Question 1

Domain: Run experiments and train models

You are building a ML pipeline using the ML Designer. Your pipeline consists of the following major modules: Get data, Split data, Clean Missing Data, Train Model, Score Model, Evaluate Model. You want to run your pipeline on an ML Compute Cluster, but for the Train Model step you want to use the massive Databricks Cluster you set up earlier for another project. You want to solve it in the most cost-effective way.

Which option should you follow?

- A. Set the pipeline's default compute target to ML Compute Cluster. Databricks cannot be used with pipelines.
- B. Set Databricks Cluster as the default compute target for the pipeline. Run the pipeline on Databricks.
- C. Set the default compute target for the pipeline to ML Compute Cluster and set the Databricks Cluster for the training step.
- D. Create two pipelines, one for running the Compute Cluster for the preparatory steps, and another one for the training/scoring tasks on the Databricks.

Explanation:

Answer: C

- Option A is incorrect because a default compute target can be set at pipeline level and you can set different targets for the steps individually. Databricks Cluster is a valid option.
- Option B is incorrect because while you can use the Databricks Cluster for the whole pipeline, it is not the most cost-effective way. You can use different targets for different steps.
- Option C is CORRECT because the best way to solve the task is setting the low-cost compute target at pipeline level as default, and changing the target for only steps that require so. Databricks Cluster is a valid option.
- Option D is incorrect because this is not a way of building pipelines. The problem can be solved within a single pipeline, by setting the compute target for steps individually.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-designer-automobile-price-train-score
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-attach-compute-targets#databricks

Question 2

Domain: Implement responsible machine learning

You have finished with training your ML model: it is optimal, hyperparameters are tuned, everything is fine. You are about to deploy it as a real-time inferencing service to an Azure Kubernetes cluster. In order to have your service in production, there is a list of activities you have to execute.

- Connect to your workspace
- Register the model
- Prepare inference configuration
- Prepare entry script
- Choose a compute target
- Define a deployment configuration
- Deploy the model

Which of the above activities are recommended but not required?

- A. Register the model
- B. Prepare entry script; Register the model
- C. Prepare inference configuration; Chose a compute target
- D. Prepare entry script

Explanation:

Answer: A

- Option A is CORRECT because in case you proceed without registering a model, you need to manually specify a source directory in your InferenceConfig and ensure that model is in that directory. Model registration can make model management easier. Therefore, registering a model is recommended but not required.
- Option B is incorrect because although registering the model is only recommended, an entry script must be prepared because it receives the data submitted to a deployed web service and passes it to the model, it is the interface between the client and your service.
- Option C is incorrect because both of these activities are required for a successful model deployment. Inference configuration describes how to set up the web-service, and a compute target to host the service must be chosen, too.
- Option D is incorrect because an entry script is a mandatory component of the model deployment. An entry script must be prepared because it receives the data submitted to a deployed web service and passes it to the model, it is the interface between the client and your service.

Reference:

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

Question 3

Domain: Manage Azure resources for machine learning

You have created a ML workspace from SDK using the following script:

```
# create workspace
from azureml.core import Workspace

ws = Workspace.create(name='myworkspace',
               subscription_id='<azure-subscription-id>',
               resource_group='myresourcegroup',
               create_resource_group=True,
               location='eastus2'
               )

ws.write_config()
...
```

For your machine learning experiments, you have several scripts, from which you need to connect to the workspace by the simplest and most flexible way.

```
...
# connect to workspace
from azureml.core import Workspace

<insert code here>
```

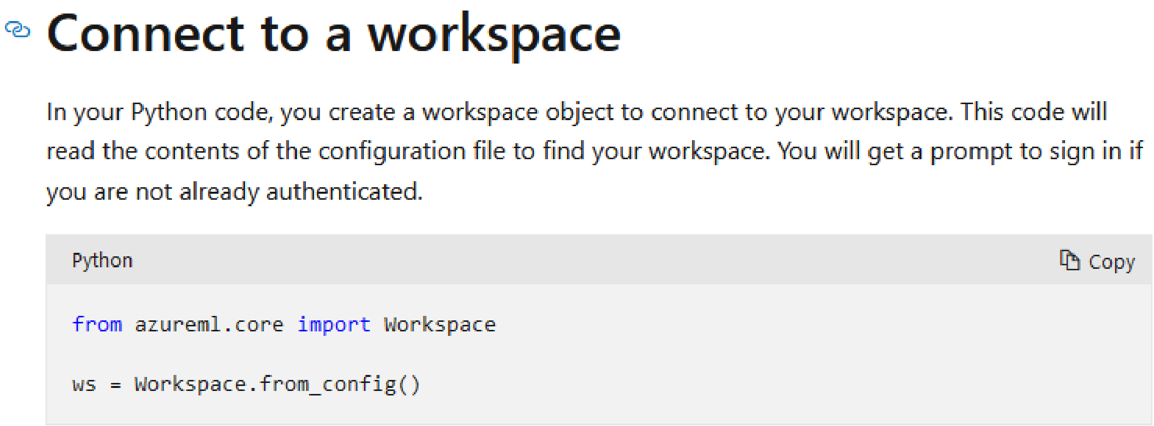
Which code segment fits best to the purpose?

- A. ws=Workspace.read_config()
- B. ws=Workspace.get()
- C. ws=get_connection()
- D. ws=Workspace.from_config()

Explanation:

Answer: D

- Option A is incorrect because there is no such method for the Workspace class. The from_config() is designed for this purpose.
- Option B is incorrect because the get() method can actually be used to connect to a workspace but this is an alternative to using a config file.It needs the workspace details (subscription, resource group etc.) to be explicitly defined. Using the config.json file to store workspace details is the recommended and more flexible technique.
- Option C is incorrect because this method is used to return a connection under a workspace, not for connecting to a workspace.
- Option D is CORRECT because this is the from_config() method which is used to connect to a workspace using the configuration file that was saved before by the write_config(). This is the simplest way of reconnecting to a workspace from several environments, without specifying its details each time.



Connect to a workspace

In your Python code, you create a workspace object to connect to your workspace. This code will read the contents of the configuration file to find your workspace. You will get a prompt to sign in if you are not already authenticated.

```python
from azureml.core import Workspace

ws = Workspace.from_config()
```

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-manage-workspace?tabs=python#create-a-workspace
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-manage-workspace?tabs=python#download-a-configuration-file

Question 4

Domain: Deploy and operationalize machine learning solutions

Your company is operating a home delivery service which requires management of a large fleet of vehicles. Because of the COVID-related restrictions, demand for your services has significantly increased. In order to serve this demand with the existing fleet, the company needs to optimize the fleet operation. You, as a data scientist, are tasked to generate demand forecasts for the next six months. You decide to use forecasting in the AutoML service.

Which item of the following list is not required when running the forecasting models?

- A. Data must contain a time feature for each observation
- B. Provide validation data for the training
- C. Data must be sorted in ascending order of the time feature
- D. Set the Forecast horizon
- E. Set the Name of the time column

Explanation:

Answer: B

- Option A is incorrect because the most important for time-series forecasting tasks is that training data contains a feature that represents a valid and consistent time series feature having observation data at each point.
- Option B is CORRECT because time series forecasting uses Rolling Origin Cross Validation (ROCV) for validating data. For the ROCV method the training and validation data must be passed together, and the number of cross validation folds must be set. No separate validation data must be provided.
- Option C is incorrect because for time-series forecasting tasks the training dataset must be set in ascending order based on the time feature, so that it represents a valid time series.
- Option D is incorrect because it has to be defined for the model how many periods forward you want to forecast. The horizon must be in units of the time series frequency (days, weeks, months etc.). It is required and its default value is 1.
- Option E is incorrect because the name of the time column is required to specify the datetime column in the input data, which is used for building the time series.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-auto-train-forecast

Question 5

Domain: Manage Azure resources for machine learning

You have just set up your ML workspace with several computes. For your machine learning experiments, you want to use Python SDK and the Jupyter notebook environment. You'll have a number of files (scripts, notebooks, data, temporary files etc.) which need to be organized and stored to ensure the best performance and to be accessible for computes.

How should you organize storing your files?

- A. Create a storage account in your workspace and store both the scripts and data in there.

- B. Store scripts and notebooks on local disks of compute instances; store data in datastores.

- C. Store scripts and notebooks in the default storage account of your workspace; store your data in datastores.
- D. Store scripts and notebooks in the default storage account of your workspace; store your large datafiles in the computes' local \tmp folder.

Explanation:

Answer: C

- Option A is incorrect because each workspace has a default storage account attached to it on creation. This is the place where all the scripts and notebooks are stored by default, and it is shared among all computes in the workspace. There is no need to create one manually. Data should be stored in ML datastores.
- Option B is incorrect because while you can store scripts and related files on the local disks of the computes, they won't be accessible for other computes.
- Option C is CORRECT because each ML workspace has a default storage account attached to it on creation. The file share in this account is mounted to each compute within the workspace. Files stored here can easily be shared among computes. For data, it is not recommended to use ML datastores which are specifically designed to store large data files consumed by ML experiments.
- Option D is incorrect because for large data files consumed by training experiments, the recommended practice is storing them in ML datastores. Avoid storing them on the local disks of computes.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-compute-instance#accessing-files

Question 6

Domain: Deploy and operationalize machine learning solutions

You are running experiments with Azure autoML service and you need to configure the model's hyperparameters. You need to define the sampling space for two parameters and you want AutoML to use a sampling method which picks samples based on the result of previous runs, in order to improve the performance in the primary metric.

```
# define parameter space
from azureml.train.hyperdrive import BayesianParameterSampling, normal,
uniform

my_parameter_space = {
        "learning_rate": normal(10, 2),
        "keep_probability": uniform(0.05, 0.1)
    }

param_sampling = BayesianParameterSampling(my_parameter_space)
...
```

Does the script above fulfills your requirement?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because the code defines the parameter search space for two parameters, learning_rate and keep_probability, with the Bayesian sampling method. Bayesian sampling improves sampling based on the result of previous runs, but it cannot be used with 'normal' distribution.
- Option B is CORRECT because while the Bayesian parameter sampling is the right choice, this sampling method only supports choice, uniform, and quniform distributions. Therefore, using 'normal' is wrong in this case.

Reference:

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

Question 7

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 investigating the problem and you decide to set the autoscale_target_utilization parameter of your AksWebservice object in your code to 60.

Does it solve the problem?

- A. Yes
- B. No

Explanation:

Answer: A

Option A is CORRECT because the default setting for autoscale target utilization is 70%. By decreasing it to 60, the flexibility increases, i.e. the infrastructure can accommodate higher fluctuations without running out of capacity. Therefore, this is the correct answer.

Option B 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, the margin narrows, further decreasing the resistance against peak demands, hence the answer is incorrect. So it does NOT solve the problem.

Reference:

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

Question 8

Domain: Manage Azure resources for machine learning

You are about setting up a machine learning environment. You already have a workspace where you need to configure the compute resources for your experiments. You are going to make use of the capabilities of Azure's AutoML feature and you want to use ML pipelines to organize your workflow, for which you want to use the ML Designer.

Which compute resource should you choose?

- A. Azure ML compute instance
- B. Azure HDInsights
- C. Remote VM
- D. Azure ML compute cluster

Explanation:

Answer: D

- Option A is incorrect because the ML compute instance is good for both AutoML and for training run of pipelines, it is not suitable for ML Designer.
- Option B is incorrect because while HDInsights is capable of running pipelines, it is not suitable for AutoML and for ML Designer.
- Option C is incorrect because remote VMs cannot be used together with ML Designer.
- Option D is CORRECT because Azure ML compute cluster is the only option suitable for AutoML, for running pipelines as well as to exploit the capabilities of the graphical ML Designer.

Reference:

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

Question 9

Domain: Run experiments and train models

Your company is operating a fleet of IoT devices used to collect several environmental parameters at many locations. They produce a huge amount of data but, for some reasons, the incoming data is regularly “contaminated” with missing values in different numerical columns. Your task is to clean data from missing values by using a predefined transformation in the ML Designer.

What is the recommended practice to achieve the goal?

- A. Drop a saved transformation as a module from Transforms list; connect it to the Apply Transformations module.
- B. In the Clean Missing Data module; set the Custom substitution value to the saved cleaning transformation

- C. In the Clean Missing Data module; set the Custom substitution value to the saved cleaning transformation; select the columns to be cleaned

- D. Drop a saved transformation from Transforms list; connect it to the Apply Transformations module; select the columns to be cleaned.

Explanation:

Answer: A

- Option A is CORRECT because the most convenient way is writing cleansing rules once and using them many times in Designer pipelines. Clean missing value rules can be defined and saved by the Clean Missing Data module, then can be re-used by the Apply Transformations module. Saved transformations can be applied for datasets with the same schema. Saved transformations appear in the Designer as drop modules.
- Option B is incorrect because the Clean Missing Data module is used to define a cleansing transformation (e.g. by setting a custom substitution rule), which then can be saved for future re-use. This module cannot use saved transformations.
- Option C is incorrect because you cannot select the columns to which the transformation to be applied when using a saved transformation. Transformation applies exactly for the columns defined earlier.
- Option D is incorrect because you cannot select the columns to which the transformation to be applied when using a saved transformation. Transformation applies exactly for the columns defined earlier.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/clean-missing-data#apply-a-saved-cleaning-operation-to-new-data

Question 10

Domain: Deploy and operationalize machine learning solutions

You are working for an insurance company which has recently introduced a ML model to predict the risk level of new customers. During the training phase, the model showed a very high accuracy of 99.5% and its test accuracy was above 96%. However, in production use, for real cases the overall accuracy is around 60% which is rather low. You know that Azure AutoML service can help you solve the problem.

Which are the best practices you should follow?

- A. The model is probably overfitting; increase the number of features in your dataset; use AutoML's regularization to control overfitting.

- B. The model is probably underfitting; decrease the number of features in the training dataset; use cross-validation.

- C. The model is probably overfitting; decrease the number of features in your dataset; use cross-validation while training.
- D. The model is probably underfitting; increase the number of features in your dataset; use cross-validation.

Explanation:

Answer: C

- Option A is incorrect because one of the possible reasons for a model's overfitting is that there are too many features in the training dataset. In this case, adding further features to the dataset doesn't solve the problem.
- Option B is incorrect because performance of an underfitting model on the training data is poor, which means that the model is not able to grab the actual relationships between variables. Your model's performance on training data is excellent, so this is not a case of underfitting.
- Option C is CORRECT because if there are too many features in the training dataset, the model tends to "memorize" some patterns which are not necessarily valid for real-life cases. By decreasing the number of features you can decrease the model's complexity. This, together with AutoML's cross-validation functionality might help prevent overfitting.
- Option D is incorrect because performance of an underfitting model on the training data is poor, which means that the model is not able to grab the relationships among variables. Your model's performance on training data is excellent, so this is not a case of underfitting.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-manage-ml-pitfalls#best-practices-you-implement

Question 11

Domain: Implement responsible machine learning

You are deploying your trained model as a web service to ACI compute in order to expose it as a REST API accessible for HTTP requests. One of your teammates has written the following code. You are reviewing it and you find that a line of code is missing.

```
# code sample
deployment_config = AciWebservice.deploy_configuration(cpu_cores = 1,
memory_gb = 1)
...
service = Model.deploy(ws, service_name, [model], inference_config,
deployment_config)
...
<insert missing line here>
...
predictions = requests.post(endpoint, input_json, headers = headers)
```

Which of the following code segments must be added to the script?

- A. endpoint = service.service_name
- B. endpoint = service.scoring_uri
- C. endpoint = service.service_endpoint
- D. endpoint = service.models

Explanation:

Answer: B

- Option A is incorrect because the name of the service must be defined during the creation (deployment) of the service. This is a string, not a URL.
- Option B is CORRECT because in order to access a deployed web service via HTTP, you need the name of the endpoint, i.e. the URL of the service. This can be queried from the *scoring_uri* parameter of the service object.
- Option C is incorrect because there is no such attribute in the AciWebservice object. It is the attribute *scoring_uri* that holds the URL for the REST calls.
- Option D is incorrect because it lists the models deployed to the Webservice, which cannot be used for REST requests.

Reference:

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

Question 12

Domain: Run experiments and train models

For your classification task, you have a large but imbalanced dataset with class A and class B as labels. Since in your database occurrences of B are relatively low, in order to have more accurate predictions, you decide to try doubling the percentage of the under-represented class. You decide to try the SMOTE (Synthetic Minority Oversampling Technique) module available in the ML Designer, and you also want reproducible results.

Which settings should you use to get the expected result?

- A. Set SMOTE percentage = 0; set Random seed = 0

- B. Set SMOTE percentage = 200; set Random seed = 1
- C. Set SMOTE percentage = 200; leave Random seed empty

- D. Set SMOTE percentage = 100; set Random seed = 0

Explanation:

Answer: B

- Option A is incorrect because setting the SMOTE percentage to 0 will generate no additional minority items, the dataset remains unchanged.
- Option B is CORRECT because in order to double the *percentage* of the minority class in the dataset, you have to set the SMOTE percentage parameter to 200. It will result in twice as much percentage (not number of cases!) in the dataset as initially. Setting the Random seed to the same, not null value for different runs guarantees reproducibility of the results.
- Option C is incorrect because setting the SMOTE percentage to 200 actually gives the expected result, but leaving the Random seed will end in different results over several runs.
- Option D is incorrect because SMOTE percentage = 100 will double the number of minority *cases*, which result in a higher percentage but the percentage is not necessarily doubled.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/smote
- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/smote#how-to-configure-smote
- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/smote#more-about-smote

Question 13

Domain: Run experiments and train models

During your machine learning experiments, you need to use the PyTorch framework for training models and you need GPU for high performance. One of your colleagues comes up with the following code, telling you that it could save you manual work:

```
# connect to workspace
from azureml.core import Workspace, Environment
ws = Workspace.from_config()
...
# set environment
my_env = Environment.get(workspace=ws,name="AzureML-PyTorch-1.1-GPU")
...
```

Does it really make your life easier or not?

- A. Yes
- B. No

Explanation:

Answer: A

- Option A is CORRECT because, besides enabling you to define your own run environments from scratch, Azure ML comes with a bunch of pre-built ("curated") environments for typical ML scenarios. You can use them easily, by simply using them in your "Environment" definition. "AzureML-PyTorch-1.1-GPU" is one of them.
- Option B is incorrect because Azure provides a lot of pre-defined environments for typical ML scenarios. If your requirements fit for these "curated" specifications, you can readily use them, instead of specifying manually.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-environments
- https://docs.microsoft.com/en-us/azure/machine-learning/resource-curated-environments

Question 14

Domain: Run experiments and train models

Runs of machine learning experiments produce a lot of metrics and outputs which you want to track across several runs, for evaluation reasons. Logging the relevant metrics helps you diagnose errors and tracking performance metrics. If you want to add named metrics to the runs, you can do it via the several logging methods of the Run object.

Which one of the following cannot be used for adding log metrics to the Run object?

- A. log()
- B. log_row()
- C. get_metrics()
- D. log_image()

Explanation:

Answer: C

- Option A is incorrect because the log() function can be used to record a numerical or string value to the run with the given name.
- Option B is incorrect because it is a valid function that creates a metric with multiple columns. It can be called once to record only one row or multiple times in a loop to generate a table.
- Option C is CORRECT because the run.get_metrics() is not for writing logs. It is used to get the user metrics of a trained model.
- Option D is incorrect because it is a valid function that logs an image file or a matplotlib plot to the run.

Reference:

- https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.run(class)?view=azure-ml-py
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-log-view-metrics

Question 15

Domain: Implement responsible machine learning

You have just finished with training of your ML model. The model is now ready to be deployed to its production environment as real-time inferencing service. Because of the very high load anticipated, for performance and scalability reasons you want to deploy the model as ***fraud-service-01*** to Azure Kubernetes Service. After connecting your ***aks-cluster-01*** cluster to your workspace, you want to go on with the deployment by using the Azure CLI (with ML extension).
You have the following CLI command as template:

```
az ml model deploy
<insert code here>
--model mymodel:1
<insert code here>
--inference-config inferenceconfig.json
--deployment-config deploymentconfig.json
```

Which two of the following parameters need to be added to the command?

