SUCCEED AT THE
FIRST ATTEMPT!

https://developercertification.com

Questions and Answers Demo PDF

Microsoft

DP-100 Exam

Microsoft Designing and Implementing a Data Science Solution on Azure Exam

Questions & Answers
Demo

Version: 16.0

Question: 1

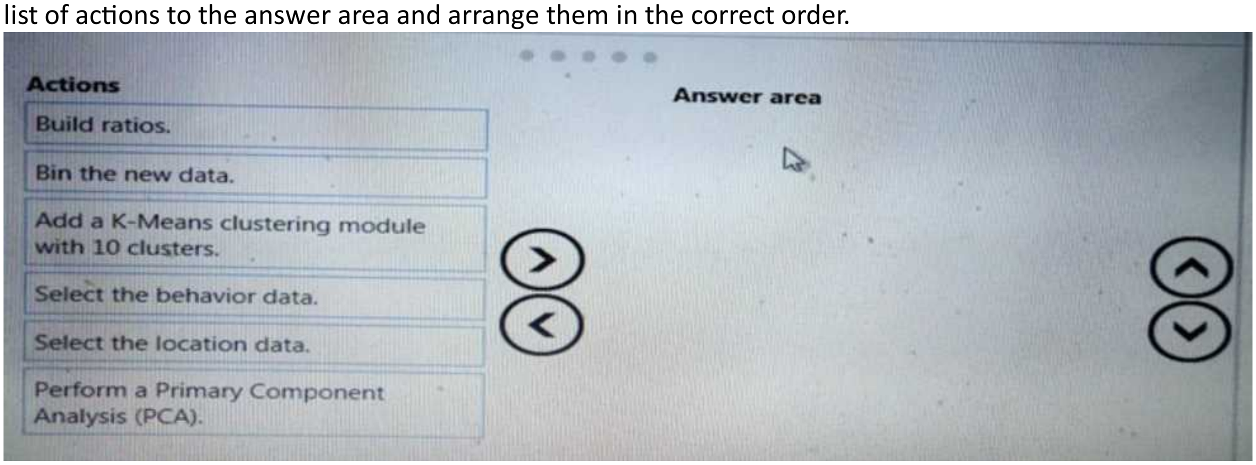
You need to resolve the local machine learning pipeline performance issue. What should you do?

A. Increase Graphic Processing Units (GPUs).
B. Increase the learning rate.
C. Increase the training iterations,
D. Increase Central Processing Units (CPUs).

Answer: A

Question: 2

DRAG DROP
You need to modify the inputs for the global penalty event model to address the bias and variance issue. 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.



Answer:

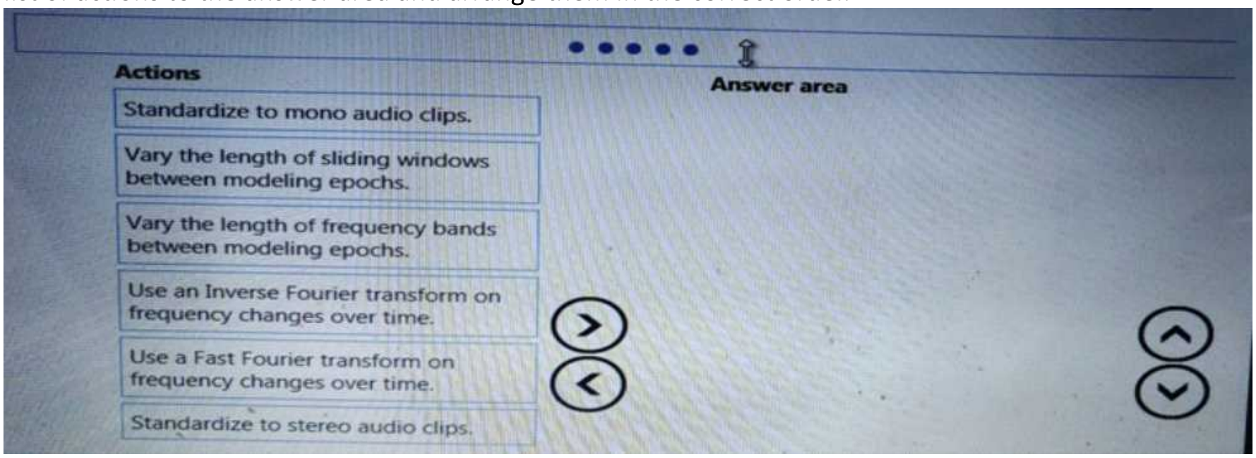
Select the Location data
Select the Behavior data
Performa a primary component Analysis (PCA)
Adda K-Means clustering module with 10 clusters.
Bin the New data
Build ratios.

