## Question 1

**Domain:** Run experiments and train models

During the pre-processing phase in an ML pipeline, you have to make transformations on your data. You, as an experienced SQL programmer, decide to write SQL scripts to solve the problem. You want to use the Apply SQL Transformation module in the ML Designer.

Which practice should you follow? Select two:

- A. Avoid using RIGHT OUTER JOIN
- B. Avoid using LEFT OUTER JOIN
- C. Always use VIEW if you want to execute INSERT / UPDATE statements
- D. Never user VIEW if you want to execute INSERT / UPDATE statements

## Explanation:

**Answers: A and D**

- Option A is CORRECT because the SQL engine used by this module is SQLite. One of its limitations is that while LEFT OUTER JOIN is implemented, RIGHT OUTER JOIN is not available!
- Option B is incorrect because the SQL engine used by this module is SQLite. LEFT OUTER JOIN is implemented in the SQLite engine, so you can use them.
- Option C is incorrect because the SQL engine used by this module is SQLite. One of its limitations is that views are read-only, i.e. cannot be used to write the underlying data.
- Option D is CORRECT because while you can create views in SQLite, views are read-only, i.e. INSERT, UPDATE and DELETE operations cannot be directly used with them.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/apply-sql-transformation#technical-notes

## Question 2

Domain: Manage Azure resources for machine learning

You have 100 CSV files in Azure Blob Storage which you have to use to train your ML model. The files contain measurement data collected from manufacturing machines and have been collected in order to analyse causes of malfunctions. Each row in the files is a snapshot of machine parameters at a given time.Using the ML Designer, you have to use the data in CSV files as input for your machine learning pipeline, ensuring reusability and versioning of data and minimizing the time to load during running experiments.What should you do?

- A. Register the files as a File Dataset in your ML workspace; add the Dataset module to your pipeline
- B. Add an Import Data module to your pipeline and configure it for accessing the files; set the Regenerate output = Yes
- C. Register the files as a Tabular Dataset in your ML workspace; add the Dataset module to your pipeline
- D. Add an ImportData module to your pipeline and configure it for accessing the files; set the Regenerate output = No

## Explanation:

**Answer: C**

- Option A is incorrect because structured files (like CSVs) have to be registered as Tabular datasets, File type is not suitable for structured data; in addition, ML Designer supports only processing Tabular datasets
- Option B is incorrect because the Import Data module imports data directly, without registering a dataset in the ML workspace. The Import Data module reloads a new set of data each time the pipeline runs, consuming excess resources.
- Option C is CORRECT because the recommended practice for getting data into the ML pipeline without repeating the input operation for each run is registering the data as a Dataset. Structured files (like CSVs) have to be registered as Tabular datasets. The registered datasets can be found in the module palette, under Datasets and can be used like any other modules. By having a dataset registered, additional features as versioning and data monitoring becomes available.
- Option D is incorrect because the Import Data module imports data directly, without registering a dataset in the ML workspace, i.e. the reusability requirement is not satisfied.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-designer-import-data
- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/import-data

## Question 3

Domain: Run experiments and train models

You are developing an ML pipeline which consists of several steps. In the chain of steps you need to pass the data from Step1 to Step2, for further processing. You are using Python SDK.How can you achieve this goal?

- A. Pass a Dataset object as argument from Step1 to Step2
- B. Define a PipelineData object in the pipeline definition script and use it as "outputs=" and "inputs=", respectively
- C. Define a PipelineData object in Step1 and pass it to Step2 as "outputs=" and "inputs=", respectively
- D. Define a PipelineParameter and use it to pass the dataset from Step1 to Step2

## Explanation:

**Answer: B**

- Option A is incorrect because you can use datasets for data that are available persistently all over the ML workspace. For intermediate data moving between pipeline steps, use the PipelineData object.
- Option B is CORRECT because PipelineData objects exist beyond single pipeline steps, they must be defined in the pipeline definition script. In an ML pipeline, PipelineData object is designed to be used for passing the resulting temporary data from one step to another.
- Option C is incorrect because PipelineData objects exist beyond single pipeline steps, so they have to be defined in the pipeline definition script.
- Option D is incorrect because PipelineParameter object is used to pass model parameters for pipeline runs. It's not appropriate for passing datasets.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-move-data-in-out-of-pipelines

## Question 4

Domain: Run experiments and train models

Your task is to ingest data from a CSV file, to train an ML model, using a regression algorithm and evaluate the model's performance. In order to do that, you need to build an ML Designer pipeline. Which of the following modules should you drag onto the canvas, in what order?

- Load data
- Import data
- Evaluate model
- Score model
- Train model
- Split data

- A. Load data -> Split data -> Train model -> Evaluate model
- B. Import data -> Split data -> Train model ->; Score model -> Evaluate model
- C. Import data -> Score model -> Split data -> Evaluate model -> Train model
- D. Load data -> Split data -> Train model -> Score model -> Evaluate model

## Explanation:

**Answer: B**

- Option A is incorrect because there is no module named Load Data on the ML designer palette; scoring must be executed before evaluation of results.
- Option B is CORRECT because this is the right order of steps you should follow.
- Option C is incorrect because splitting data needs to be done before training; scoring and evaluating can be executed on the results of the training step.
- Option D is incorrect because there is no module named Load Data on the ML designer palette. You can use Import data.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-designer-automobile-price-train-score

## Question 5

Domain: Run experiments and train models

You've built an ML pipeline which trains a regression model to predict inventory levels for the next month. You have completed several runs and you need to decide which of them gives the best performance. You use the output of the *Evaluate model* designer module, i.e.:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- Coefficient of Determination

You select the run as best performer, for which...

- A. ...MAE is low; RMSE is low; Coefficient of Determination is low
- B. ...MAE is high; RMSE high; Coefficient of Determination is high
- C. ...MAE high; RMSE low; Coefficient of Determination is high
- D. ...MAE is low; RMSE is low; Coefficient of Determination is high

## Explanation:

**Answer: D**

- Option A is incorrect because for Coefficient of Determination, the higher values are favorable.
- Option B is incorrect because the value of metrics representing the size of the error (MA, RMSE) should be as low as possible.
- Option C is incorrect because the value of metrics representing the size of the error (MA, RMSE) should be as low as possible.
- Option D is CORRECT because MAE and RMSE both measure how close the model's predicted values are to the actual results. The lower these values the better. Coefficient of Determination shows "how powerful" the model is, in terms of predictions. Higher values (close to 1) represent higher predicting power..

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-designer-automobile-price-train-score#evaluate-models

## Question 6

Domain: Run experiments and train models

You are writing an ML experiment script using the Python SDK. This code snippet is used to turn on the logging for the run:

```
...
# Start logging data from the experiment
run = experiment.start_logging()

# load the dataset and count the rows
train_data = pd.read_csv('train_data.csv')
# set metrics to log
row_count = (len(train_data))

# Log the row count
run.log('rowcount', row_count)
...
```

After completing the run, which two methods can you use to retrieve the metrics logged for the run?

- A. Use run.get_log()
- B. Use the JSON from run.get_metrics()
- C. Use RunDetails(run).show()Jupyter widget
- D. Use run.get_details()

## Explanation:

**Answers: B and C**

- Option A is incorrect because the run object doesn't have a get_log()method.
- Option B is CORRECT because the JSON object returned by the get_metrics() method is the way of getting the metrics dogged during a run.
- Option C is CORRECT because when working in a Jupyter notebook, the show() widget is a nice way of displaying metrics on the run.
- Option D is incorrect because the get_details() method is used to get some "descriptive" information (definition, status, log files etc.) on the current run, but no metrics.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-track-experiments

## Question 7

Domain: Manage Azure resources for machine learning

You are tasked to set up a ML environment for running experiments. While configuring the compute environment for your experiments, your priority is to provision the necessary capacity but keeping costs as low as possible.

Which is not the way of optimizing costs?

- A. Develop code in local, low-cost environment
- B. Use Azure Kubernetes Service
- C. Use managed computes that start automatically on-demand and stop when not needed
- D. Set automatic scaling for computes, based on the workload

## Explanation:

**Answer: B**

- Option A is incorrect because using your own local environment while developing code generates no extra cost, not even for long execution times.
- Option B is CORRECT because AKS is recommended for high-scale production environments. Not the best way to keep costs low.
- Option C is incorrect because in a cloud environment, utilizing the pay-as-you-go model, automatically launching and stopping computes is one of the best ways of preventing costs from "exploding".
- Option D is incorrect because in the cloud environment, you can set up compute clusters which provide the necessary compute power while scaling up and down automatically, according to the actual demand.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-compute-target

## Question 8

Domain: Manage Azure resources for machine learning

For your ML model training tasks you have to provision an appropriate compute resource. Training runs are executed periodically and in idle periods you don't need the resource, but during training runs, the compute has to cope with really high loads.

How do you fulfil the requirement while avoiding excess cost?

- A. Provision a managed Azure Compute Instance
- B. Attach a remote VM as a compute
- C. Provision Azure ML Compute Cluster
- D. Make use of Azure Kubernetes Service

## Explanation:

**Answer: C**

- Option A is incorrect because it is a single VM which doesn't scale down when idle and doesn't scale up with peak demand; it has to be managed manually which induces a risk of unexpected costs.
- Option B is incorrect because remote VMs are not managed by Azure, scale up/down must be separately managed, which might not fulfil the requirements of peaks in training runs.
- Option C is CORRECT because Azure compute clusters are the most flexible and cost-effective environments for train experiments. Clusters, as a multi-node environment can scale up for training runs and autoscale to 0 nodes when idle.
- Option D is incorrect because AKS is a great and powerful compute environment inference runs but it is unnecessarily expensive, therefore not recommended for training/development purposes. Use Compute Clusters instead.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-compute-target

## Question 9

Domain: Deploy and operationalize machine learning solutions

You have a forecasting ML model in production. After several months of live operation, you notice that the model has been degrading in terms of predictive accuracy. You have to introduce tools to detect and quantify the problem.

What is the best way to do that?

