

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.leads4pass.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 have been tasked with evaluating your model on a partial data sample via k-fold cross-validation.

You have already configured a k parameter as the number of splits. You now have to configure the k parameter for the cross-validation with the usual value choice.

Recommendation: You configure the use of the value k=1.

Will the requirements be satisfied?

A. Yes

B. No

Correct Answer: B

QUESTION 2**HOTSPOT**

You have a multi-class image classification deep learning model that uses a set of labeled photographs. You create the following code to select hyperparameter values when training the model.

```
from azureml.train.hyperdrive import BayesianParameterSampling
param_sampling = BayesianParametersSampling ({
    "learning_rate": uniform(0.01, 0.1),
    "batch_size": choice(16, 32, 64, 128)}
)
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

	Yes	No
Hyperparameter combinations for the runs are selected based on how previous samples performed in the previous experiment run.	<input type="checkbox"/>	<input type="checkbox"/>
The learning rate value 0.09 might be used during model training.	<input type="checkbox"/>	<input type="checkbox"/>
You can define an early termination policy for this hyperparameter tuning run.	<input type="checkbox"/>	<input type="checkbox"/>

Correct Answer:

	Yes	No
Hyperparameter combinations for the runs are selected based on how previous samples performed in the previous experiment run.	<input checked="" type="radio"/>	<input type="radio"/>
The learning rate value 0.09 might be used during model training.	<input checked="" type="radio"/>	<input type="radio"/>
You can define an early termination policy for this hyperparameter tuning run.	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes

Hyperparameters are adjustable parameters you choose to train a model that govern the training process itself. Azure Machine Learning allows you to automate hyperparameter exploration in an efficient manner, saving you significant time

and resources. You specify the range of hyperparameter values and a maximum number of training runs. The system then automatically launches multiple simultaneous runs with different parameter configurations and finds the configuration

that results in the best performance, measured by the metric you choose. Poorly performing training runs are automatically early terminated, reducing wastage of compute resources. These resources are instead used to explore other

hyperparameter configurations.

Box 2: Yes

uniform(low, high) - Returns a value uniformly distributed between low and high

Box 3: No

Bayesian sampling does not currently support any early termination policy.

Reference:

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

QUESTION 3

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

others might not have a correct solution.

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

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

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

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

You plan to use this configuration to run a script that trains a random forest model and then tests it with validation data. The label values for the validation data are stored in a variable named `y_test` variable, and the predicted probabilities from the model are stored in a variable named `y_predicted`.

Solution: Run the following code:

```
import numpy as np  
from sklearn.metrics import roc_auc_score  
from azureml.core.run import Run  
run = Run.get_context()  
# code to train model omitted  
auc = roc_auc_score(y_test, y_predicted)  
run.log("AUC", np.float(auc))
```

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A

QUESTION 4

You are creating an experiment by using Azure Machine Learning Studio.

You must divide the data into four subsets for evaluation. There is a high degree of missing values in the data. You must prepare the data for analysis.

You need to select appropriate methods for producing the experiment.

Which three modules should you run in sequence? To answer, move the appropriate actions from the list of actions 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:

Actions

- Build Counting Transform
- Missing Values Scrubber
- Feature Hashing
- Clean Missing Data
- Replace Discrete Values
- Import Data
- Latent Dirichlet Transformation
- Partition and Sample

Answer Area



Correct Answer:

Actions

- Build Counting Transform
- Missing Values Scrubber
- Feature Hashing
-
- Replace Discrete Values
-
- Latent Dirichlet Transformation
-

Answer Area

- Import Data
- Clean Missing Data
- Partition and Sample



The Clean Missing Data module in Azure Machine Learning Studio, to remove, replace, or infer missing values.

Incorrect Answers:

Latent Dirichlet Transformation: Latent Dirichlet Allocation module in Azure Machine Learning Studio, to group otherwise unclassified text into a number of categories. Latent Dirichlet Allocation (LDA) is often used in natural language processing (NLP) to find texts that are similar. Another common term is topic modeling. **Build Counting Transform:** Build Counting Transform module in Azure Machine Learning Studio, to analyze training data. From this data, the module builds a count table as well as a set of count-based features that can be used in a predictive model.

Missing Value Scrubber: The Missing Values Scrubber module is deprecated.

Feature hashing: Feature hashing is used for linguistics, and works by converting unique tokens into integers.

Replace discrete values: the Replace Discrete Values module in Azure Machine Learning Studio is used to generate a probability score that can be used to represent a discrete value. This score can be useful for understanding the information

value of the discrete values.

References:

<https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/clean-missing-data>

QUESTION 5

You create an Azure Machine Learning workspace. You are implementing hyperparameter tuning for a model training from a notebook.

You must configure a Bandit termination policy that provides the following outcome:

If the value of the primary metric of AUC is 0.8 at the point of evaluation intervals, any run with the primary metric value below 0.66 will be terminated.

You need to identify which Bandit termination policy configuration to use.

What should you identify?

- A. Set `slack_amount` to 0.2.
- B. Set `slack_factor` to 0.1.
- C. Set `slack_factor` to 0.2.
- D. Set `slack_amount` to 0.1.

Correct Answer: C

Bandit policy

Bandit policy is based on slack factor/slack amount and evaluation interval. Bandit policy ends a job when the primary metric isn't within the specified slack factor/slack amount of the most successful job.

Specify the following configuration parameters:

slack_factor or slack_amount: the slack allowed with respect to the best performing training job. slack_factor specifies the allowable slack as a ratio. slack_amount specifies the allowable slack as an absolute amount, instead of a ratio.

For example, consider a Bandit policy applied at interval 10. Assume that the best performing job at interval 10 reported a primary metric is 0.8 with a goal to maximize the primary metric. If the policy specifies a slack_factor of 0.2, any training

jobs whose best metric at interval 10 is less than 0.66 ($0.8/(1+\text{slack_factor})$) will be terminated.

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

[Latest DP-100 Dumps](#)

[DP-100 Practice Test](#)

[DP-100 Study Guide](#)