Question: 3

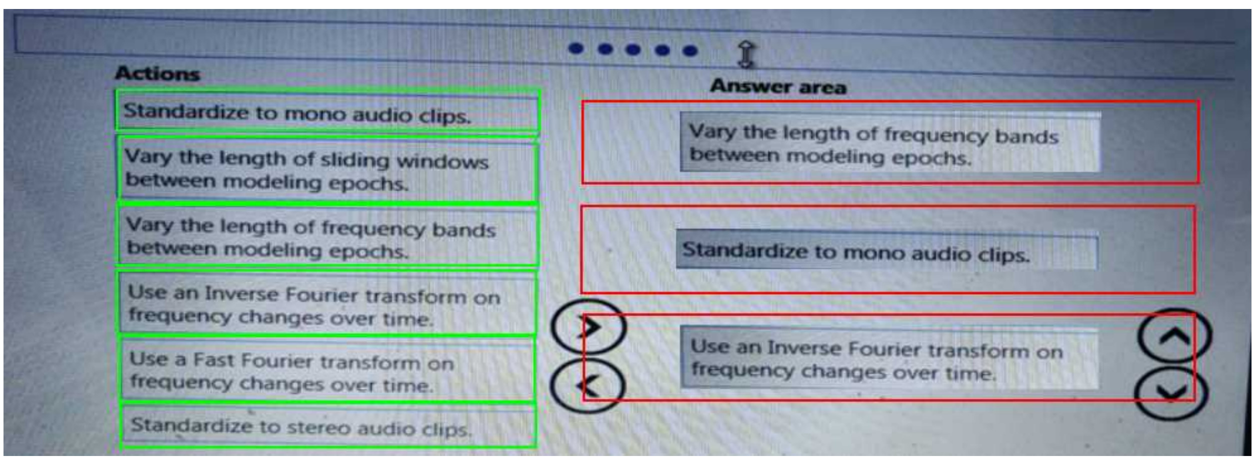
DRAG DROP

You need to define a process for penalty event detection.

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.



Answer:

