DP-100 Q&As

Designing and Implementing a Data Science Solution on Azure

Pass Microsoft DP-100 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/dp-100.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

You create an experiment in Azure Machine Learning Studio. You add a training dataset that contains 10,000 rows. The first 9,000 rows represent class 0 (90 percent).

The remaining 1,000 rows represent class 1 (10 percent).

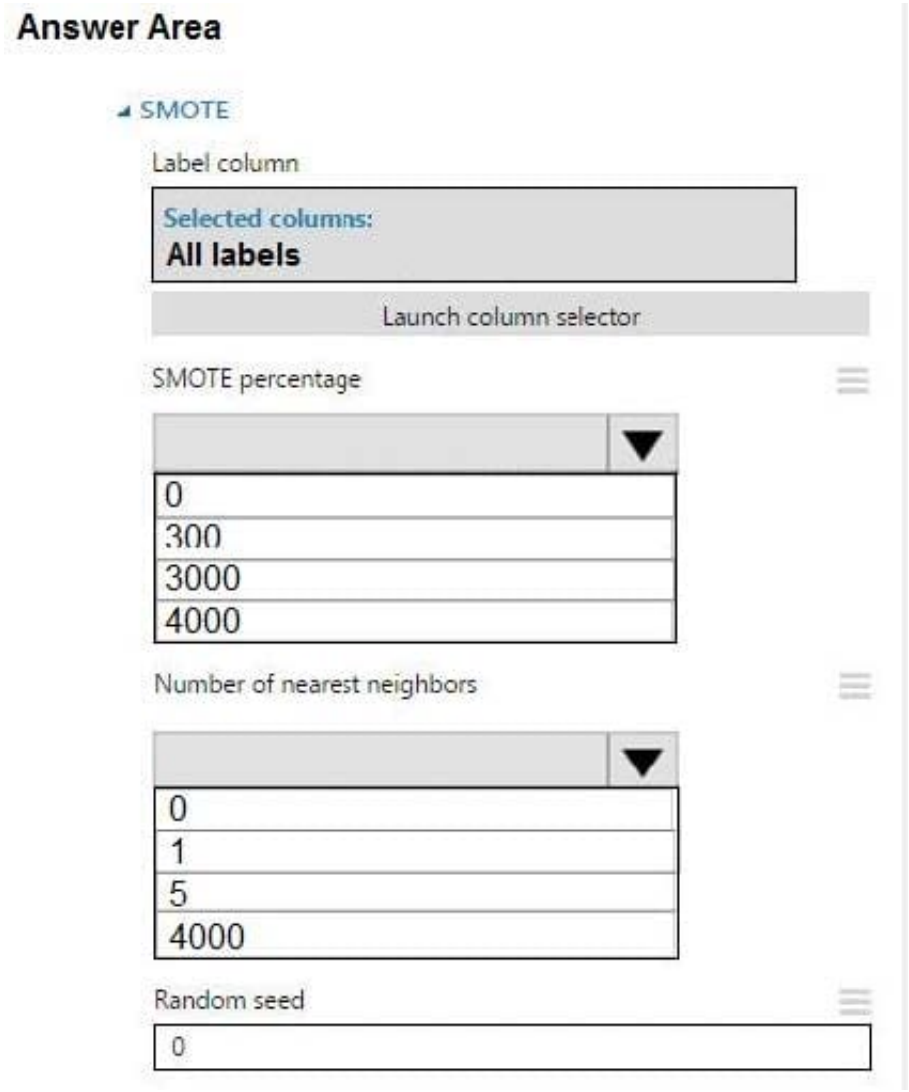
The training set is imbalances between two classes. You must increase the number of training examples for class 1 to 4,000 by using 5 data rows. You add the Synthetic Minority Oversampling Technique (SMOTE) module to the experiment.

You need to configure the module.

Which values should you use? To answer, select the appropriate options in the dialog box in the answer area.

