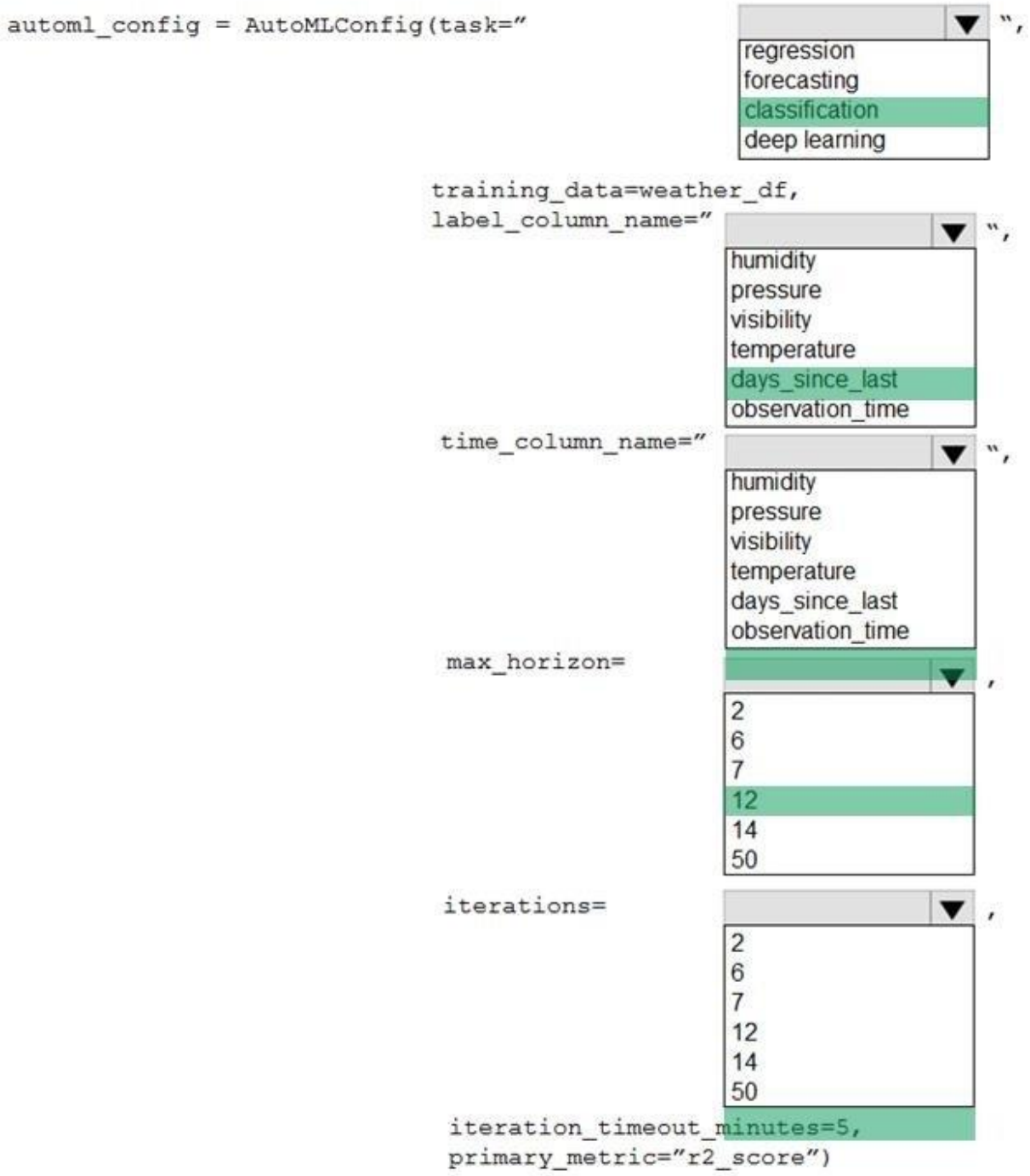
HOTSPOT You collect data from a nearby weather station. You have a pandas dataframe named weather_df that includes the following data:

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.

CHECK BELOW THE RIGHT ANSWER

Answer Area



CORRECT ANSWER:
KEEP OPEN

EXPLANATION:

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

TEST 4 - QUESTION: 50/50

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 have a Python script named train.py in a local folder named scripts. The script trains a regression model by using scikit-learn. The script includes code to load a training data file which is also located in the scripts folder. You must run the script as an Azure ML experiment on a compute cluster named aml-compute. You need to configure the run to ensure that the environment includes the required packages for model training. You have instantiated a variable named aml-compute that references the target compute cluster. Solution: Run the following code:

```
from azureml.train.estimator import Estimator
sk_est = Estimator(source_directory='./scripts',
 compute_target=aml-compute,
 entry_script='train.py')
```


Does the solution meet the goal?

☐ A

Yes

☐ B

No

CORRECT ANSWER: B

KEEP OPEN

EXPLANATION:

Explanation: There is a missing line: conda_packages=['scikit-learn'], which is needed. Correct example: sk_est = Estimator(source_directory='./my-sklearn-proj', script_params=script_params, compute_target=compute_target, entry_script='train.py', conda_packages=['scikit-learn']) Note: The Estimator class represents a generic estimator to train data using any supplied framework. This class is designed for use with machine learning frameworks that do not already have an Azure Machine Learning pre-configured estimator. Pre-configured estimators exist for Chainer, PyTorch, TensorFlow, and SKLearn. Example: from azureml.train.estimator import Estimator script_params = { # to mount files referenced by mnist dataset '--data-folder': ds.as_named_input('mnist').as_mount(), '--regularization': 0.8 } Reference: