Question #4 Section 9

Introductory Info Case study -

Overview -
You are a data scientist in a company that provides data science for professional sporting events. Models will use 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.
Assess 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 -
Media used for penalty event detection will be provided by consumer devices. Media may include images and videos captured during the sporting event and shared 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 video transcripts or radio commentary, and logs from related social media feeds captured during the sporting events.
Crowd sentiment will include audio recordings submitted by event attendees in both mono and stereo formats.

Penalty detection and sentiment -
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.

Advertisements -
During the initial weeks in production, the following was observed:
Ad response rated 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 uncorrelated features.
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 shows 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.
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.
Sampling must guarantee mutual and collective exclusively between 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.
Ad response models must support non-linear boundaries of features.
The ad propensity model uses a cut threshold is 0.45 and retraining occurs if weighted Kappa deviated from 0.1 +/- 5%.
The ad propensity model uses cost factors shown in the following diagram:

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th>0</th>
<th>1</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Performance curves of current and proposed cost factor scenarios are shown in the following diagram:
You need to implement a new cost factor scenario for the ad response models as illustrated in the performance curve exhibit. Which technique should you use?

- A. Set the threshold to 0.5 and retrain if weighted Kappa deviates +/- 5% from 0.45.
- B. Set the threshold to 0.05 and retrain if weighted Kappa deviates +/- 5% from 0.5.
- C. Set the threshold to 0.2 and retrain if weighted Kappa deviates +/- 5% from 0.6.
- D. Set the threshold to 0.75 and retrain if weighted Kappa deviates +/- 5% from 0.15.

**Answer:** A

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

The ad propensity model uses a cut threshold is 0.45 and retrains occur if weighted Kappa deviated from 0.1 +/- 5%.

Develop models

**Question #1 Section 10**

**Introductory Info Case study**
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

**To start the case study**
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.
Overview -
You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules.

Datasets -
There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapitCrimeRate</td>
<td>per capita crime rate by town</td>
</tr>
<tr>
<td>Zoned</td>
<td>proportion of residential land zoned for lots over 25,000 square feet</td>
</tr>
<tr>
<td>NonRetailAcres</td>
<td>proportion of retail business acres per town</td>
</tr>
<tr>
<td>NextToRiver</td>
<td>proximity of a property to the river</td>
</tr>
<tr>
<td>NitrogenOxideConcentration</td>
<td>nitric oxides concentration (parts per 10 million)</td>
</tr>
<tr>
<td>AvgRoomsPerHouse</td>
<td>average number of rooms per dwelling</td>
</tr>
<tr>
<td>Age</td>
<td>proportion of owner-occupied units built prior to 1940</td>
</tr>
<tr>
<td>DistanceToEmploymentCenter</td>
<td>weighted distances to employment centers</td>
</tr>
<tr>
<td>AccessibilityToHighway</td>
<td>index of accessibility to radial highways to a value of two decimal places</td>
</tr>
<tr>
<td>Tax</td>
<td>full value property tax rate per $10,000</td>
</tr>
<tr>
<td>PupilTeacherRatio</td>
<td>pupil to teacher ratio by town</td>
</tr>
<tr>
<td>ProfessionalClass</td>
<td>professional class percentage</td>
</tr>
<tr>
<td>LowerStatus</td>
<td>percentage lower status of the population</td>
</tr>
<tr>
<td>MedianValue</td>
<td>median value of owner-occupied homes in $1000s</td>
</tr>
</tbody>
</table>

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format.

Data issues -

Missing values -
The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values.

Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail.

Model fit -
The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting.

Experiment requirements -
You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset.
You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships.
You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns.

Model training -
Permutation Feature Importance -
Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model.

Hyperparameters -
You must configure hyperparameters in the model learning process to speed the learning phase. In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful.
You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs.
Testing -
You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio.

Cross-validation -
You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city’s main river. You must complete this task before the data goes through the sampling process.

Linear regression module -
When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent.

Data visualization -
You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results.
You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another. Question DRAG DROP -
You need to implement an early stopping criteria policy for model training.
Which three code segments should you use to develop the solution? To answer, move the appropriate code segments from the list of code segments 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:

**Code segments**

```python
early_termination_policy = TruncationSelectionPolicy(evaluation_interval=1,
truncation_percentage=20, delay_evaluation=5)
```

```python
import TruncationSelectionPolicy
```

```python
from azureml.train.hyperdrive
```

```python
import BanditPolicy
```

```python
early_termination_policy = BanditPolicy
(slack_factor = 0.1, evaluation_interval=1,
delay_evaluation=5)
```
You need to implement an early stopping criterion on models that provides savings without terminating promising jobs. Truncation selection cancels a given percentage of lowest performing runs at each evaluation interval. Runs are compared based on their performance on the primary metric and the lowest X% are terminated.

Example:
```python
from azureml.train.hyperdrive import TruncationSelectionPolicy
early_termination_policy = TruncationSelectionPolicy(evaluation_interval=1, truncation_percentage=20, delay_evaluation=5)
```

Incorrect Answers:
Bandit is a termination policy based on slack factor/slack amount and evaluation interval. The policy early terminates any runs where the primary metric is not within the specified slack factor / slack amount with respect to the best performing training run.

Example:
```python
from azureml.train.hyperdrive import BanditPolicy
early_termination_policy = BanditPolicy(slack_factor = 0.1, evaluation_interval=1, delay_evaluation=5)
```

Reference:
expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules.

Datasets -
There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapitaCrimeRate</td>
<td>per capita crime rate by town</td>
</tr>
<tr>
<td>Zoned</td>
<td>proportion of residential land zoned for lots over 25,000 square feet</td>
</tr>
<tr>
<td>NonRetailAcres</td>
<td>proportion of retail business acres per town</td>
</tr>
<tr>
<td>NextToRiver</td>
<td>proximity of a property to the river</td>
</tr>
<tr>
<td>NitrogenOxideConcentration</td>
<td>nitric oxides concentration (parts per 10 million)</td>
</tr>
<tr>
<td>AvgRoomsPerHouse</td>
<td>average number of rooms per dwelling</td>
</tr>
<tr>
<td>Age</td>
<td>proportion of owner-occupied units built prior to 1940</td>
</tr>
<tr>
<td>DistanceToEmploymentCenter</td>
<td>weighted distances to employment centers</td>
</tr>
<tr>
<td>AccessibilityToHighway</td>
<td>index of accessibility to radial highways to a value of two decimal places</td>
</tr>
<tr>
<td>Tax</td>
<td>full value property tax rate per $10,000</td>
</tr>
<tr>
<td>PupilTeacherRatio</td>
<td>pupil to teacher ratio by town</td>
</tr>
<tr>
<td>ProfessionalClass</td>
<td>professional class percentage</td>
</tr>
<tr>
<td>LowerStatus</td>
<td>percentage lower status of the population</td>
</tr>
<tr>
<td>MedianValue</td>
<td>median value of owner-occupied homes in $1000s</td>
</tr>
</tbody>
</table>

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format.

Data issues -

Missing values -
The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values.

Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail.

Model fit -
The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting.

Experiment requirements -
You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset.

You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships.

You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns.

Model training -

Permutation Feature Importance -
Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model.

Hyperparameters -
You must configure hyperparameters in the model learning process to speed the learning phase. In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful.

You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs.

Testing -
You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio.

Cross-validation -
You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city's main river. You must complete this task before the data goes through the sampling process.

Linear regression module -
When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent.

Data visualization -
You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results. You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another. Question DRAG DROP -
You need to correct the model fit 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. Select and Place:

<table>
<thead>
<tr>
<th>Actions</th>
<th>Answer Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add the Ordinal Regression module.</td>
<td></td>
</tr>
<tr>
<td>Add the Two-Class Averaged Perception module.</td>
<td></td>
</tr>
<tr>
<td>Augment the data.</td>
<td></td>
</tr>
<tr>
<td>Add the Bayesian Linear Regression module.</td>
<td></td>
</tr>
<tr>
<td>Decrease the memory size for L-BFGS.</td>
<td></td>
</tr>
<tr>
<td>Add the Multiclass Decision Jungle module.</td>
<td></td>
</tr>
<tr>
<td>Configure the regularization weight.</td>
<td></td>
</tr>
</tbody>
</table>
Step 1: Augment the data.
Scenario: Columns in each dataset contain missing and null values. The datasets also contain many outliers.
Step 2: Add the Bayesian Linear Regression module.
Scenario: You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules.
Step 3: Configure the regularization weight.
Regularization typically is used to avoid overfitting. For example, in L2 regularization weight, type the value to use as the weight for L2 regularization. We recommend that you use a non-zero value to avoid overfitting.
Scenario:
Model fit: The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting.
Incorrect Answers:
Multiclass Decision Jungle module:
Decision jungles are a recent extension to decision forests. A decision jungle consists of an ensemble of decision directed acyclic graphs (DAGs).
L-BFGS:
L-BFGS stands for "limited memory Broyden-Fletcher-Goldfarb-Shanno". It can be found in the wwo-Class Logistic Regression module, which is used to create a logistic regression model that can be used to predict two (and only two) outcomes.
Reference:
To start the case study:
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Overview:
You are a data scientist for Fabrikam Residences, a company specializing in quality private and commercial property in the United States. Fabrikam Residences is considering expanding into Europe and has asked you to investigate prices for private residences in major European cities. You use Azure Machine Learning Studio to measure the median value of properties. You produce a regression model to predict property prices by using the Linear Regression and Bayesian Linear Regression modules.

Datasets:
There are two datasets in CSV format that contain property details for two cities, London and Paris. You add both files to Azure Machine Learning Studio as separate datasets to the starting point for an experiment. Both datasets contain the following columns:

<table>
<thead>
<tr>
<th>Column heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapitaCrimeRate</td>
<td>per capita crime rate by town</td>
</tr>
<tr>
<td>Zoned</td>
<td>proportion of residential land zoned for lots over 25,000 square feet</td>
</tr>
<tr>
<td>NonRetailAcres</td>
<td>proportion of retail business acres per town</td>
</tr>
<tr>
<td>NextToRiver</td>
<td>proximity of a property to the river</td>
</tr>
<tr>
<td>NitrogenOxideConcentration</td>
<td>nitric oxides concentration (parts per 10 million)</td>
</tr>
<tr>
<td>AvgRoomsPerHouse</td>
<td>average number of rooms per dwelling</td>
</tr>
<tr>
<td>Age</td>
<td>proportion of owner-occupied units built prior to 1940</td>
</tr>
<tr>
<td>DistanceToEmploymentCenter</td>
<td>weighted distances to employment centers</td>
</tr>
<tr>
<td>AccessibilityToHighway</td>
<td>index of accessibility to radial highways to a value of two decimal places</td>
</tr>
<tr>
<td>Tax</td>
<td>full value property tax rate per $10,000</td>
</tr>
<tr>
<td>PupilTeacherRatio</td>
<td>pupil to teacher ratio by town</td>
</tr>
<tr>
<td>ProfessionalClass</td>
<td>professional class percentage</td>
</tr>
<tr>
<td>LowerStatus</td>
<td>percentage lower status of the population</td>
</tr>
<tr>
<td>MedianValue</td>
<td>median value of owner-occupied homes in $1000s</td>
</tr>
</tbody>
</table>

An initial investigation shows that the datasets are identical in structure apart from the MedianValue column. The smaller Paris dataset contains the MedianValue in text format, whereas the larger London dataset contains the MedianValue in numerical format.

Data issues:
Missing values:
The AccessibilityToHighway column in both datasets contains missing values. The missing data must be replaced with new data so that it is modeled conditionally using the other variables in the data before filling in the missing values.
Columns in each dataset contain missing and null values. The datasets also contain many outliers. The Age column has a high proportion of outliers. You need to remove the rows that have outliers in the Age column. The MedianValue and AvgRoomsInHouse columns both hold data in numeric format. You need to select a feature selection algorithm to analyze the relationship between the two columns in more detail.

Model fit:
The model shows signs of overfitting. You need to produce a more refined regression model that reduces the overfitting.

