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.passapply.com/dp-100.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

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. Refactor the Jupyter Notebook code into functions

B. Save each function to a separate Python file

C. Define a main() function in the Python script

D. Remove all comments and functions from the Python script

Correct Answer: AC

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.

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

Reference: https://www.guru99.com/learn-python-main-function-with-examples-understand-main.html
https://towardsdatascience.com/from-jupyter-notebook-to-deployment-a-straightforward-example-1838c203a437

QUESTION 2

You are in the process of constructing a deep convolutional neural network (CNN). The CNN will be used for image classification.

You notice that the CNN model you constructed displays hints of overfitting.

You want to make sure that overfitting is minimized, and that the model is converged to an optimal fit.

Which of the following is TRUE with regards to achieving your goal?

A. You have to add an additional dense layer with 512 input units, and reduce the amount of training data.

B. You have to add L1/L2 regularization, and reduce the amount of training data.

C. You have to reduce the amount of training data and make use of training data augmentation.

D. You have to add L1/L2 regularization, and make use of training data augmentation.

E. You have to add an additional dense layer with 512 input units, and add L1/L2 regularization.

Correct Answer: B

B: Weight regularization provides an approach to reduce the overfitting of a deep learning neural network model on the training data and improve the performance of the model on new data, such as the holdout test set. Keras provides a weight regularization API that allows you to add a penalty for weight size to the loss function.

Three different regularizer instances are provided; they are:

L1: Sum of the absolute weights.

L2: Sum of the squared weights.

L1L2: Sum of the absolute and the squared weights.

Because a fully connected layer occupies most of the parameters, it is prone to overfitting. One method to reduce overfitting is dropout. At each training stage, individual nodes are either "dropped out" of the net with probability 1-p or kept with

probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed.

By avoiding training all nodes on all training data, dropout decreases overfitting.

Reference:

https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/

https://en.wikipedia.org/wiki/Convolutional_neural_network

QUESTION 3

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. precision

C. accuracy

D. mean absolute error

E. coefficient of determination

Correct Answer: BC

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

QUESTION 4

DRAG DROP

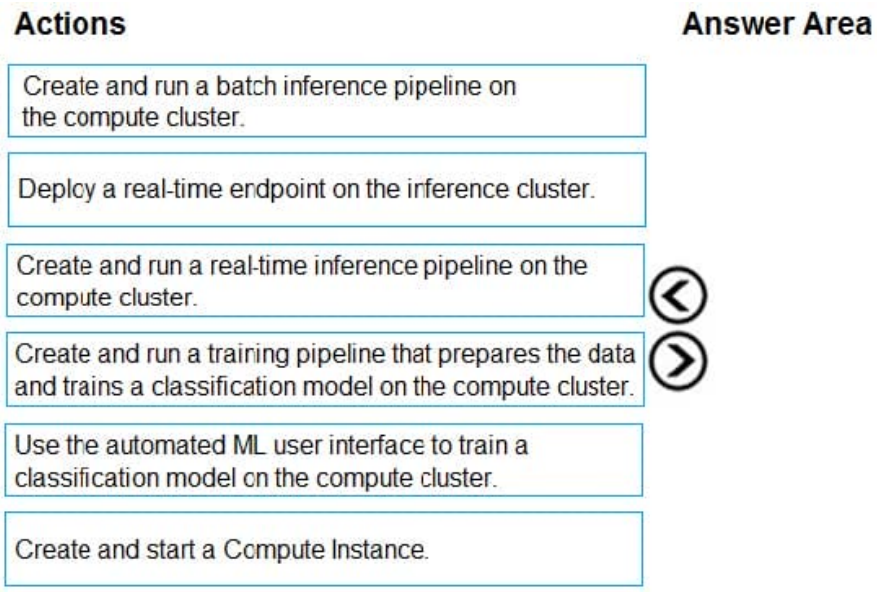
You have an Azure Machine Learning workspace that contains a CPU-based compute cluster and an Azure Kubernetes Services (AKS) inference cluster. You create a tabular dataset containing data that you plan to use to create a

classification model.

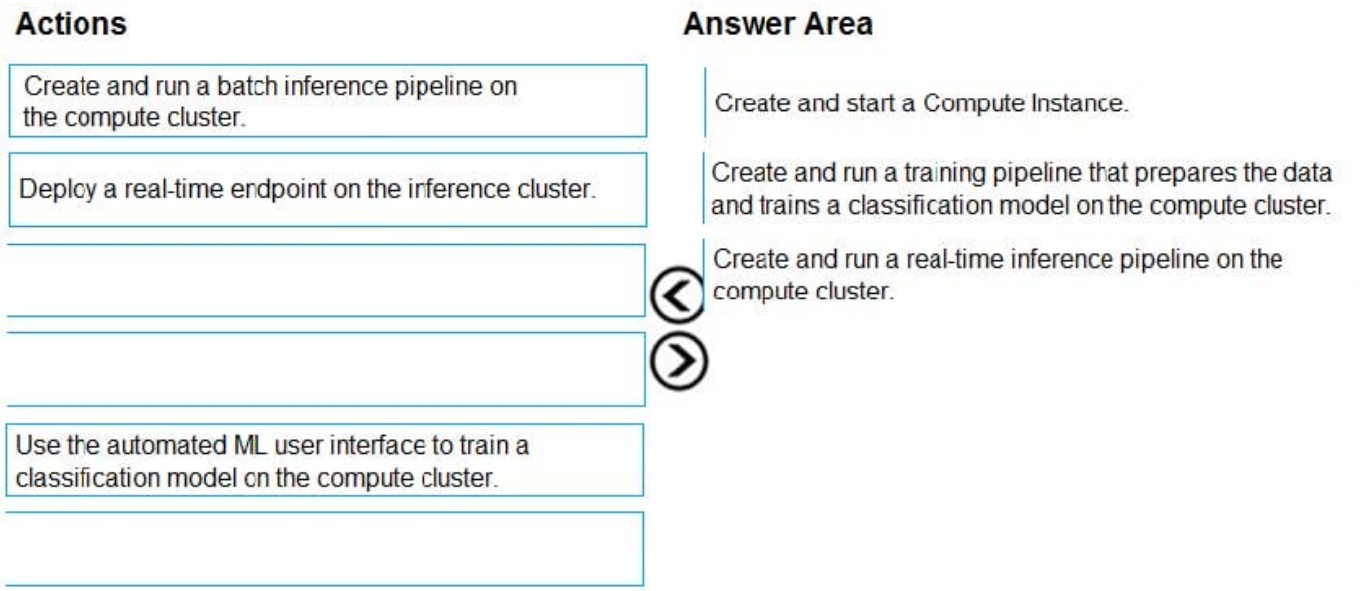
You need to use the Azure Machine Learning designer to create a web service through which client applications can consume the classification model by submitting new data and getting an immediate prediction as a response.

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

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. Yes

B. No

Correct Answer: B

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