- A. Collect new data; add a new ImportData step to your pipeline; import the new data and use it further on
- B. Collect new data; add the new data to the training dataset; retrain the model; configure a DataDriftDetector
- C. Collect new data; use the training dataset as baseline; register the new data as target dataset; configure a DataDriftDetector
- D. Collect new data as a new version of the training dataset; write a Python script to profile and compare them; retrain the model if necessary; publish the new model

## Explanation:

**Answer: C**

- Option A is incorrect because so that you can use Azure's data drift monitoring capabilities, registered datasets must be used, i.e. the direct load functionality of the ImportData module doesn't fit for the solution.
- Option B is incorrect because DataDriftDetector needs two registered datasets which it can use to compare from time to time, in order to determine any significant change in the profile of the data.
- Option C is CORRECT because in order to provide the early detection of drifting data, you need two registered datasets, a baseline and a target which can then be regularly compared and evaluated by using the Azure ML DataDriftDetector class.
- Option D is incorrect because you don't need to write your own script. Use Azure ML DataDriftMonitor instead, which is specifically designed for this purpose.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-datasets

## Question 10

Domain: Manage Azure resources for machine learning

Your company is storing hundreds of GBs of data in a distributed Cosmos DB. This huge amount of data contains tons of valuable information about sales transactions and the company is going to make use of it by running machine learning models against it. Your task is to design how to feed Azure ML processes with Cosmos DB data.

Which option should you choose to get the data to Azure ML in the cheapest and most effective way?

- A. Transfer data to Azure Blob Storage and register Blob Storage as datastore
- B. Register Cosmos DB as a data store
- C. Register Cosmos DB as dataset
- D. Transfer data to Azure SQL Database and register it as a datastore

## Explanation:

**Answer: A**

- Option A is CORRECT because If you need to ingest data from Cosmos DB, the cheapest and most powerful way is transferring it to Blob Storage (e.g. by using Data Factory) and register it as a datastore.
- Option B is incorrect because Cosmos DB currently is not supported as a datastore.
- Option C is incorrect because Cosmos DB currently is not supported as a datastore. (If it would be, it should be registered as a datastore.)
- Option D is incorrect because Cosmos DB is a no-SQL data storage. Transferring data to a structured format would be resource intensive. In addition, SQL Database is not the most cost effective data storage compared to Blob Storage.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-data#datastores

## Question 11

Domain: Manage Azure resources for machine learning

Your company gathers a lot of data from distributed sensors via an Internet of Things network. Raw data is accumulated in an Azure Blob Storage container. You are going to use this data in your machine learning experiments, therefore you need to register the storage as a data store in your ML workspace.

Which two authentication methods can you choose?

- A. Account_key
- B. Service principal
- C. SQL authentication
- D. SAS token

## Explanation:

**Answer: A and D**

- Option A is CORRECT because one of the authentication methods for Storage Account is using the account key (found on the Settings pane of the SA).
- Option B is incorrect because service principal is not a valid way of authenticating to a Storage Account. It can be used in the case of Azure Data Lake storage or Azure SQL.
- Option C is incorrect because SQL authentication can be used for accessing SQL databases. It is not applicable for blob storages.
- Option D is CORRECT because for granting access to Storage Account, Shared Access Keys (SAS) can also be used (found on the Settings pane of the SA).

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-access-data

## Question 12

Domain: Manage Azure resources for machine learning

You have an existing, powerful Databricks cluster in your local environment and, instead of provisioning an Azure ML compute, you decide to use it in your ML experiments as a training compute.

```
# Create the compute
databricks_cluster = [select the passing code segment here](ws,
compute_name, compute_config)
databricks_cluster.wait_for_completion(show_output=True)
```

Which code segment should you choose to complete the code?

- A. ComputeTarget.create
- B. ComputeTarget.provisioning_configuration
- C. ComputeTarget.attach
- D. ComputeTarget.use

## Explanation:

**Answer: C**

- Option A is incorrect because the create() method of the ComputeTarget class is used for adding Azure-managed computes to the workspace. It cannot be used for external computes.
- Option B is incorrect because provisioning_configuration is a parameter of the create() method.
- Option C is CORRECT because compute targets defined outside the Azure space (typically a Databricks cluster) can be added to the workspace by the attach() method.
- Option D is incorrect because use is not a valid method name for compute targets.

**Reference:**

- https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.compute.amlcompute.amlcompute?view=azure-ml-py

## Question 13

Domain: Run experiments and train models

For running your ML experiments, you want to create a separate Python script for configuring and running the experiment, and store it in a folder for future use. While writing the script, there is a list of key steps you have to include in a specific order.

Which of the following options reflects the right order of the required steps within the script?

- A. Workspace() -> Compute target() -> RunConfiguration() -> Experiment.submit()
- B. Run.get_context() -> Experiment() -> ScriptRunConfiguration() -> Workspace()
- C. Workspace() -> Run.get_context() -> ScriptRunConfig() -> Experiment.submit()
- D. Workspace() -> ScriptRunConfig() -> Run.get_context() -> Experiment()

## Explanation:

**Answer: C**

- Option A is incorrect because defining compute targets is not part of the experiment script; instead of RunConfiguration() the Run.get_context() has to be used.
- Option B is incorrect because connecting to a Machine Learning workspace must be the very first step.
- Option C is CORRECT because the very first step is connection to an ML workspace, then the run context for running the script has to be retrieved, then a ScriptRunConfig is needed to define the script to be run. Finally, you have to submit the experiment by the submit() method.
- Option D is incorrect because the Experiment.submit() method must be used as the last step, in order to run the experiment.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-1st-experiment-sdk-train

## Question 14

Domain: Run experiments and train models

For your machine learning experiments, you are going to use the Scikit-Learn framework. You want to keep your Python code defining the run configuration as simple and compact as possible.

Which is the best way to achieve this goal?

- A. Use CondaDependencies.create(conda_packages=['scikit-learn']...) to define the environment and use it as the environment_definition parameter of an Estimator

- B. Import the SKLearn package and use the SKLearn pre-configured estimator to define the run configuration
- C. Import the Estimator package and use Estimator with parameter conda_packages=['scikit-learn']

- D. You don't need to set anything special because the azure ML environments are pre-configured for the Scikit-Learn framework

## Explanation:

**Answer: B**

- Option A is incorrect because while this solution can be used to set the run configuration, in the case of Scikit-Learn framework, using the pre-configured SKLearn estimator is the best solution.
- Option B is CORRECT because the simplest way to define the run configuration for the learning script built on a given ML framework (like Scikit-Learn) is to use the framework-specific estimators
- Option C is incorrect because while this can be used to set the run configuration, in the case of Scikit-Learn framework, using the pre-configured SKLearn estimator is the best solution.
- Option D is incorrect because the specific ML packages (like ScikitLearn, PyTorch etc.) are not contained in the base configuration. If you need Scikit-Learn, you have to add it to your run configuration (either via ScriptRunConfig or via an estimator).

**Reference:**

- https://docs.microsoft.com/en-us/python/api/azureml-train-core/azureml.train.estimator?view=azure-ml-py&preserve-view=true

## Question 15

Domain: Manage Azure resources for machine learning

For your ML experiments, you need to process CSV data files. Size of your files is about 2GB each. Your training script loads the ingested data to a pandas dataframe object. During the first run, you get an "Out of memory" error. You decide to double the size of the compute's memory (which is 16GB currently).

Is this a possible solution to the problem?

- A. Yes
- B. No

## Explanation:

**Answer: A**

- Option A is CORRECT because the data loaded from a CSV file can expand even as much as 10 times when loaded into a dataframe in memory. It is recommended to set the size of the memory at least two times the size of the input data.
- Option B is incorrect because a typical reason for "Out of memory" errors during this process is that data loaded from a CSV file expands significantly when loaded to a dataframe. Extending the compute memory is one possible solution.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-register-datasets

## Question 16

Domain: Run experiments and train models

While running your ML experiments, you make some changes in the underlying data in one of the datasets used during the preparation step in your. After running your pipeline, you notice that output doesn't change, regardless of changes in the data.

What should you do in order to solve the problem? Select two:

- A. In your PythonScriptStep, set allow_reuse = False
- B. In your PythonScriptStep, set allow_reuse = True
- C. In your PythonScriptStep, set regenerate_outputs=True
- D. In the experiment.submit, set regenerate_outputs=True
- E. In the experiment.submit, set regenerate_outputs=False

## Explanation:

**Answers: A and D**

- Option A is CORRECT because reusing the output of previous steps in the pipeline is enabled by default. You have to disable it if you want to prevent steps, use previous outputs.
- Option B is incorrect because this is the default setting for reusing outputs from the previous run. Leaving it "True" will not solve the problem.
- Option C is incorrect because there is no "regenerate_output" in the PythonScriptStep.
- Option D is CORRECT because setting regenerate_outputs=True at the experiment level forces all the steps in the pipeline NOT to use results from previous runs.
- Option E is incorrect because setting regenerate_outputs=False at the lets all the steps in the pipeline use results from previous runs, i.e. it won't solve the problem.

**Reference:**

- https://docs.microsoft.com/en-us/python/api/azureml-pipeline-steps/azureml.pipeline.steps.python_script_step.pythonscriptstep?view=azure-ml-py

## Question 17

Domain: Manage Azure resources for machine learning

Machine Learning workspace is the root object for running ML experiments in Azure. You have created one so that you can train your models, run auto ML experiments, build reusable workflows, evaluate models etc. You have to grant access to a number of team members to resources in your workspace.

Which tools can you use to complete this task?

- A. ML Studio + Azure Portal + Python SDK
- B. Python SDK + Azure CLI
- C. Portal + ML Studio + Azure CLI + SDK
- D. Azure Portal + Azure CLI

## Explanation:

**Answer: D**

- Option A is incorrect because neither ML studio nor the SDKs have workspace management features.
- Option B is incorrect because the SDKs don't have workspace management features.
- Option C is incorrect because neither ML studio nor the SDKs have workspace management features.
- Option D is CORRECT because for your access management tasks you have to use the Portal or Azure CLI. These are the tools Azure provides for access management.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-workspace#workspace-management

## Question 18

Domain: Deploy and operationalize machine learning solutions

Your training dataset, besides many others, contains the following attributes: row_id, transaction_date, transaction_value. In order to optimize the training runs, you need to do some feature engineering on these data. You are using the autoML functionality in Azure ML Studio.

