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

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

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

NOTE: Each correct selection is worth one point.

A. The label data must be a negative value.

B. The label data must be whole numbers.

C. The label data must be non-discrete.

D. The label data must be a positive value.

E. The label data can be positive or negative.

Correct Answer: BD

Poisson regression is intended for use in regression models that are used to predict numeric values, typically counts. Therefore, you should use this module to create your regression model only if the values you are trying to predict fit the following conditions:

1.

The response variable has a Poisson distribution.

2.

Counts cannot be negative. The method will fail outright if you attempt to use it with negative labels.

3.

A Poisson distribution is a discrete distribution; therefore, it is not meaningful to use this method with non-whole numbers.

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

QUESTION 2

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while

others might not have a correct solution.

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

You are creating a model to predict the price of a student\\'s artwork depending on the following variables:

the student\\'s length of education, degree type, and art form.

You start by creating a linear regression model.

You need to evaluate the linear regression model.

Solution: Use the following metrics: Mean Absolute Error, Root Mean Absolute Error, Relative Absolute Error, Accuracy, Precision, Recall, F1 score, and AUC.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

Accuracy, Precision, Recall, F1 score, and AUC are metrics for evaluating classification models.

Note: Mean Absolute Error, Root Mean Absolute Error, Relative Absolute Error are OK for the linear regression model.

Reference:

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

QUESTION 3

You are building a binary classification model by using a supplied training set.

The training set is imbalanced between two classes.

You need to resolve the data imbalance.

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

NOTE: Each correct selection is worth one point.

A. Penalize the classification

B. Resample the dataset using undersampling or oversampling

C. Normalize the training feature set

D. Generate synthetic samples in the minority class

E. Use accuracy as the evaluation metric of the model

Correct Answer: ABD

A: Try Penalized Models

You can use the same algorithms but give them a different perspective on the problem. Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training. These penalties

can bias the model to pay more attention to the minority class.

B: You can change the dataset that you use to build your predictive model to have more balanced data. This change is called sampling your dataset and there are two main methods that you can use to even-up the classes:

1.

Consider testing under-sampling when you have an a lot data (tens- or hundreds of thousands of instances or more)

2.

Consider testing over-sampling when you don\\'t have a lot of data (tens of thousands of records or less)

D: Try Generate Synthetic Samples

A simple way to generate synthetic samples is to randomly sample the attributes from instances in the minority class.

Reference:

https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning- dataset/

QUESTION 4

DRAG DROP

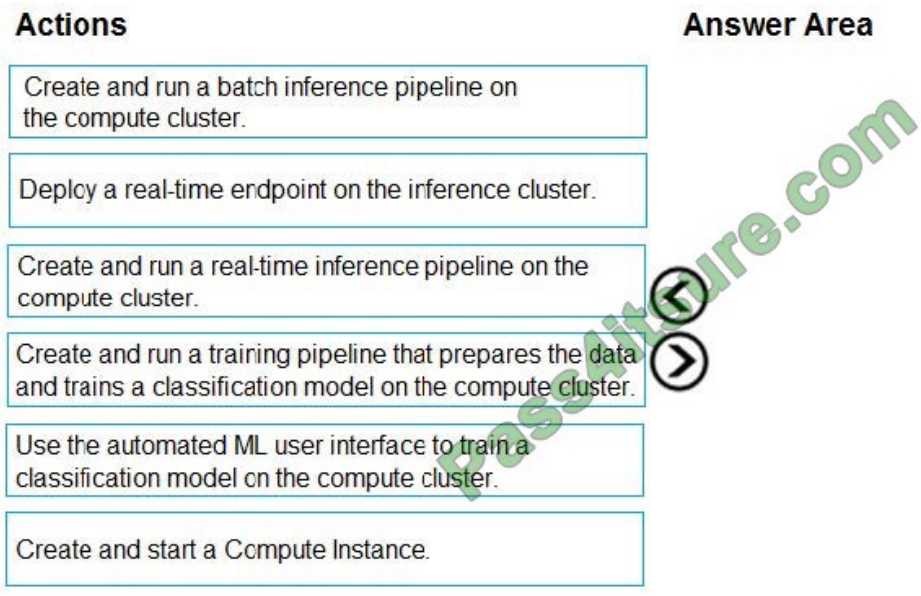
You have an Azure Machine Learning workspace that contains a CPU-based compute cluster and an Azure Kubernetes Services (AKS) inference cluster. You create a tabular dataset containing data that you plan to use to create a

classification model.

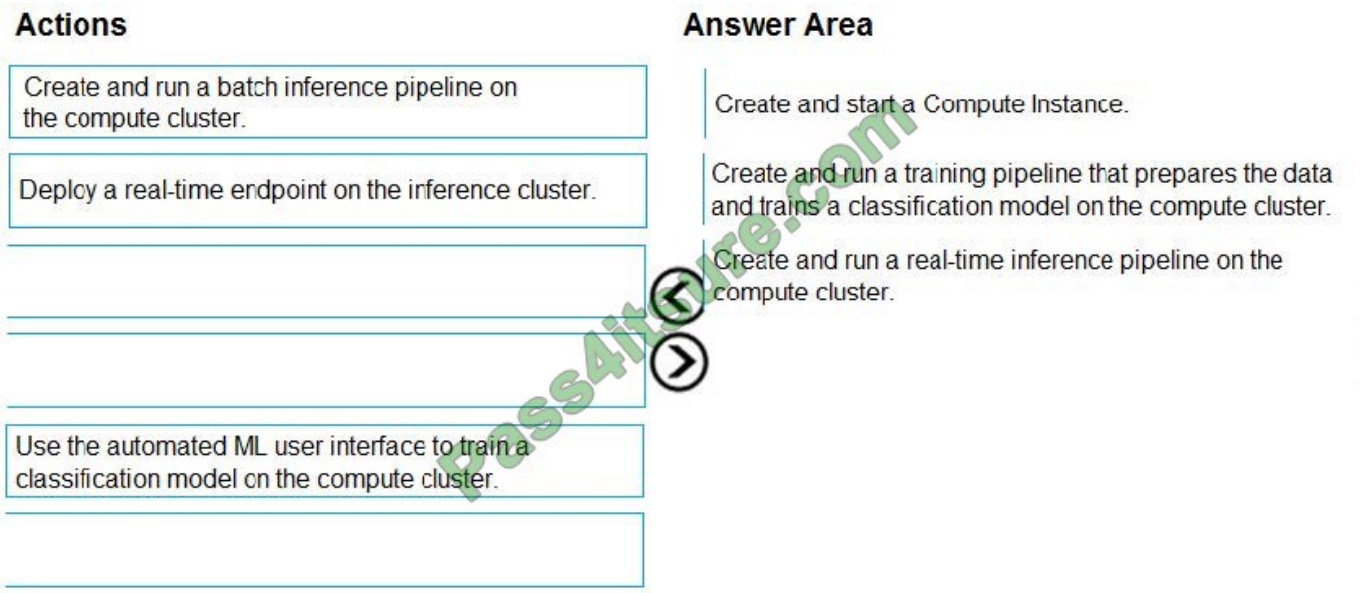
You need to use the Azure Machine Learning designer to create a web service through which client applications can consume the classification model by submitting new data and getting an immediate prediction as a response.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:



Correct Answer:



Step 1: Create and start a Compute Instance

To train and deploy models using Azure Machine Learning designer, you need compute on which to run the training process, test the model, and host the model in a deployed service.

There are four kinds of compute resource you can create:

Compute Instances: Development workstations that data scientists can use to work with data and models.

Compute Clusters: Scalable clusters of virtual machines for on-demand processing of experiment code.

Inference Clusters: Deployment targets for predictive services that use your trained models.

Attached Compute: Links to existing Azure compute resources, such as Virtual Machines or Azure Databricks clusters.

Step 2: Create and run a training pipeline..

After you\\'ve used data transformations to prepare the data, you can use it to train a machine learning model. Create and run a training pipeline

Step 3: Create and run a real-time inference pipeline

After creating and running a pipeline to train the model, you need a second pipeline that performs the same data transformations for new data, and then uses the trained model to inference (in other words, predict) label values based on its

features. This pipeline will form the basis for a predictive service that you can publish for applications to use.

Reference:

https://docs.microsoft.com/en-us/learn/modules/create-classification-model-azure-machine-learning-designer/

QUESTION 5

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.

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.pass4itsure.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: