

# PASSTCERT

QUESTION & ANSWER

Higher Quality  
Better Service!

We offer free update service for one year  
[HTTP://WWW.PASSTCERT.COM](http://www.passtcert.com)

**Exam** : **DP-100**

**Title** : Designing and Implementing  
a Data Science Solution on  
Azure

**Version** : DEMO

1. Topic 1, Case Study 1

**Overview**

You are a data scientist in a company that provides data science for professional sporting events.

Models will be global and local market data to meet the following business goals:

- Understand sentiment of mobile device users at sporting events based on audio from crowd reactions.
- Access a user's tendency to respond to an advertisement.
- Customize styles of ads served on mobile devices.
- Use video to detect penalty events.

**Current environment**

**Requirements**

- Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and snared using social media. The images and videos will have varying sizes and formats.
- The data available for model building comprises of seven years of sporting event media. The sporting event media includes: recorded videos, transcripts of radio commentary, and logs from related social media feeds feeds captured during the sporting events.
- Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo Formats.

**Advertisements**

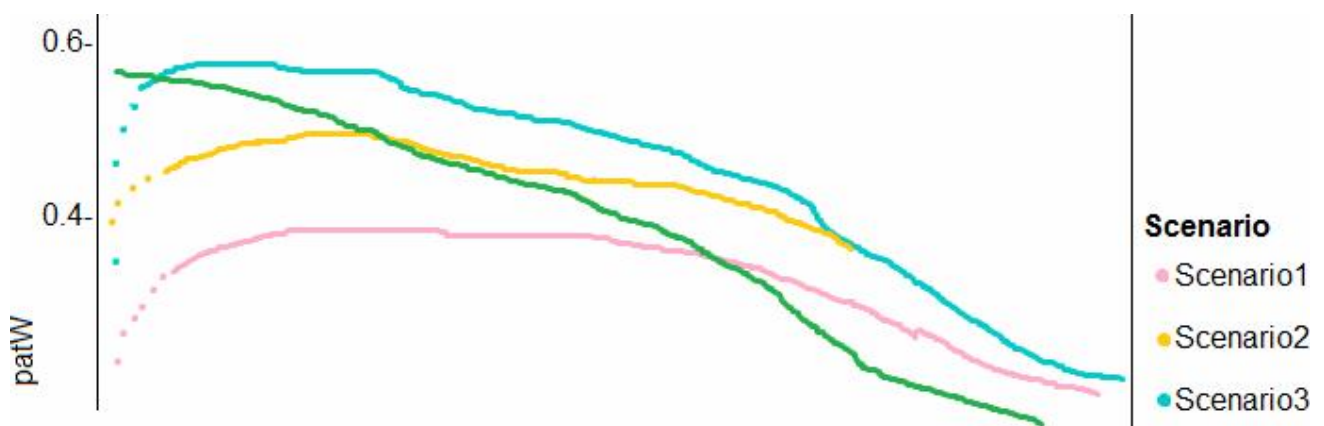
- Ad response models must be trained at the beginning of each event and applied during the sporting event.
- Market segmentation nxxlels must optimize for similar ad resporr.r history.
- Sampling must guarantee mutual and collective exclusivity local and global segmentation models that share the same features.
- Local market segmentation models will be applied before determining a user's propensity to respond to an advertisement.
- Data scientists must be able to detect model degradation and decay.
- Ad response models must support non linear boundaries features.
- The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviates from 0.1 +/-5%.
- The ad propensity model uses cost factors shown in the following diagram:

		Actual	
		1	0
Predicted	0	1	2
	1	2	1

- The ad propensity model uses proposed cost factors shown in the following diagram:

		Actual	
		1	0
Predicted	0	1	5
	1	5	1

Performance curves of current and proposed cost factor scenarios are shown in the following diagram:



### Penalty detection and sentiment

#### Findings

- Data scientists must build an intelligent solution by using multiple machine learning models for penalty event detection.
- Data scientists must build notebooks in a local environment using automatic feature engineering and model building in machine learning pipelines.
- Notebooks must be deployed to retrain by using Spark instances with dynamic worker allocation
- Notebooks must execute with the same code on new Spark instances to recode only the source of the data.
- Global penalty detection models must be trained by using dynamic runtime graph computation during training.
- Local penalty detection models must be written by using BrainScript.
- 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.
- All shared features for local models are continuous variables.
- Shared features must use double precision. Subsequent layers must have aggregate running mean and standard deviation metrics Available.

#### segments

During the initial weeks in production, the following was observed:

- Ad response rates declined.
- Drops were not consistent across ad styles.
- The distribution of features across training and production data are not consistent.

Analysis shows that of the 100 numeric features on user location and behavior, the 47 features that come from location sources are being used as raw features. A suggested experiment to remedy the bias and variance issue is to engineer 10 linearly uncorrected features.

### Penalty detection and sentiment

- Initial data discovery shows a wide range of densities of target states in training data used for crowd sentiment models.
- All penalty detection models show inference phases using a Stochastic Gradient Descent (SGD) are running too slow.
- Audio samples show that the length of a catch phrase varies between 25%-47%, depending on region.
- The performance of the global penalty detection models show lower variance but higher bias when comparing training and validation sets. Before implementing any feature changes, you must confirm the bias and variance using all training and validation cases.

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.

### Actions

### Answer Area

Add new features for retraining supervised models.

Filter labeled cases for retraining using the shortest distance from centroids.

Evaluate the changes in correlation between model error rate and centroid distance

Impute unavailable features with centroid aligned models

Filter labeled cases for retraining using the longest distance from centroids.

Remove features before retraining supervised models.

Answer:

## Actions

Add new features for retraining supervised models.

Filter labeled cases for retraining using the shortest distance from centroids.

Evaluate the changes in correlation between model error rate and centroid distance

Impute unavailable features with centroid aligned models

Filter labeled cases for retraining using the longest distance from centroids.

Remove features before retraining supervised models.

## Answer Area

Add new features for retraining supervised models.

Evaluate the changes in correlation between model error rate and centroid distance

Filter labeled cases for retraining using the shortest distance from centroids.

### 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.

Reference:

[https://en.wikipedia.org/wiki/Nearest\\_centroid\\_classifier](https://en.wikipedia.org/wiki/Nearest_centroid_classifier)

<https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/sweep-clustering>

2.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.

### 3.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.

#### Answer Area

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

▼
train_smampler = 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_smampler, 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.Sequential([
...
train_sampler.set_epoch(epoch)
for data, target in train_loader:
data, target = data.to(device), target.to(device)
..

Answer:

## Answer Area

```
import torch as deeplearninglib
import tensorflow as deeplearninglib
import cntk as deeplearninglib
```

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

### Explanation:

Box 1: import torch 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 exclusivity between local and global segmentation models that share the same features.

Box 3: optimizer = deeplearninglib.train. GradientDescentOptimizer(learning\_rate=0.10)

4.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.

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

**5.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.

Actions	Answer area
Build ratios.	
Bin the new data.	
Add a K-Means clustering module with 10 clusters.	
Select the behavior data.	
Select the location data.	
Perform a Primary Component Analysis (PCA).	

**Answer:**

<b>Actions</b>	<b>Answer area</b>
Build ratios.	Select the behavior data.
Bin the new data.	Add a K-Means clustering module with 10 clusters.
Add a K-Means clustering module with 10 clusters.	Perform a Primary Component Analysis (PCA).
Select the behavior data.	
Select the location data.	
Perform a Primary Component Analysis (PCA).	