Which actions should you take?

- A. Normalize the row_id values; write a custom Python code to transform transaction_date to additional year, month etc. columns; replace the missing transaction_value with random numbers.

- B. Leave all the feature engineering tasks all to Azure autoML because it removes the id column, generates new features from "date" type columns, and replaces missing values in numeric columns with the average of the not null values.
- C. In Azure autoML settings, set the "Drop high cardinality features = True" for row_id; for transaction_date set "Generate additional features = True"; for transaction_values set "Impute missing values = True".

- D. Drop the row_id column as irrelevant from training perspective; write a custom Python code to derive additional year, month etc. columns from transaction_date; remove rows with missing transaction_values.

## Explanation:

**Answer: B**

- Option A is incorrect because there is no sense in normalizing an attribute with kind of serial numbers. The attribute should be dropped instead.
- Option B is CORRECT because autoMLs built-in featurization removes the row_id, generates some derived features from transaction_date, fills up missing values with the average of the existing values in the transaction_value. All this is done automatically.
- Option C is incorrect because when featurization is enabled, these transformations are applied automatically by autoML. Via the ML Studio, there are only limited options to modify featurization settings.
- Option D is incorrect because in general, all the featurization can be left for the autoML. Specifically, removing rows that have missing values will affect the result of the training process, so it is incorrect.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features#featurization

## Question 19

Domain: Manage Azure resources for machine learning

For your ML experiments, you need to process CSV data files. Size of your files is about 10GB each. Your training script loads the ingested data to a pandas dataframe object. During the runs, you get an "Out of memory" error. You decide to convert the files to Parquet format and process it partially, i.e. loading only the columns relevant from the modelling point of view.

Does it solve the problem?

- A. Yes
- B. No

## Explanation:

**Answer: A**

- Option A is CORRECT because the data loaded from a CSV file can expand significantly when loaded into a dataframe in memory. Converting it to the columnar Parquet format is a viable solution because it enables loading selected columns which are necessary for the training process.
- Option B is incorrect because using the columnar Parquet format instead of CSV can be used to optimize memory consumption, therefore it is a good solution.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-register-datasets
- https://docs.microsoft.com/en-us/azure/machine-learning/concept-optimize-data-processing

## Question 20

Domain: Deploy and operationalize machine learning solutions

You need to run several autoML experiments and you want to keep your costs under control, as well as minimize the running times. ML Studio provides controls to achieve these goals.

Which two autoML controls should you use?

- A. Set exit criterion Training job time
- B. Set Max concurrent iterations to 4
- C. Set criterion Metric score threshold
- D. Set Primary metric to "AUC_weighted"

## Explanation:

**Answers: A and C**

- Option A is CORRECT because by using the Training job time exit criterion, you can limit the duration of the training runs.
- Option B is incorrect because setting the Max concurrent iterations higher 1 results in running multiple jobs in parallel. Without either limiting the running time or setting exit criterion it doesn't comply with the requirements.
- Option C is CORRECT because by setting the Metric score threshold exit criterion you can tell the pipelines to stop running as soon as the minimum value of the primary metric is reached.
- Option D is incorrect because the metric used for scoring the model has nothing to do with the running time or costs. It can be used in the exit criteria to set a threshold for it.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-use-automated-ml-for-ml-models#customize-featurization

## Question 21

Domain: Deploy and operationalize machine learning solutions

You are responsible for training and deploying a classification model which can identify fraud attempts among banking transactions. In order to minimize the time and effort, you decide to use Azure autoML services. While configuring autoML, you need to select a primary metric which can be used by autoML to select the best run.

Which metrics can you choose from?

- A. r2_score; spearman_correlation
- B. r2_score; normalized_mean_absolute_error
- C. normalized_mean_absolute_error; r2_score; accuracy
- D. AUC_weighted; norm_macro_recall; accuracy

## Explanation:

**Answer: D**

- Option A is incorrect because these are metrics for the Regression or Forecasting task types.
- Option B is incorrect because these are metrics for the Regression or Forecasting task types.
- Option C is incorrect because the list is a mix of metrics for the Regression or Forecasting task types.
- Option D is CORRECT because these are the metrics you can use to score a Classification model.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-train#primary-metric

## Question 22

Domain: Deploy and operationalize machine learning solutions

While you are experimenting with Azure autoML service, you need to configure the Tuning Hyperparameters feature. You want autoML to try running experiments varying the number_of_hidden_layers parameter of a neural network algorithm, as well as the batch_size.

Which is NOT a valid configuration for the autoML runs?

- A. Search space: normal; sampling: Bayesian; early termination: Yes
- B. Search space: quniform; sampling: random; early termination: BanditPolicy
- C. Search space: range; sampling: random; early termination: MedianStoppingPolicy
- D. Search space: choice; sampling: grid; early termination: None

## Explanation:

**Answer: A**

- Option A is CORRECT because the Bayesian sampling method can be used with choice, uniform and *quniform* space definition, but it cannot be combined with early termination option.
- Option B is incorrect because the search space can be defined by a *quniform* distribution; random sampling can be used to select parameter combinations; Bandit policy as a termination strategy can be a valid option.
- Option C is incorrect because the search space can be defined as a range of values; random sampling can be used to select parameter combinations; MedianStopping can be a valid option as a termination strategy. This is a valid option.
- Option D is incorrect because for discrete hyperparameters, the search space can be a choice of values, grid sampling can be used to try all possible combinations; if the number of combinations is not too high, the early termination might not be necessary, so this is a valid option.

**Reference:**

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

## Question 23

Domain: Deploy and operationalize machine learning solutions

You are running Azure autoML experiments to find the best performing regression model for your dataset. You want to use AML's hyperparameter tuning functionality. You select the Bayesian method for sampling the hyperparameters, and you also want to limit the duration of the run.

How do you configure the Early termination?

- A. early_termination_policy = TruncationSelectionPolicy
- B. early_termination_policy = MedianStoppingPolicy
- C. early_termination_policy = None
- D. early_termination_policy = BanditPolicy

## Explanation:

**Answer: C**

- Option A is incorrect because TruncationSelectionPolicy cannot be used together with Bayesian sampling.
- Option B is incorrect because MedianStoppingPolicy cannot be used together with Bayesian sampling.
- Option C is CORRECT because when you select Bayesian sampling, early termination option cannot be used, i.e. it has to be set to "None".
- Option D is incorrect because BanditPolicy is not applicable for Bayesian sampling.

**Reference:**

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

## Question 24

Domain: Deploy and operationalize machine learning solutions

You are going to train an ML model based on a neural network (NN) algorithm, using autoML. Based on your experience from previous experiments, the model's performance is expected to degrade after a number of iterations which can easily result in unnecessarily long execution times with no practical result. In order to save time, you decide to instruct autoML to stop iterations when the current run significantly underperforms the best of the previous runs.

How do you set this instruction in the Python code?

- A. early_termination_policy = MedianStoppingPolicy(...)
- B. early_termination_policy = None
- C. delay_evaluation = 5
- D. early_termination_policy = BanditPolicy(...)

## Explanation:

**Answer: D**

- Option A is incorrect because the Median stopping policy terminates the execution when the performance metric proves to be worse than the median of the running averages for the runs.
- Option B is incorrect because setting the termination policy to 'None' won't provide the expected result, i.e. the execution of the iterations will run, even if the model's performance degrades after a certain number of iterations.
- Option C is incorrect because delay_evaluation is a parameter of the termination policies. It instructs the selected termination rule to take effect only after the set delay value. Therefore, it is not a termination policy on its own.
- Option D is CORRECT because the Bandit policy setting for early termination has to be used if you want to terminate the iterations when the primary performance metric underperforms the best of the previous runs. All the other policies use different termination conditions.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters#bandit-policy

## Question 25

Domain: Deploy and operationalize machine learning solutions

You are building a classification model (logistic regression) which you want to optimize using the Azure ML hyperparameter tuning. For running Hyperdrive experiments, you have the following script:

```
...
sampling = GridParameterSampling(
    {
        '--regularization': choice(0.001, 0.01, 0.1, 1.0)
    }
)

hyperdrive = HyperDriveConfig(estimator=hyper_estimator,
                              hyperparameter_sampling=sampling,
                              policy=None,
                      [select the passing code segment here],
                              max_total_runs=6)
...
run = experiment.submit(config=hyperdrive)

...
```

The script is still missing some configuration details necessary for Hyperdrive.

Which code segments need to be added to the script?

- A. primary_metric_name='r2_score', primary_metric_goal=PrimaryMetricGoal.MINIMIZE
- B. primary_metric_name='AUC', primary_metric_goal=PrimaryMetricGoal.MAXIMIZE right
- C. primary_metric_name='AUC', max_concurrent_runs=4
- D. primary_metric_name='AUC', primary_metric_goal=PrimaryMetricGoal.MINIMIZE

## Explanation:

**Answer: B**

- Option A is incorrect because the 'r2_score' is used for regression models and it is not applicable for classification tasks; in addition, when applicable, it should be maximized in order to find the best performing run.
- Option B is CORRECT because the primary metric and the method of selecting the best performing run are two parameters which are needed for the Hyperdrive to complete its task.
- Option C is incorrect because the maximum number of concurrent runs is set to 'None' as default, therefore it is not mandatory; for selecting the best run, the primary metric goal must be set.
- Option D is incorrect because the best run is which has the highest value for the AUC metric. Therefore, setting the goal parameter to 'MINIMIZE' is incorrect.

**Reference:**

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

## Question 26

Domain: Deploy and operationalize machine learning solutions

You are using classification algorithms in AutoML to train a model to predict whether your customers are expected to take a loan or not. Your model has predictors as *marital status*, *job* and *education*. After running ML experiments, you want to find which predictor is most relevant in predicting the target variable.
Which action should you take?

- A. Select the feature with the highest local importance
- B. Enable auto-featurization
- C. Select the feature with the lowest global importance
- D. Select the feature with the highest global importance

## Explanation:

**Answer: D**

- Option A is incorrect because by examining the local importance of features can help understand how each feature contributes to the result of a specific prediction.
- Option B is incorrect because auto-featurization instructs AutoML to generate derived features based on the original features of the dataset. It has no effect on the model explainability.
- Option C is incorrect because while global feature importance can be used to understand the relative importance of features in the test dataset, in this case you should look for the feature with the *highest* global importance.
- Option D is CORRECT because global feature importance can be used to understand the relative importance of features. You should look for features with the highest global importance as the strongest contributors to the predictions.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability

## Question 27

Domain: Deploy and operationalize machine learning solutions

You have set up a machine learning workspace where you have completed a number of ML experiments on your dataset. Finally, the run showing the best performance has been selected. You are now in the phase of examining the test results in detail, to understand the contribution of the features to the predictions of the model.

Which combination of explainers would you NOT select for interpreting global and local feature importance?

- A. Permutation Feature Importance (PFI) for global; PFI for local
- B. Mimic for global; Mimic for local
- C. Tabular for global; Tabular for local
- D. Permutation Feature Importance (PFI) for global; Tabular for local

## Explanation:

**Answer: A**

- Option A is CORRECT because while Permutation Feature Importance (PFI) model explainer can only be used to explain how strongly the features contribute to the prediction at the dataset level, it doesn't support evaluation of local importances.
- Option B is incorrect because the Mimic Explainer can be used for interpreting both the global and local importance of features, so this option is valid.
- Option C is incorrect because the Tabular Explainer can be used for interpreting both the global and local importance of features, so this option is valid.
- Option D is incorrect because the PFI Explainer can be used for interpreting the global feature importance while the Tabular Explainer is a good choice for examining local importance, so this option is valid.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability

## Question 28

Domain: Deploy and operationalize machine learning solutions

You are running autoML experiments. Although during automated ML experiments several featurization techniques are applied automatically, in this particular case you want to customize featurization process and you want to manually select columns to be dropped.

Which of the following code segments will you need, in what order?

```
1.      automl_config = AutoMLConfig(name='Automated ML Experiment',
        …
        featurization='off'
       )
2.      featurization_config.drop_columns = ['aspiration', 'stroke']
3.      automl_config = AutoMLConfig(name='Automated ML Experiment',
        …
        featurization='FeaturizationConfig'
       )
4.      featurization_config.enabled _transformers = ['DropColumns']
5.      featurization_config = FeaturizationConfig()
```

- A. 1, 5, 2
- B. 3, 5, 2
- C. 5, 4, 2
- D. 3, 5, 4, 2

## Explanation:

**Answer: B**

- Option A is incorrect because setting featurization 'off' simply disables automatic featurization and doesn't allow customization. You have to set 'FeaturizationConfig'.
- Option B is CORRECT because to customize featurization, AutoMLConfig object must be created, with the featurization parameter set to 'FeaturizationObject';' then a FeaturizationConfig object must be created with setting the drop columns operation.
- Option C is incorrect because the AutoMLConfig object must be created with featurization='FeaturizationConfig'; you cannot "enable transformers" because they are executed by default and you can only disable them (setting enabled_transformers); there is no transformer 'DropColumns'.
- Option D is incorrect because you cannot "enable transformers" because they are executed by default and you can only disable them (setting enabled_transformers); there is no transformer 'DropColumns'. Therefore, the 'enabled_transformers' step is not needed.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features#featurization
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features#featurization

## Question 29

Domain: Deploy and operationalize machine learning solutions

You are running autoML experiments. Although during automated ML experiments several featurization techniques are applied by autoML automatically, in this particular case you want to customize featurization process and, instead of using the default solution, you want to manually configure how the missing values in the 'engine-size' column should be handled.

Which settings should you use in your code?

- A. AutoMLObject.featurization = 'FeaturizationConfig'; featurization_config.add_transformer_params('Imputer', ['engine-size'], {"strategy": "median"}) right
- B. AutoMLObject.featurization = 'FeaturizationConfig'; featurization_config.blocked_transformers('Imputer', ['engine-size'], {"strategy": "median"})

- C. AutoMLObject.featurization = off'; featurization_config.add_transformer_params('Imputer', ['engine-size'], {"strategy": "median"})

- D. AutoMLObject.featurization = 'FeaturizationConfig'; featurization_config.blocked_transformers('Imputer', ['engine-size'], {"strategy": "average"})

## Explanation:

**Answer: A**

- Option A is CORRECT because in order to customize autoML's featurization transformers, you have to use the add_transformers_params of the FeaturizationObject; setting featurization to 'FeaturizationConfig' is a precondition.
- Option B is incorrect because the add_transformer_params of the FeaturizationConfig object can be used for customization. The block_transformers is also a valid parameter but used for disabling some auto featurization algorithms.
- Option C is incorrect because autoML's featurization must be set to 'FeaturizationConfig' to enable customization of the automated featurization. That's why setting it to 'off' is incorrect.
- Option D is incorrect because the add_transformer_params of the FeaturizationConfig object can be used for customization. The block_transformers is also a valid parameter but used for disabling some auto featurization algorithms. Value "average" is the default imputer strategy.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-features#featurizatio

## Question 30

Domain: Deploy and operationalize machine learning solutions

You are training your ML model using autoML experiments, which, after ten iterations has come to a model which is able to predict target values with the required performance. You are about to deploy it to the inference environment. As the first step of the process, you need to register your model.

Which is the simplest way to register the best model?

- A. az ml model register -n sklearn_mnist --asset-path outputs/sklearn_mnist_model.pkl --experiment-name myexperiment --run-id myrunid --tag area=mnist

- B. model = run.register_model(model_name='sklearn_mnist', tags={'area': 'mnist'}, model_path='outputs/sklearn_mnist_model.pkl')

- C. from azureml.train.automl.run import AutoMLRun model = run.register_model(description = 'My AutoML Model', tags={'area': 'mnist'})
- D. from azureml.train.automl.run import AutoMLRun model = run.register_model(description = 'My AutoML Model', tags={'area': 'mnist'}, iteration = 5)

## Explanation:

**Answer: C**

- Option A is incorrect because this Azure CLI command is used with the azureml.core. Run object, while you are working in Azure autoML which means that the AutoMLRun must be used. In this case you have to select the best run manually, which is not the simplest solution.
- Option B is incorrect because this script is used with the azureml.core. Run object, while you are working in Azure autoML which means that the AutoMLRun must be used. In this case you have to select the best run manually.
- Option C is CORRECT because in the case of autoML support, the simplest way of registering the best model is to invoke the 'run' object from the AutoMLRun, without setting the iteration and metric parameters. The best run will be registered automatically.
- Option D is incorrect because in the case of autoML support, the simplest way of registering the best model is to invoke the 'run' object from the AutoMLRun. In this case the number of best iteration is 10, therefore setting iteration=5 will not give the best one.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=azcli#registermodel

## Question 31

Domain: Manage Azure resources for machine learning

While setting up your machine learning workspace, you want to register a blob container from your storage account, using the Python SDK.

Which two authentication modes can you use to connect to your storage?

```
from azureml.core import Workspace, Datastore
ws = Workspace.from_config()
# Register Datastore
blob_ds = Datastore.register_azure_blob_container(workspace=ws,
datastore_name='blob_data',

container_name='data_container',

account_name='az_store_acct',

<select code snippet here>)
```

- A. account_key='123456abcde789...'
- B. username='mytestuser', password='12345678'
- C. sas_token='123456abcde789...'
- D. Tenant_id ='23488jseko#j2', client_id='adjlaKLA2882'

## Explanation:

**Answers: A and C**

- Option A is CORRECT because in order to connect to a blob storage, you either need the account key or an SAS token (for temporary access).
- Option B is incorrect because username/password as the way of authentication is used with SQL datastores (Azure SQL, PostgreSQL etc.). It is not applicable for storage accounts.
- Option C is CORRECT because in order to connect to a blob storage, you either need the account key or an SAS token (for temporary access).
- Option D is incorrect because tenant_id and client_id of the service principal have to be used when registering an Azure Data Lake as a datastore.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-access-data

## Question 32

Domain: Manage Azure resources for machine learning

You are designing your ML work environment. Your data resides in an Azure storage account, in a blob storage container. You want to prevent unauthorized access to your source data and don't want to risk exposing access credentials.

Which options should you use to fulfil the above requirements?

- A. Describe the connection data in the training script
- B. Register the blob storage as a Datastore
- C. Register the blob storage as a Dataset
- D. Register the blob storage using an Estimator

## Explanation:

**Answer: B**

- Option A is incorrect because embedding any sensitive information (IDs, keys, tokens etc.) in the code must be avoided by all means.
- Option B is CORRECT because in the Azure ML environment, datastores are designed to store connection information like subscription IDs, access keys etc. By using datastores, all these information will be stored securely, and used via referencing the datastore which keeps the sensitive data hidden from scripts, applications etc.
- Option C is incorrect because Datasets are references to your data. They use the connection information stored in the datastore to access data from the location it actually resides. They are used in connection with datastores.
- Option D is incorrect because Estimator is a coding construct, it is an object that combines a run configuration and a script configuration in a single object for simpler use. They have nothing to do with accessing data, connection information etc.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-azure-machine-learning-architecture

## Question 33

Domain: Run experiments and train models

You are developing a ML pipeline using Python SDK, and you have to separate your data for training the model as well as for testing the trained model. You got a code snippet from your less experienced teammate, which is a great help, if it works. You have to check if it does the job.

By its description, the script loads data from the default datastore and separates the 70% of the observations for training and the rest of them for testing, by using the scikit-learn package, in a reproducible way.

```
from sklearn.model_selection import train_test_split

# Get the experiment run context
run = Run.get_context()

# load data
print("Loading Data...")
diabetes_data =
run.input_datasets['diabetes_train'].to_pandas_dataframe()

# Separate features and labels
X, y = diabetes_data[['Pregnancies','PlasmaGlucose',
        'DiastolicBloodPressure','BMI','Age']].values,
        diabetes['Diabetic'].values

# Split data into training set and test set
X_test, X_train, y_test, y_train =
        train_test_split(X, y, test_size=0.30, random_state=None)
```

After reviewing the code, do you think it does its job as described?

- A. Yes
- B. No

## Explanation:

**Answer: B**

- Option A is incorrect because there are two errors in the line of code which is responsible for splitting the data. The correct statement is:

```
# Split data into training set and test set
X_train, X_test, y_train, y_test =
        train_test_split(X, y, test_size=0.30, random_state=0)
```

- Option B is CORRECT because the 'train' and 'test' variables are swapped in the code, which would result in a 30/70 train/test rate between the two sets (instead of the required 70/30) In addition, the random_state is set to 'None' (which is its default), therefore the reproducibility of splitting cannot be ensured.

**Reference:**

- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html?highlight=split#sklearn.model_selection.train_test_split

## Question 34

Domain: Run experiments and train models

Your task is to build an ML pipeline for training a regression model to predict a car's price based on its technical features. Since you can't decide in advance which ML algorithm to use, you decide to train two regression algorithms (Boosted Decision Tree and Decision Forest) in parallel and compare their performance in the simplest way, so that execution requires the least amount of time. You are working with Azure ML Designer.

Which of the following Designer modules do you need to duplicate in the pipeline because of the comparison of two algorithms?

- Get data (Import, Dataset)
- Select Columns in Dataset
- Split Data
- Clean Missing Data
- Train Model
- Evaluate Model
- Score Model

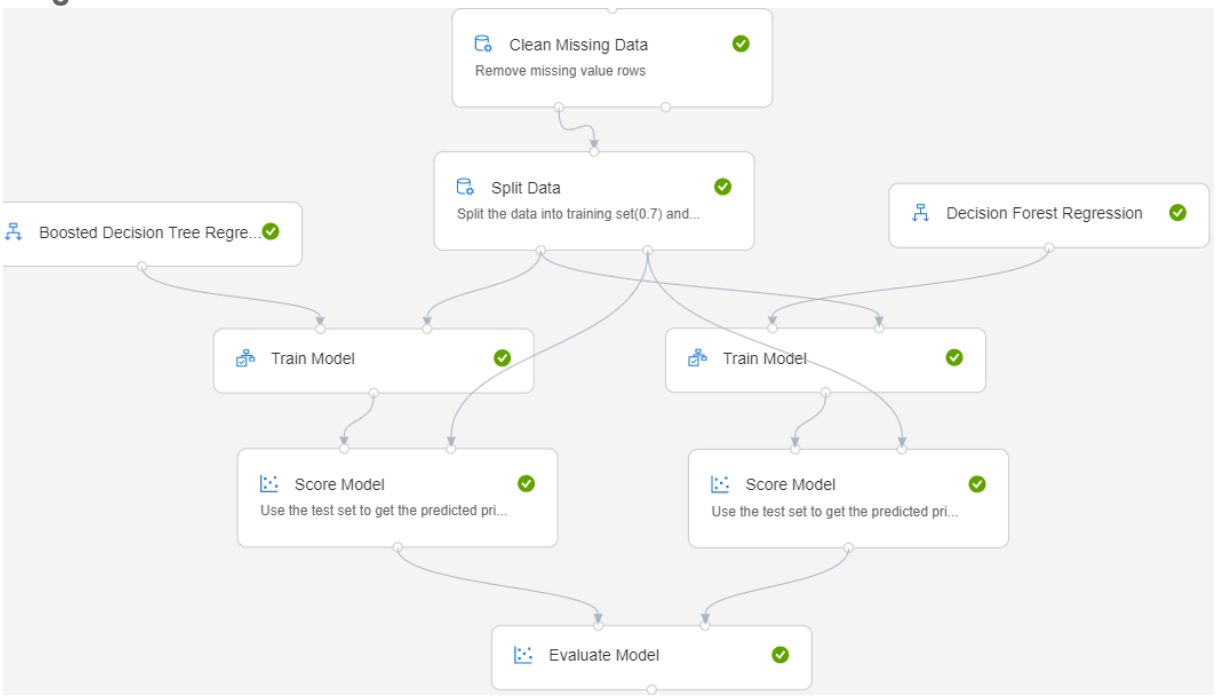
- A. Get Data, Select Columns in Dataset, Clean Missing Data, Train Model
- B. Split Data, Clean Missing Data, Score model, Evaluate Model
- C. Split Data, Train Model, Evaluate Model
- D. Train Model, Score Model

## Explanation:

**Answer: D**

- Option A is incorrect because getting the data and the data preparation process is the same, regardless of the number of algorithms used. Train Model is correct.
- Option B is incorrect because getting the data and the data preparation process is the same, regardless of the number of algorithms used. Score Model is correct.
- Option C is incorrect because splitting the data prepares the same datasets for both algorithms, therefore it doesn't need to be duplicated. Evaluate Model compares the performance metrics of the two algorithms. Train Model is correct.
- Option D is CORRECT because the data preparation process - up to splitting the data - is the same for the two algorithms. Only two steps need to be added for each of the algorithms: they need to be trained (Train Model) and scored (Score Model) separately. The Evaluation step compares the outputs of the two Scorings.

**Diagram:**




**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/samples-designer

## Question 35

Domain: Run experiments and train models

Your company is storing hundreds of GBs of data in a distributed Cosmos DB. This huge amount of data contains tons of valuable information about sales transactions and the company is going to make use of it by running machine learning models against it. Your task is to design how to feed Azure ML processes with Cosmos DB data. You decide to use Azure Data Factory for ingesting the data.

How do you configure ADF?

- A. Source: Blob Storage Container; Sink: Cosmos DB table; Integration runtime: Azure; Activity: Custom; Linked service type1: CosmosDb; Linked service type2: AzureBlobStorage; Copy activity source type: CosmosDbSqlApiSource

- B. Source: Cosmos DB table; Sink: Blob Storage Container; Integration runtime: Azure; Activity: Copy; Linked service type1: CosmosDb; Linked service type2: AzureBlobStorage; Copy activity source type: CosmosDbSqlApiSource
- C. Source: Cosmos DB table; Sink: Blob Storage Container; Integration runtime: Self-hosted; Activity: Copy; Linked service type1: AzureBlobStorage; Linked service type2: CosmosDb; Copy activity source type: CosmosDbSqlApiSource

- D. Source: Cosmos DB table; Sink: Blob Storage Container; Integration runtime: Azure-SSIS; Activity: Copy; Linked service type1: AzureBlobStorage; Linked service type2: CosmosDb

## Explanation:

**Answer: B**

- Option A is incorrect because you want to move data from Cosmos DB to Blob Storage, which means that your *source* is Cosmos DB and the *target/sink* is Blob Storage. In addition, Copy activity will do the task instead of Custom.
- Option B is CORRECT because your source of data is Cosmos DB, the target/sink is Blob Storage, the linked services are set accordingly, and - since the source is Cosmos DB, the source type in Copy Activity has to be CosmosDbSqlApiSource.
- Option C is incorrect because when you need to move data between Azure cloud data sources (i.e. Cosmos DB and Blob Storage), the Azure integration runtime must be used, i.e. setting 'Self hosted' is incorrect.
- Option D is incorrect because when you need to move data between Azure cloud data sources (i.e. Cosmos DB and Blob Storage), the Azure integration runtime must be used, i.e. setting 'Azure-SSIS' is not applicable here. Copy activity source is missing.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-data-ingest-adf
- https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db

## Question 36

Domain: Run experiments and train models

You are developing a ML pipeline using Python SDK, and you have to separate your data for training the model as well as for testing the trained model. You got a code snippet from your teammate, which is a great help, if it works. You have to check if it does the job.

By its description, the script loads data from the default datastore and separates the 70% of the observations for training and the rest of them for testing, by using the scikit-learn package, in a reproducible way.

```
from sklearn.model_selection import train_test_split

# Get the experiment run context
run = Run.get_context()

# load data
print("Loading Data...")
diabetes_data =
run.input_datasets['diabetes_train'].to_pandas_dataframe()

# Separate features and labels
X, y = diabetes_data[['Pregnancies','PlasmaGlucose',
        'DiastolicBloodPressure','BMI','Age']].values,
        diabetes_data['Diabetic'].values

# Split data into training set and test set
X_train, X_test, y_train, y_test =
        train_test_split(X, y, test_size=0.30, random_state=0)
```

After reviewing the code, do you think it does its job as described?

- A. Yes
- B. No

## Explanation:

**Answer: A**

- Option A is CORRECT because the code is correct. It does its job exactly as it is described.
- Option B is incorrect because the code is correct.

**Reference:**

- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html?highlight=split#sklearn.model_selection.train_test_split

## Question 37

Domain: Implement responsible machine learning

You have deployed your real-time inference web service on Azure Kubernetes Service (AKS). You need to ensure that only consumers after proper authentication can access the service.

In this particular case, which is not a correct way to set the authentication mode?

- A. Authentication mode: Key
- B. Authentication mode: Token; set token_auth_enabled = True
- C. Authentication mode: Token
- D. Authentication mode: Token; set auth_enabled=False

## Explanation:

**Answer: C**

- Option A is incorrect because by default, Key authentication is enabled when deploying on AKS. There is no need to explicitly enable it.
- Option B is incorrect because Token authentication is disabled by default when deploying to AKS. In order to use it, it must be enabled by setting oken_auth_enabled=True.
- Option C is CORRECT because by default, Token authentication is disabled by default when deploying to AKS. It cannot be used without enabling while creating or updating the deployment.
- Option D is incorrect because Token authentication is disabled by default when deploying to AKS. In order to use it, you have to use token_auth_enabled. auth_enabled must be disabled in this case.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-consume-web-service?tabs=python#authentication-for-services
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-setup-authentication#token-based-web-service-authentication
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-authenticate-web-service#token-based-authentication

## Question 38

Domain: Implement responsible machine learning

After successfully training your ML model, you have successfully deployed it as a real-time service to an AKS inference environment. During the live operation, you experience an error and your service crashes when you post data to the scoring endpoint.