Experiment requirements:
You must set up the experiment to cross-validate the Linear Regression and Bayesian Linear Regression modules to evaluate performance. In each case, the predictor of the dataset is the column named MedianValue. You must ensure that the datatype of the MedianValue column of the Paris dataset matches the structure of the London dataset.
You must prioritize the columns of data for predicting the outcome. You must use non-parametric statistics to measure relationships.
You must use a feature selection algorithm to analyze the relationship between the MedianValue and AvgRoomsInHouse columns.

Model training:
Permutation Feature Importance:
Given a trained model and a test dataset, you must compute the Permutation Feature Importance scores of feature variables. You must be determined the absolute fit for the model.

**Hyperparameters**
- You must configure hyperparameters in the model learning process to speed the learning phase. In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful.
- You are concerned that the model might not efficiently use compute resources in hyperparameter tuning. You also are concerned that the model might prevent an increase in the overall tuning time. Therefore, must implement an early stopping criterion on models that provides savings without terminating promising jobs.

**Testing**
- You must produce multiple partitions of a dataset based on sampling using the Partition and Sample module in Azure Machine Learning Studio.

**Cross-validation**
- You must create three equal partitions for cross-validation. You must also configure the cross-validation process so that the rows in the test and training datasets are divided evenly by properties that are near each city’s main river. You must complete this task before the data goes through the sampling process.

**Linear regression module**
- When you train a Linear Regression module, you must determine the best features to use in a model. You can choose standard metrics provided to measure performance before and after the feature importance process completes. The distribution of features across multiple training models must be consistent.

**Data visualization**
- You need to provide the test results to the Fabrikam Residences team. You create data visualizations to aid in presenting the results.
- You must produce a Receiver Operating Characteristic (ROC) curve to conduct a diagnostic test evaluation of the model. You need to select appropriate methods for producing the ROC curve in Azure Machine Learning Studio to compare the Two-Class Decision Forest and the Two-Class Decision Jungle modules with one another. Question DRAG DROP - You need to implement early stopping criteria as stated in the model training requirements.
- Which three code segments should you use to develop the solution? To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.
- NOTE: More than one order of answer choices is correct. You will receive the credit for any of the correct orders you select.

Select and Place:
early_termination_policy = TruncationSelectionPolicy
evaluation_interval=1, truncation_percentage=20, delay_evaluation = 5

import BanditPolicy

import TruncationSelectionPolicy

early_termination_policy = BanditPolicy (slack_factor = 0.1, evaluation_interval = 1, delay_evaluation = 5)

from azureml.train.hyperdrive

early_termination_policy = MedianStoppingPolicy (evaluation_interval = 1, delay_evaluation=5)

import MedianStoppingPolicy
**Answer:**

Step 1: from azureml.train.hyperdrive
Step 2: Import TruncationSelectionPolicy

Truncation selection cancels a given percentage of lowest performing runs at each evaluation interval. Runs are compared based on their performance on the primary metric and the lowest X% are terminated.

Scenario: You must configure hyperparameters in the model learning process to speed the learning phase. In addition, this configuration should cancel the lowest performing runs at each evaluation interval, thereby directing effort and resources towards models that are more likely to be successful.

Step 3: early_termination_policy = TruncationSelectionPolicy.

Example:

```python
from azureml.train.hyperdrive import TruncationSelectionPolicy
early_termination_policy = TruncationSelectionPolicy(evaluation_interval=1, truncation_percentage=20, delay_evaluation=5)
```

In this example, the early termination policy is applied at every interval starting at evaluation interval 5. A run will be terminated at interval 5 if its performance at interval 5 is in the lowest 20% of performance of all runs at interval 5.

Incorrect Answers:

Median:
Median stopping is an early termination policy based on running averages of primary metrics reported by the runs. This policy computes running averages across all training runs and terminates runs whose performance is worse than the median of the running averages.

Slack:
Bandit is a termination policy based on slack factor/slack amount and evaluation interval. The policy early terminates any runs where the primary metric is not within the specified slack factor / slack amount with respect to the best performing training run.

Reference:

Develop models
For More exams visit https://killexams.com/vendors-exam-list