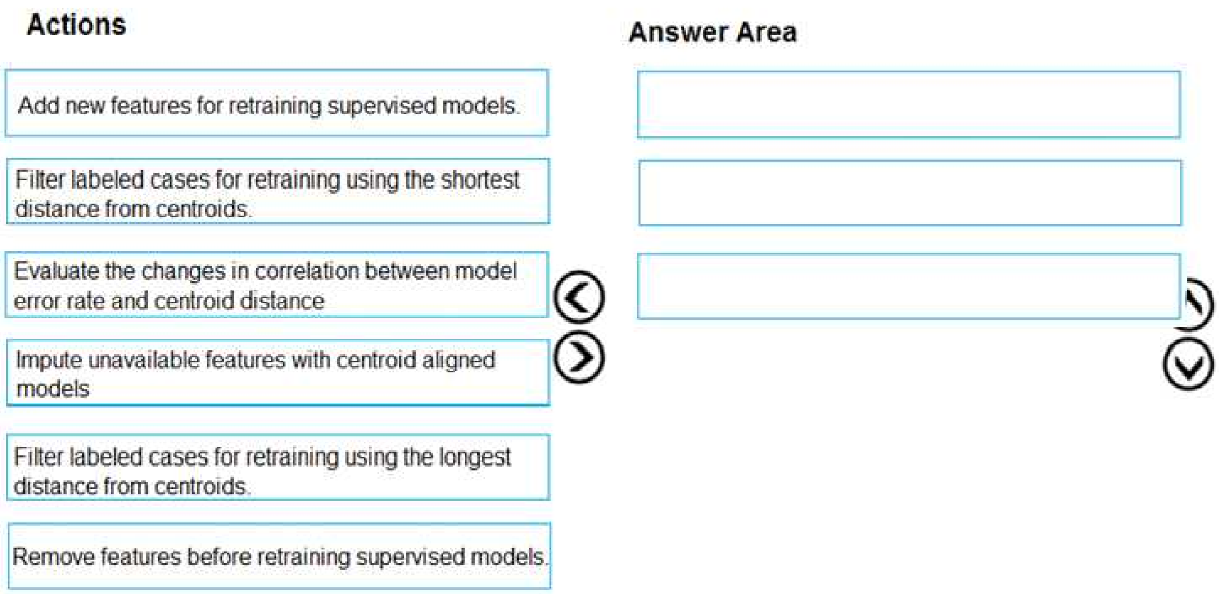


Question: 4

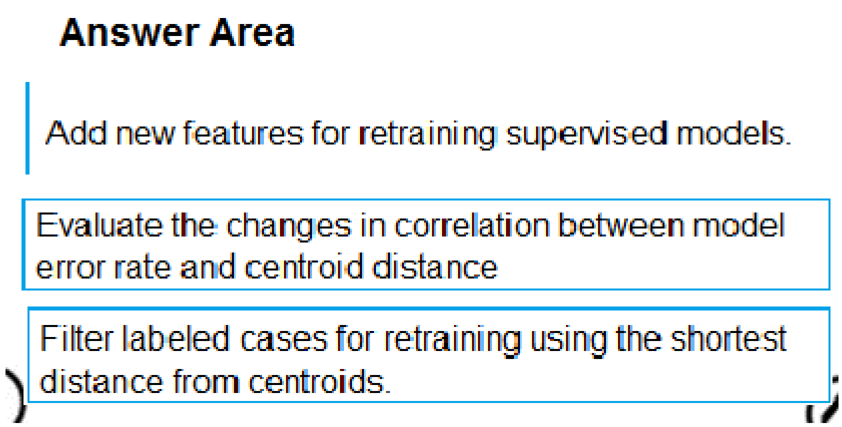
DRAG DROP

You need to define an evaluation strategy for the crowd sentiment models.

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.