Which is the best way to investigate the cause of the problem?

- A. In your PROD and DEV environment, add an error catching statement to your run() function so that it returns a detailed error message
- B. In your DEV environment, add an error catching statement to your run() function so that it returns a detailed error message
- C. In your PROD and DEV environment, add an error catching statement to your init() function so that it returns a detailed error message
- D. In your DEV environment, add error an catching statement to your init() function so that it returns a detailed error message

## Explanation:

**Answer: B**

- Option A is incorrect because including statements to return error messages from the run() function should only be used for debugging purposes. For security and performance reasons, this should be avoided in a production environment. Try debugging errors in a local container environment.
- Option B is CORRECT because including statements to return error messages from the run() function should only be used for debugging purposes. Debugging errors should be done in a local container environment, hence this is the correct answer.
- Option C is incorrect because inference errors can be caught within the run() function. init() is not the right place for that. In addition, for security reasons, returning detailed error messages should be avoided in production environments.
- Option D is incorrect because inference errors can be caught within the run() function. init() is not the right place for that.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment#function-fails-runinput_data

## Question 39

Domain: Implement responsible machine learning

You are working for a medical center where patients are being tested five days a week and their data, including the test results is collected in multiple CSV files. Data collected during the week should be fed into a ML model for classification, in order to determine which patients are at risk of COVID-19 infection. Your task is to implement this process using the SDK, using the following steps:

1. Upload CSV files and register them as a file dataset
2. Create ParallelRunStep object
3. Reference input dataset in the ParallelRunConfig object
4. Reference dataset in the ParallelRunStep object
5. Use ParallelRunStep object in a Pipeline
6. Upload CSV files and register them as a tabular dataset
7. Create ParallelRunConfig object

Which of the steps above should you include in what logical order?

- A. 1, 7, 2, 3, 5
- B. 1, 7, 2, 4, 5
- C. 6, 2, 7, 4, 5
- D. 1, 7, 2, 5

## Explanation:

**Answer: B**

- Option A is incorrect because reference to the dataset (input data) must be added to the ParallelRunStep object, not to the ParallelRunConfig.
- Option B is CORRECT because to process multiple data files for inference in batch mode, the ParallelRunStep can be used for working with data in parallel. Parameters of parallel processing can be set via the ParallelRunConfig object. While ParallelRunStep can either use Tabular or File dataset, in this case the File type is the right choice.
- Option C is incorrect because while working with a large number of data files, File dataset is the practical choice. In addition, creating the ParallelRunConfig must precede the creation of the ParallelRunStep.
- Option D is incorrect because reference to the dataset (input data) must be added to the ParallelRunStep object.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-pipeline-batch-scoring-classification
- https://github.com/MicrosoftLearning/DP100/blob/master/07B%20-%20Creating%20a%20Batch%20Inferencing%20Service.ipynb

## Question 40

Domain: Implement responsible machine learning

You have an Azure ML real-time inference model deployed to Azure Kubernetes Service. While running the model, clients sometimes experience a HTTP 503 (Service Unavailable) error. As a data engineer, you started to investigate the problem and you found that the error occurs when there are spikes in the number of requests.

Which two things can you do to prevent the problem?

- A. Increase the utilization level at which autoscaling creates new replicas.
- B. Set creating autoscale replicas faster
- C. Increase the minimum number of autoscaling replicas.
- D. Decrease the utilization level at which autoscaling creates new replicas.
- E. Increase the service's timeout

## Explanation:

**Answers: C and D**

- Option A is incorrect because the utilization level used to trigger creating new replicas is set to 70%, by default, meaning that the "buffer" to handle fluctuations is the remaining 30%. By increasing the limit, this margin narrows, further decreasing the resistance against peak demands.
- Option B is incorrect because creating new replicas is quick and responsive, with the time needed to create a new instance being around 1 second. There are no settings to control the speed of creation.
- Option C is CORRECT because setting the minimum number of autoscaling replicas will result in a larger space for handling sudden performance needs.
- Option D is CORRECT because the default setting for autoscale target utilization 70%. By decreasing it, the flexibility increases, i.e. the infrastructure can accommodate higher fluctuations without running out of capacity.
- Option E is incorrect because increasing the timeout (which is 1s, by default) might be a solution in the case of timeout (HTTP 504) error. It won't cure unavailability problems.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment#http-status-code-503

## Question 41

Domain: Implement responsible machine learning

You have an Azure ML real-time inference model deployed to Azure Kubernetes Service. While running the model, clients sometimes experience a HTTP 503 (Service Unavailable) error. As a data engineer, you have started to investigate the problem and you decide to set the autoscale_target_utilization parameter of your AksWebservice object in your code to 80.

Does it solve the problem?

- A. Yes
- B. No

## Explanation:

**Answer: B**

- Option A is incorrect because the utilization level used to trigger creating new replicas is set to 70%, by default, meaning that the "buffer" to handle fluctuations is the remaining 30%. By increasing the limit to 80, this margin narrows, further decreasing the resistance against peak demands, hence the answer is incorrect. So it does NOT solve the problem.
- Option B is CORRECT because the default setting for autoscale target utilization is 70%. By decreasing it, the flexibility increases, i.e. the infrastructure can accommodate higher fluctuations without running out of capacity. Therefore, this is the correct answer.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment#http-status-code-503

## Question 42

Domain: Deploy and operationalize machine learning solutions

You are using Azure's Auto ML functionality to train models on your dataset containing around 15 000 observations. In order to validate the models, Auto ML needs a dataset to compare the results of the predictions with. You decide to use 20% of your input data to validate the results.

You have the following configuration script which needs to be completed:

```
# configure Auto ML
my_data = Dataset.Tabular.from_delimited_files(data)

automl_config = AutoMLConfig(compute_target = aml_remote_compute,
                             task = 'classification',
                             primary_metric = 'AUC_weighted',
                             training_data = my_data,
                                  <insert code here,>
                             label_column_name = 'Class'
                            )
```

Which of the following options can be used to achieve this goal?

- A. The configuration is complete, no code needs to be added
- B. validation_data = validation_data,
- C. validation_data = validation_data,validation_size = 0.2,
- D. validation_size = 0.2,

## Explanation:

**Answer: D**

- Option A is incorrect because In the case no validation data is provided explicitly, auto ML applies default methods for validation, depending on the number of rows (observations) in the input dataset. If the dataset contains less than 20 000 rows, the cross-validation method is selected and used automatically, i.e. partitions of the original training data are used for cross-checking the performance of the runs. By default, it takes the 10% of the original data to use for validation. Since you want 20%, leaving the code as it is not the right option for you.
- Option B is incorrect because while setting the validation_data would be a valid option to define a second set to be used for validation, in this case your code should contain a statement for splitting the original data into training and validation sets. Since there is no such statement present, the option is incorrect.
- Option C is incorrect because in order to define the validation dataset, you can either define the training/validation split manually (by explicitly setting validation_data) or by giving only one dataset (training_data) and specifying the validation_size. These two ways cannot be mixed.
- Option D is CORRECT because you provided only one dataset for your experiments, which is training data. By setting the validation_size parameter to 0.2, you instruct Auto ML to keep 20% of the dataset for validation purposes.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-cross-validation-data-splits#default--data-splits-and-cross-validation

## Question 43

Domain: Run experiments and train models

Your team is tasked with building a ML solution to classify and label a large amount of images. The images are stored as files in an Azure blob storage and the model to be used is a pre-trained, ready-to-use neural network. You decide to make use of the ML pipelines which are powerful tools for building automated workflows. You jump into the high level design of a pipeline which fits your goal. The main building blocks you can choose from:

1. Register blob_storage as a Datastore
2. Register blob_storage as a Dataset
3. Register the model to your Workspace
4. Register the model to your Pipeline
5. Create the Pipeline
6. Register image files as a Datastore
7. Register image files as Dataset
8. Create Pipeline steps

Which of the above blocks should you use?

- A. 2, 4, 5, 7, 8
- B. 2, 3, 5, 7, 8
- C. 1, 3, 5, 7, 8
- D. 1, 3, 6, 7, 8

## Explanation:

**Answer: C**

- Option A is incorrect because the source container of data (the blob storage) must be registered as a Datastore, hence "Dataset" is incorrect in this context. In addition, the model has to be registered to the ML Workspace (not to the Pipeline).
- Option B is incorrect because the source container of data (the blob storage) must be registered as a Datastore, hence "Dataset" is incorrect in this context.
- Option C is CORRECT because in order to build the required ML pipeline, you need to register source storage as a datastore; register the image files as a Dataset; register your pre-trained model to your ML workspace; define the necessary Steps for the pipeline; and, finally, you have to build the Pipeline object.
- Option D is incorrect because the image files to be processed will serve as a Dataset for the pipeline, therefore trying to use them as a "Datastore" is incorrect.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-pipeline-batch-scoring-classification
- https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.builder.pipelinestep?view=azure-ml-py&preserve-view=true#remarks

## Question 44

Domain: Run experiments and train models

As part of a ML team, your task is to build a workflow to chain several steps of the ML process in order to automate the series of tasks and to allow automation and reuse. You are going to use the Pipeline feature of the Azure ML stack. You have planned the main steps in the pipeline, and now you are about to optimize how to organize your files around your pipeline.

Which two of the following options are true and should be used as best practice in pipeline design?