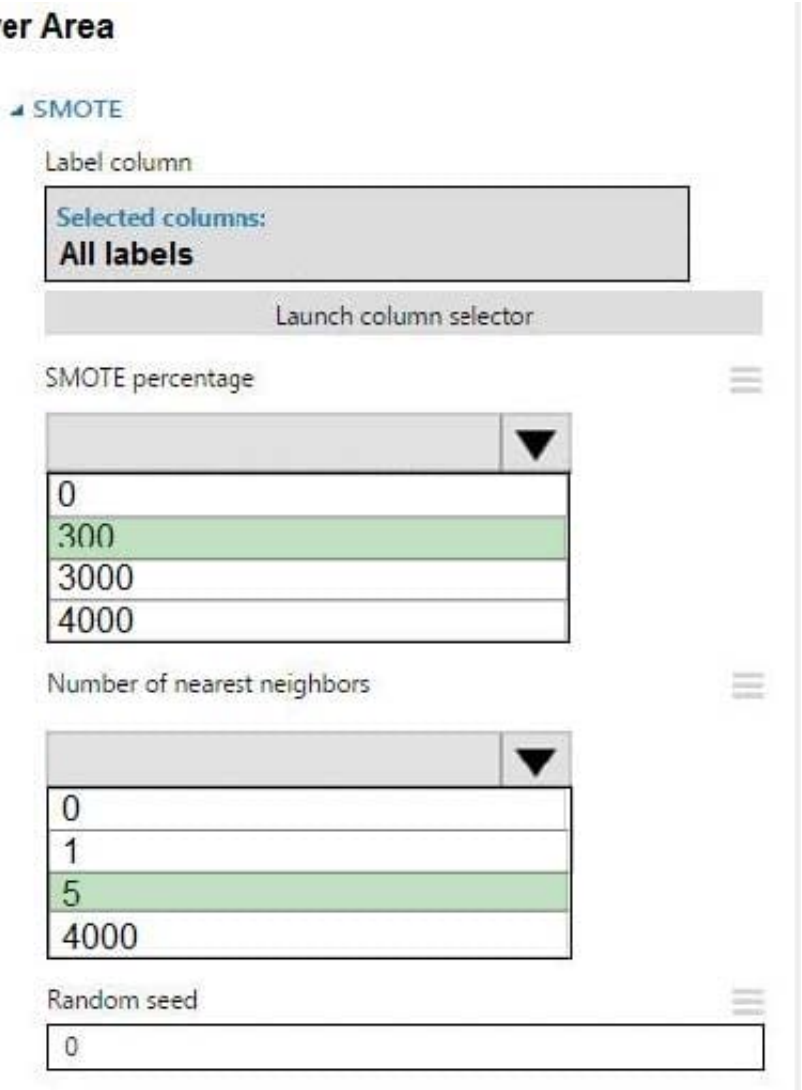
NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:

Answer Area



Box 1: 300

You type 300 (%), the module triples the percentage of minority cases (3000) compared to the original dataset (1000).

Box 2: 5

We should use 5 data rows.

Use the Number of nearest neighbors option to determine the size of the feature space that the SMOTE algorithm uses when in building new cases. A nearest neighbor is a row of data (a case) that is very similar to some target case. The

distance between any two cases is measured by combining the weighted vectors of all features.

By increasing the number of nearest neighbors, you get features from more cases.

By keeping the number of nearest neighbors low, you use features that are more like those in the original sample.

References:

https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/smote

QUESTION 2

You are performing feature engineering on a dataset.

You must add a feature named CityName and populate the column value with the text London.

You need to add the new feature to the dataset.

Which Azure Machine Learning Studio module should you use?

A. Extract N-Gram Features from Text

B. Edit Metadata

C. Preprocess Text

D. Apply SQL Transformation

Correct Answer: B

Typical metadata changes might include marking columns as features.

Reference:

https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/edit-metadata

QUESTION 3

You use an Azure Machine Learning workspace.

You have a trained model that must be deployed as a web service. Users must authenticate by using Azure Active Directory.

What should you do?

A. Deploy the model to Azure Kubernetes Service (AKS). During deployment, set the token_auth_enabledparameter of the target configuration object to true

B. Deploy the model to Azure Container Instances. During deployment, set the auth_enabledparameter of the target configuration object to true

C. Deploy the model to Azure Container Instances. During deployment, set the token_auth_enabledparameter of the target configuration object to true

D. Deploy the model to Azure Kubernetes Service (AKS). During deployment, set the auth.enabledparameter of the target configuration object to true

Correct Answer: A

To control token authentication, use the token_auth_enabled parameter when you create or update a deployment

Token authentication is disabled by default when you deploy to Azure Kubernetes Service.

Note: The model deployments created by Azure Machine Learning can be configured to use one of two authentication methods:

key-based: A static key is used to authenticate to the web service.

token-based: A temporary token must be obtained from the Azure Machine Learning workspace (using Azure Active Directory) and used to authenticate to the web service.

Incorrect Answers:

C: Token authentication isn\\'t supported when you deploy to Azure Container Instances.

Reference: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-authenticate-web-service

QUESTION 4

DRAG DROP

You plan to explore demographic data for home ownership in various cities. The data is in a CSV file with the following format:

age,city,income,home_owner21,Chicago,50000,035,Seattle,120000,123,Seattle,65000,045,Seattle,130000,118,Chicago,48000,0

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

1.

the number of observations in the dataset

2.

a box plot of income by home_owner

3.

a dictionary containing the city names and the average income for each city You need to use the appropriate logging methods of the experiment\\'s run object to log the required information. How should you complete the code? To answer, drag the appropriate code segments to the correct locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to

view content.