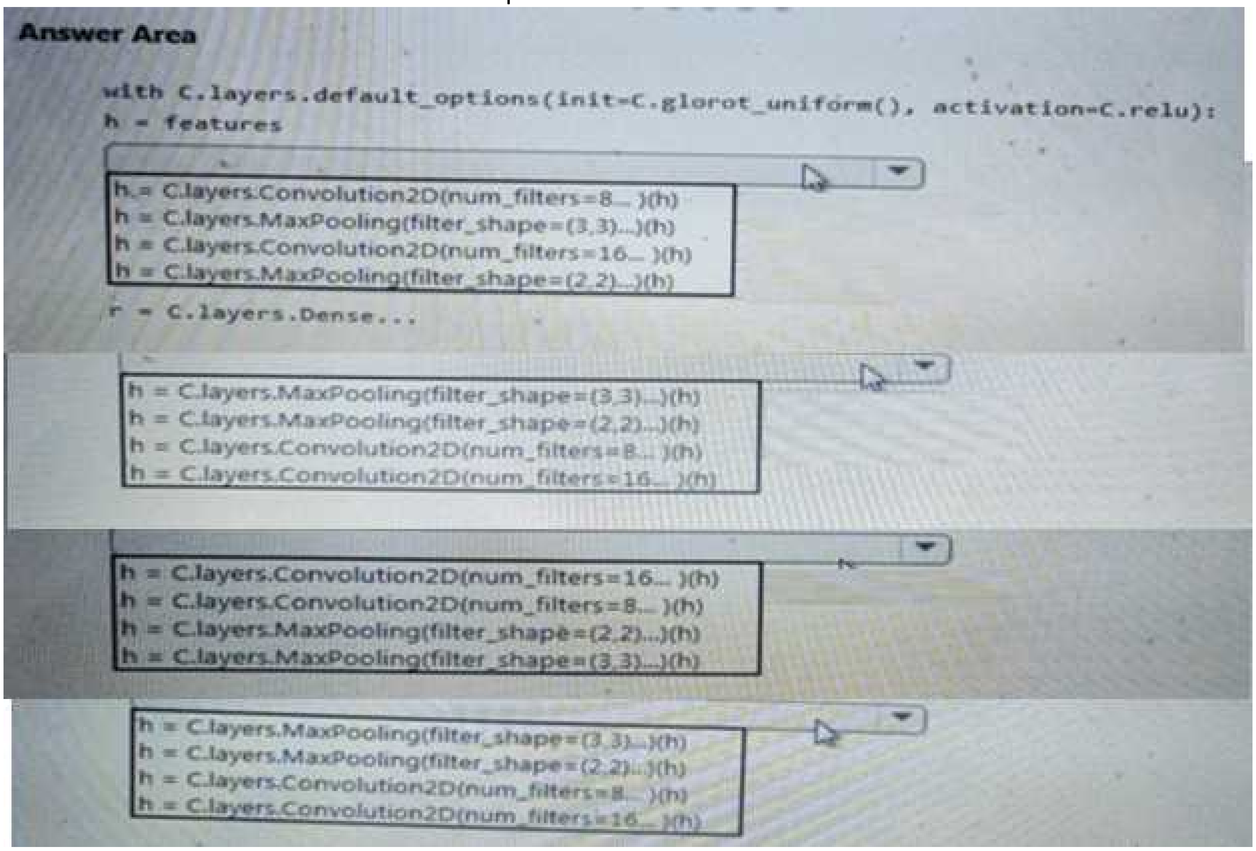
Answer:



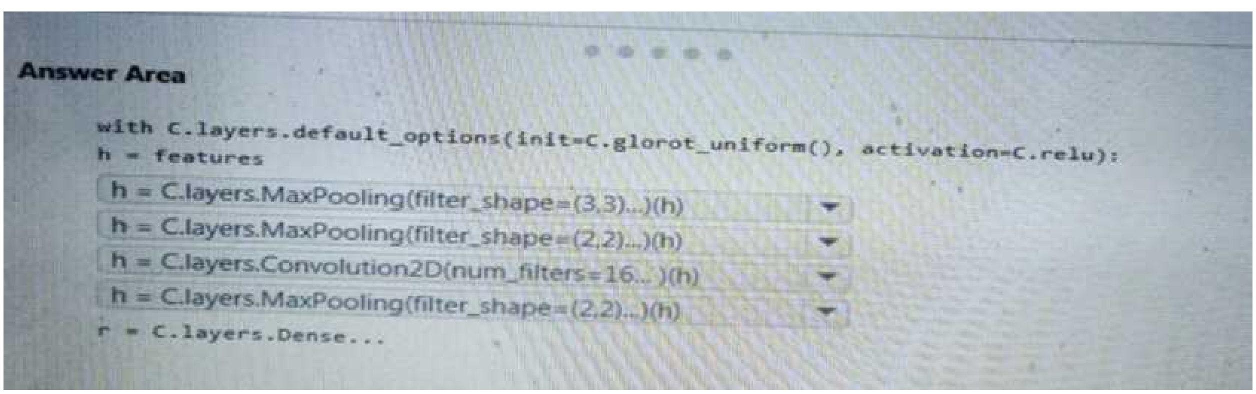
Explanation:
Scenario:
Experiments for local crowd sentiment models must combine local penalty detection data.
Crowd sentiment models must identify known sounds such as cheers and known catch phrases. Individual crowd sentiment models will detect similar sounds.
Note: Evaluate the changed in correlation between model error rate and centroid distance
In machine learning, a nearest centroid classifier or nearest prototype classifier is a classification model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation.
References:
https://en.wikipedia.org/wiki/Nearest_centroid_classifier
https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/sweep-clustering

Question: 5

HOTSPOT
You need to build a feature extraction strategy for the local 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.