- A. Store scripts and dependencies of the whole pipeline in a single source directory in order to take the advantage of data reuse
- B. Store scripts and dependencies for each step in separate source directories in order to take advantage of data reuse
- C. Store scripts and dependencies of the whole pipeline in a single source directory in order to reduce the size of the snapshots for given steps
- D. Store scripts and dependencies for each step in separate source directories in order to reduce the size of snapshot for given steps
- E. Force output regeneration for steps in a run by setting the allow_reuse to False

## Explanation:

**Answers: B and D**

- Option A is incorrect because if all the scripts belonging to the pipeline are kept in a single directory, every time the content of the directory changes, it will force all steps to rerun, i.e. no data reuse.
- Option B is CORRECT because Steps in a pipeline can be configured to reuse results from their previous runs if the step's scripts, dependencies, inputs etc. are unchanged. Keeping the files for each step in separate folders ensures that if only files of any step changes, only the output data of the given step is regenerated, while all the others' remain unchanged and reusable.
- Option C is incorrect because if the scripts and dependencies of the whole pipeline are kept in a single directory, snapshotting of the steps takes an unnecessary high amount of time and storage.
- Option D is CORRECT because keeping the scripts and dependencies of each step in separate directories helps minimize the resources necessary for snapshotting the steps.
- Option E is incorrect because the default setting for allow_reuse is True which means that the results of the previous step run is reused until the content of the source_directory is unchanged. In order to save time and resources, changing this default behavior is only recommended if there is a special reason why the results must be re-generated.

**Reference:**

- https://docs.microsoft.com/en-us/python/api/azureml-pipeline-core/azureml.pipeline.core.builder.pipelinestep?view=azure-ml-py&preserve-view=true#remarks

## Question 45

Domain: Manage Azure resources for machine learning

When working with Azure ML services and tools, you have several options to select the execution environment (compute targets) for the experiments. Some compute targets are appropriate for development and testing at low cost, while others, being high-performance and scalable engines, can easily generate unexpected bills, if not used properly.

Which two of the following recommendations should you consider when selecting compute targets?

- A. Use Azure Container Instances for high-scale, real-time inferencing in PROD environment
- B. Use Azure Container Instances for low-scale, testing scenarios requiring <48 GB RAM
- C. Consider using local computes for low-scale, low-cost training tasks
- D. Use Azure Kubernetes Services for high-scale, real-time inferencing in PROD environment
- E. Use Azure Kubernetes Services for low-scale model training in TEST environment

## Explanation:

**Answers: B and D**

- Option A is incorrect because ACI is suitable for small models (<1GB in size), and the number of models also limited, and according to the recommendation, the RAM requirement should be under 48GB. For highscale, real-time inferencing AKS should be used.
- Option B is CORRECT because ACI is suitable for small models (<1GB in size), and the number of models also limited, and according to the recommendation, the RAM requirement should be under 48GB.
- Option C is incorrect because local computes are only recommended for low-cost debugging tasks; not recommended for training scenarios which typically require high compute capacity with autoscaling.
- Option D is CORRECT because AKS is the compute target specifically designed for heavy production workloads. With its sophisticated containerized runtime infrastructure, AKS provides the fast response and scalability of the deployed services.
- Option E is incorrect because AKS is the compute target specifically designed for heavy, high-scale production workloads. Due to its very high capacity and expensiveness, it is recommended for PROD targets only.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-compute-target#train

## Question 46

Domain: Implement responsible machine learning

After successfully training your ML model and after selecting the best run, you are about to deploy it as a web service to the production environment. Because you anticipate a massive amount of requests to be handled by the service, you choose AKS as a compute target. You want to use the following script to deploy your model:

```
# deploy model
from azureml.core.model import Model

service = Model.deploy(workspace=ws,
                       name = 'my-inference-service',
                       models = [classification_model],
                       inference_config = my_inference_config,
                       deployment_config = my_deploy_config,
                       deployment_target = my_production_cluster)
service.wait_for_deployment(show_output = True)
```

After running the deployment script, you receive an error. After a short investigation you find that an important setting is missing from the inference_config definition:

```
# inference config
from azureml.core.model import InferenceConfig

inference_config = InferenceConfig(runtime= "python",
                                   source_directory = 'my_files',
                                   <insert code here>,
                                   conda_file="environment.yml")
```

You decide to add <entry_script="my_scoring.py">

Does this solve the problem??

- A. Yes
- B. No

## Explanation:

**Answer: A**

- Option A is CORRECT because the InferenceConfig object is used to combine two important things: the entry script and the environment definition. The entry_script defines the path to the file that contains the ML code to execute, therefore it must be set.
- Option B is incorrect because the entry_script parameter is actually missing from the InferenceConfig definition. Adding it does solve the problem.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python

## Question 47

Domain: Implement responsible machine learning

After successfully training your ML model and after selecting the best run, you are about to deploy it as a web service to the production environment. Because you anticipate a massive amount of requests to be handled by the service, you choose AKS as a compute target. You want to use the following script to deploy your model:

```
# deploy model
from azureml.core.model import Model

service = Model.deploy(workspace=ws,
                       name = 'my-inference-service',
                       models = [classification_model],
                       inference_config = my_inference_config,
                       deployment_config = my_deploy_config,
                       deployment_target = my_production_cluster)
service.wait_for_deployment(show_output = True)
```

After running the deployment script, you receive an error. After a short investigation you find that an important setting is missing from the inference_config definition:

```
# inference config
from azureml.core.model import InferenceConfig

inference_config = InferenceConfig(runtime= "python",
                                   source_directory = 'my_files',
                                   <insert code here>,
                                   conda_file="environment.yml")
```

You decide to add <cluster_name = 'aks-cluster'>

Does this solve the problem??

- A. Yes
- B. No

## Explanation:

**Answer: B**

- Option A is incorrect because the InferenceConfig object is used to combine two important things: the entry script and the environment definition. The entry_script defines the path to the file that contains the ML code to execute, therefore it is missing and it must be set: entry_script="my_scoring.py".
- Option B is CORRECT because the cluster_name parameter is actually important for the deployment, but it is part of the ComputeTarget configuration (which is, in your case, the my_production_cluster), i.e. set elsewhere in your code.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python

## Question 48

Domain: Run experiments and train models

You have just completed several runs of your ML experiment in Azure ML Studio. You have run your multiclass classification experiments, trying several algorithms. In order to determine which model gives the best result, you start evaluating the runs using the graphical tools provided by Studio. First, you want to eliminate the models with the weakest performance.

By looking at the confusion matrices, which model should you keep?

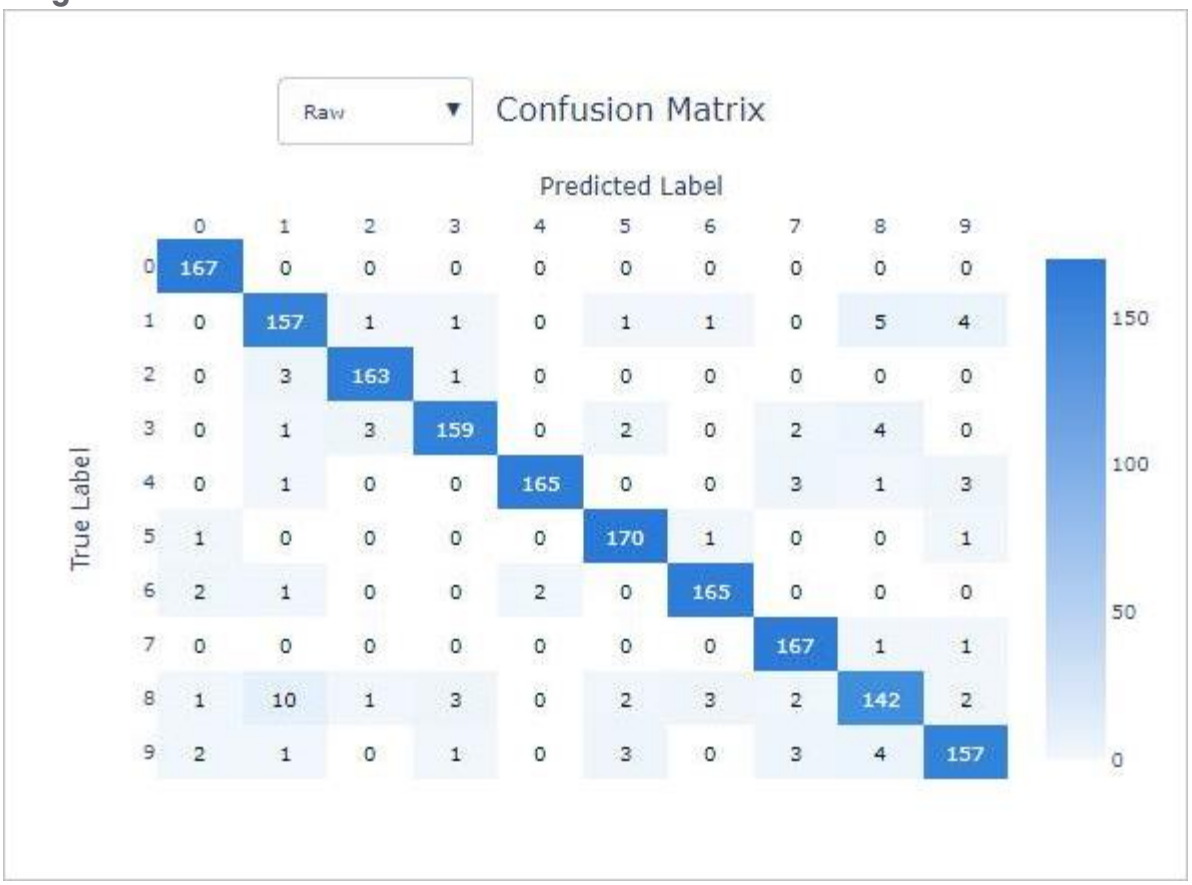
- A. Where the non-zero values are concentrated in columns
- B. Where the non-zero values are concentrated in rows
- C. Where the zero values are found in the diagonal from top left
- D. Where the non-zero values are in the diagonal from top left

## Explanation:

**Answer: D**

- Option A is incorrect because for a high-accuracy model, the values in the confusion matrix are expected to center in the diagonal. If most of the numbers are off-diagonal, then it is a sign of a weak model.
- Option B is incorrect because for a high-accuracy model, the values in the confusion matrix are expected to center in the diagonal. If most of the numbers are off-diagonal, then it is a sign of a weak model.
- Option C is incorrect because for a high-accuracy model, the values in the confusion matrix are expected to center in the diagonal. Zeros in the diagonal (with non-zeros out of diagonal) are signs of low model performance.
- Option D is CORRECT because the confusion matrix visualizes the number of predicted labels compared to number of actual labels, i.e. showing the model's accuracy. With the actual labels on the Y axis and with the predictions on the X, for a high accuracy model the non-zero values must concentrate around the diagonal from top left.