NOTE: Each correct selection is worth one point.

Select and Place:

Code segments

log
log_list
log_row
log_table
log_image

Answer Area

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

Correct Answer:

Code segments

log_list
log_row

Answer Area

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

Box 1: log

The number of observations in the dataset.

run.log(name, value, description=\\'\\')

Scalar values: Log a numerical or string value to the run with the given name. Logging a metric to a run causes that metric to be stored in the run record in the experiment. You can log the same metric multiple times within a run, the result

being considered a vector of that metric.

Example: run.log("accuracy", 0.95)

Box 2: log_image

A box plot of income by home_owner.

log_image Log an image to the run record. Use log_image to log a .PNG image file or a matplotlib plot to the run. These images will be visible and comparable in the run record.

Example: run.log_image("ROC", plot=plt)

Box 3: log_table

A dictionary containing the city names and the average income for each city.

log_table: Log a dictionary object to the run with the given name.

QUESTION 5

DRAG DROP

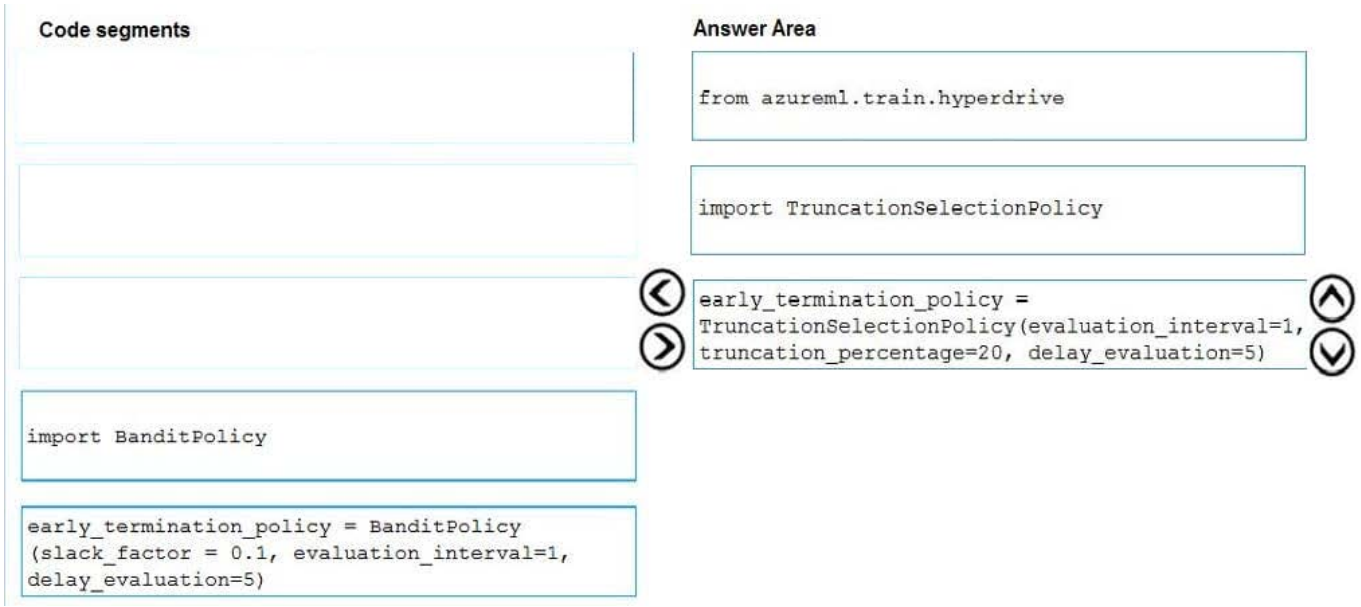
You need to implement an early stopping criteria policy for model training.

Which three code segments should you use to develop the solution? To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Select and Place:

Correct Answer:



You need to implement an early stopping criterion on models that provides savings without terminating promising jobs.

Truncation selection cancels a given percentage of lowest performing runs at each evaluation interval. Runs are compared based on their performance on the primary metric and the lowest X% are terminated.

Example:

from azureml.train.hyperdrive import TruncationSelectionPolicy

early_termination_policy = TruncationSelectionPolicy(evaluation_interval=1, truncation_percentage=20, delay_evaluation=5)

Incorrect Answers:

Bandit is a termination policy based on slack factor/slack amount and evaluation interval. The policy early terminates

any runs where the primary metric is not within the specified slack factor / slack amount with respect to the best performing

training run.

Example:

from azureml.train.hyperdrive import BanditPolicy

early_termination_policy = BanditPolicy(slack_factor = 0.1, evaluation_interval=1, delay_evaluation=5

References:

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