Answer:



Question: 6

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

A. Streaming
B. Weight
C. Batch
D. Cosine

Answer: C

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.
References:
https://docs.microsoft.com/en-us/cognitive-toolkit/post-batch-normalization-statistics

Question: 7

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.

import pytorch as deeplearninglib
import tensorflow as deeplearninglib
import cntk as deeplearninglib

train_smapler = deeplearminglib.DistributedSampler.(penalty_video_dataset)
train_sampler = deeplearninglib.log_uniform_candidate_sampler.(penalty_video_dataset)
train_sampler = deeplearninglib.WeightedRandomSampler.(penalty_video_dataset)
train_sampler = deeplearninglib.all_candidate_sampler.(penalty_video_dataset)

```
...
train loader –
...
(train_smapler, penalty_video_dataset)
```

optimizer = deeplearninglib.optim.SGD(model.parameters().lr=0,01)
optimizer = deeplearninglib.train.GradientDescentOptimizer(learning_rate=0.10)

model = deeplearninglib.parallel.Distributed(DataParallel(model)
model = deeplearninglib.nn.parallel.DistributedDataParallelCPU(model)
model = deeplearninglib.keras.Model([
model = deeplearninglib.keras.Sequental([

```
...
train_sampler.set_epoch(epoch)
  for data, target in train_loader:
        data, target = data.to(device), target.to(device)
```

Answer:

- import pytorch as deeplearninglib
- import tensorflow as deeplearninglib
- import cntk as deeplearninglib

- train_smapler = deeplearninglib.DistributedSampler.(penalty_video_dataset)
- train_sampler = deeplearninglib.log_uniform_candidate_sampler.(penalty_video_dataset)
- train_sampler = deeplearninglib.WeightedRandomSampler.(penalty_video_dataset)
- train_sampler = deeplearninglib.all_candidate_sampler.(penalty_video_dataset)

```
...
train loader –
...
(train_smapler, penalty_video_dataset)
```

- optimizer = deeplearninglib.optim.SGD(model.parameters().lr=0,01)
- optimizer = deeplearninglib.train.GradientDescentOptimizer(learning_rate=0.10)

- model = deeplearninglib.parallel.Distributed(DataParallel(model)
- model = deeplearninglib.nn.parallel.DistributedDataParallelCPU(model)
- model = deeplearninglib.keras.Model([
- model = deeplearninglib.keras.Sequental([

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`.
References:
https://github.com/pytorch/pytorch/blob/master/torch/utils/data/distributed.py

Question: 8

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

A. Apply an analysis of variance (ANOVA).
B. Apply a Pearson correlation coefficient.
C. Apply a Spearman correlation coefficient.
D. Apply a linear discriminant analysis.

Answer: D

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.
References:
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

Question: 9

DRAG DROP
You need to define a modeling strategy for ad 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.

Action

- Implement a K-Means Clustering model.
- Use the raw score as a feature in a Score Matchbox Recommender model.
- Use the cluster as a feature in a Decision Jungle model.
- Use the raw score as a feature in a Logistic Regression model.
- Implement a Sweep Clustering model.

Answer area

Answer:

Answer area

- Implement a K-Means Clustering model.
- Use the cluster as a feature in a Decision Jungle model.
- Use the raw score as a feature in a Score Matchbox Recommender model.

Explanation:
Step 1: Implement a K-Means Clustering model
Step 2: Use the cluster as a feature in a Decision jungle model.
Decision jungles are non-parametric models, which can represent non-linear decision boundaries.
Step 3: Use the raw score as a feature in a Score Matchbox Recommender model
The goal of creating a recommendation system is to recommend one or more "items" to "users" of the system. Examples of an item could be a movie, restaurant, book, or song. A user could be a person, group of persons, or other entity with item preferences.
Scenario:
Ad response rated declined.
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.
Ad response models must support non-linear boundaries of features.
References:
https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-jungle
https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/score-matchbox-recommender

Question: 10

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

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

Answer: A

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