**Diagram:**




**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-understand-automated-ml

## Question 49

**Domain:** Run experiments and train models

You have just completed an ML experiment in Azure ML. You have trained models with several regression algorithms, which is to be used to predict effectiveness of some newly developed medicine.

Which two evaluation tools/metrics would help you decide how powerful your model is?

- A. Normalized root mean squared error (RMSE)
- B. Predicted vs. True chart
- C. ROC Chart
- D. Recall
- E. AUC

## Explanation:

**Answers: A and B**

- Option A is CORRECT because the root mean squared error is a single value that summarizes the errors in the model. Its normalized version (RMSE divided by the range of the data) is one of the metrics typically used for regression problems . The closer its value to 0.0 the better.
- Option B is CORRECT because one of the visualizations Azure ML provides for evaluating regression models is the Predicted vs. True diagram. It shows the relationship between a predicted value and its correlating true value. It indicates good model performance if the predicted values are close to the y=x line.
- Option C is incorrect because the ROC (Receiver Operating Characteristic) curve displays the correctly classified labels vs. the incorrectly classified ones. It actually indicates how "strong" the model is but it is only aApplicable for classification problems.
- Option D is incorrect because Recall is a metric expressing the percent of correctly labeled elements of a certain class (the percent of the total amount of relevant instances that were actually found). Applicable only for classification problems, not for regression.
- Option E is incorrect because AUC (Area Under Curve) is a metric used for classification scenarios. It shows the relationship between the true positives and the false positives, in a graphical form. This is a good visual metric but not for regression models.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-understand-automated-ml

## Question 50

Domain: Deploy and operationalize machine learning solutions

You have to train a ML model on your dataset consisting of a large number of columns. Based on your experience, you anticipate long and expensive training runs. In order to improve the time- and cost-efficiency of your work, you want to decrease the amount of input data by removing columns of little relevance. ML Designer offers several modules to use in your pipeline:

1. Select Columns Transform
2. Filter Based Feature Selection
3. Apply Transformation
4. Permutation Feature Importance (PFI)

Which designer modules should you include, in what order?

- A. 1, 2, 3
- B. 2, 1, 3
- C. 1, 2, 4
- D. 4, 1, 3

## Explanation:

**Answer: B**

- Option A is incorrect because while you actually need these three modules, Filter Based Feature Selection must be the first in the sequence because it holds the logic to calculate the relevance of features, i.e. the sequence is incorrect.
- Option B is CORRECT because in order to filter out irrelevant columns from your dataset before the model is created, you need to use filter based feature selection by choosing a statistical measure which is calculated for each feature and used to determine their relevance. You will then use only the columns with the best scores for the best efficiency. You then deed to add Select Columns Transform to generate a dynamic set of columns, then Apply Transformation.
- Option C is incorrect because Filter Based Feature Selection and PFI cannot be mixed this way. Not the right modules selected.
- Option D is incorrect because If you want to filter out columns from your dataset before the model is created, you need to use filter based feature selection. Permutation Feature Importance can be used to generate a set of feature scores after the model has been trained, to calculate feature importance afterwards. PFI uses a trained model and a dataset as inputs, while FPFS uses a dataset (typically train data) as input. Hence, using PFI is incorrect.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/filter-based-feature-selection?WT.mc_id=docs-article-lazzeri

## Question 51

Domain: Implement responsible machine learning

You have deployed your real-time inference web service on the Azure Container Instances (ACI) environment. You need to ensure that only consumer services having the appropriate authentication credentials can have access to it.

Which authentication method can you use?

- A. User/Password
- B. Key
- C. SAS
- D. Token

## Explanation:

**Answer: B**

- Option A is incorrect because Azure ML provides two ways to control access to web services: Kea and Token. User/Password is not applicable here.
- Option B is CORRECT because using a Key is the only way to authenticate consumers of a real-time inference service on ACI. The Key has to be included in the Authorization header of the request.
- Option C is incorrect because the SQL engine
- Option D is incorrect because while using time limited tokens for authentication can be an option for inference models, it is not supported for ACI; it is only available for AKS.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-consume-web-service?tabs=python#authentication-for-services

## Question 52

Domain: Implement responsible machine learning

You have a real-time inference web service which you have just deployed to Azure Kubernetes Service. During its run, some unexpected errors occur. You need to troubleshoot it quickly and cost-effectively.

Which is the quickest and cheapest option you should use?

- A. Deploy it as a local web service and debug locally
- B. Deploy it to ACI
- C. Use a compute instance as deployment target for debugging
- D. Deploy it to AKS and set the maximum number of replicas to one; debug it in the production environment

## Explanation:

**Answer: A**

- Option A is CORRECT because using a local web service makes it easier to troubleshoot and debug problems. If you have problems with your model deployed to ACI or AKS, try deploying it as a local web service. You can then troubleshoot runtime issues by making changes to the scoring file that is referenced in the inference configuration, and reloading the service without redeploying it. This can be done only with local services.
- Option B is incorrect because Azure Container Instances is designed to be used for low-scale production deployments. It is highly recommended to debug locally before deploying the web service.
- Option C is incorrect because while an Azure compute instance might be a good target platform for debugging, using a local containerized environment is even better in this case.
- Option D is incorrect because Azure Kubernetes Service is to be used for high-scale production deployments. Being a powerful runtime environment, it is far too expensive for testing and debugging.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment#debug-locally

## Question 53

Domain: Deploy and operationalize machine learning solutions

You are using Azure's Auto ML functionality to train models on your dataset containing around 15 000 observations. The child runs need to validate the model by comparing the predictions made by model with labels in the validation data. Therefore, the Auto ML needs to be provided with both training and validation data. You provided the necessary training data, but no data for validation has been given.

What do you expect to happen?

- A. Train/validation split is applied automatically
- B. A "Missing validation data" exception is thrown and the execution stops
- C. Cross-validation is applied automatically
- D. An error message is written to the log of the Run which can be retrieved by RunDetails().show()

## Explanation:

**Answer: C**

- Option A is incorrect because Auto ML applies default methods for validation in case no validation data is provided explicitly. The method to be applied depends on the number of rows (observations) in the input dataset. The Train/validation split method is used automatically if the dataset contains more than 20 000 rows. Since your dataset has less than 20 000 rows, this option is incorrect.
- Option B is incorrect because Auto ML applies default methods for validation in case no validation data is provided explicitly. No exception occurs simply by the lack of explicit validation data.
- Option C is CORRECT because Auto ML applies default methods for validation in case no validation data is provided explicitly. The method to be applied depends on the number of rows (observations) in the training dataset. If the dataset contains less than 20 000 rows, cross-validation method with the default number of folds (depending on the number of rows) is selected and used automatically, i.e. parts of the original training data are used for cross-checking the performance of the runs.
- Option D is incorrect because the log of the Run object is primarily used to log metrics during experiment runs. In addition, in this case no error occurs, which means there is no errors to log.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-cross-validation-data-splits#default--data-splits-and-cross-validation

## Question 54

Domain: Deploy and operationalize machine learning solutions

You are working for a company which is operating a webshop. All the transactions flowing through the site are directed to a real-time inferencing web service to identify potentially risky transactions. One of the transactions is classified by the model as "suspicious" and, before taking actions, you are tasked to investigate which features made the model "think" so.

You decide to use Mimic Explainer to help you understand why this specific transaction has been classified as "suspicious".

Does it serve your purpose?

- A. Yes
- B. No

## Explanation:

**Answer: A**

- Option A is CORRECT because Azure offers a selection of model explainers: Tabular, Mimic and Permutation Feature Importance. All of them can be used for explaining global importance of features, but only two of them (Tabular, Mimic) are applicable if you need to interpret local importance. Mimic is a good choice for your task.
- Option B is incorrect because either Mimic or Tabular explainer can be used for interpreting the local importance of features, i.e. Mimic is a good choice.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability

## Question 55

Domain: Implement responsible machine learning

You have created a ML pipeline which accepts daily transaction data to make predictions for the forthcoming period. The experience shows that patterns in the data slightly change from time to time, that's why you decide to re-train the model weekly, which means that, from week to week, you have to feed the pipeline with new training data. You want to implement a solution which fulfils this requirement with the least possible effort, while ensuring operation of the inference pipeline.

How can you achieve this goal?

- A. In the pipeline definition, create a PipelineParameter; publish the modified pipeline; pass the actual parameter value in the REST call as JSON document when new training data is available.
- B. In the pipeline definition, create a PipelineData; re-publish the pipeline; pass the actual parameter value in the REST call as JSON document when new training data is available.
- C. Using your pipeline and the latest data, you run weekly training sessions in your training environment and re-deploy the modified inference pipeline weekly.
- D. In a PythonScriptStep definition, create a PipelineParameter; publish the modified pipeline; pass the actual parameter value in the REST call as JSON document when new training data is available.

## Explanation:

**Answer: A**

- Option A is CORRECT because inference pipelines can be configured to accept parameters which can lend them some dynamic behavior. One typical use can be feeding different sets of data by passing the source of the data as a parameter. In Azure ML, the PipelineParameter object is designed for this purpose.
- Option B is incorrect because inference pipelines actually can be configured to accept parameters, for example to be run with different sets of data. In Azure ML, using the PipelineParameter is the way of achieving this. PipelineData is a valid parameter, but it is designed to pass data between pipeline steps.
- Option C is incorrect because offline re-training the pipeline and re-publishing it from week to week requires a lot more effort than configuring and deploying the pipeline once and using it in dynamic mode, using parameters.
- Option D is incorrect because while PipelineParameter is the way of dynamically setting dataset for a pipeline, it must be defined at pipeline level, not at step level.

**Reference:**

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-pipelines#changing-datasets-and-datapaths-without-retraining
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-pipelines