- A. --name aks-cluster-01
- B. --compute-target aks-cluster-01
- C. --inference-target aks-cluster-01
- D. --name fraud-service-01
- E. --deployment-target fraud-service-01

Explanation:

Answers: B and D

- Option A is incorrect because the name parameter is used to set the name of the service to be deployed. In this case it should be: myservice.
- Option B is CORRECT because the name of the compute target, i.e. the AKS cluster (in the sample: aks-cluster-01) must be specified. Omitting the parameter will result in deploying the service to ACI.
- Option C is incorrect because this parameter doesn't exist. The parameter used to set the inference compute is "compute-target".
- Option D is CORRECT because the name of the service to be deployed is a required parameter. The correct command looks like this: az ml model deploy -ct aks-cluster-01 -m mymodel:1 -n myservice -ic inferenceconfig.json -dc deploymentconfig.json
- Option E is incorrect because this parameter doesn't exist. The parameter used to set the inference compute is "compute-target".

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-azure-kubernetes-service?tabs=azure-cli#deploy-to-aks

Question 16

Domain: Manage Azure resources for machine learning

You are going to build an automated process in order to speed up the integration and testing of your models, to make a robust development and testing process. This process needs to have access to the objects in your ML workspace, and it should work in the background, requiring no user interaction.

Which authentication method is recommended to use in most cases?

- A. Role Based Access Control
- B. Managed identity
- C. Service principal
- D. Interactive

Explanation:

Answer: C

- Option A is incorrect because the Role Based Access Control (RBAC) is used to limit and control the scope granted to an identity over a resource. It helps you manage who has access to Azure resources, and what they can do with those resources. It is used together with authentication workflows like interactive, service principal and managed identity.
- Option B is incorrect because managed identity is only supported when using Azure ML SDK from an Azure virtual machine.
- Option C is CORRECT because using service principals is the preferred way of authentication when there is a need to connect to the ML workspace without individual user accounts, within automated processes.
- Option D is incorrect because the interactive way of authentication is used for user-based authentication, and it requires direct authentication from users, i.e. it is not suitable for automated processes.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-setup-authentication
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-assign-roles

Question 17

Domain: Run experiments and train models

Your task is to gather data from several company data sources, so that it could be available for machine learning models. Data is coming from disparate sources and you need to find a solution to create standardized flows for ingesting and preprocessing in an automated way. The transformations you have to incorporate in the ETL process are typically rather complicated, long-running (sometimes over 30 mins) processes. You decide to use Azure Data Factory, due to its versatility in supported methods.

Which combination of tools would support your task in a most cost-effective way?

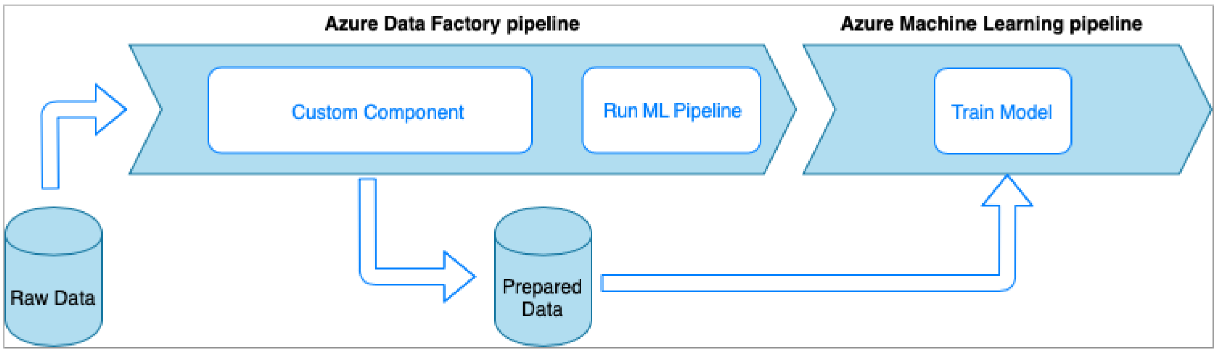
- A. ADF with Azure Function Activity
- B. ADF with Custom Activity
- C. ADF with Azure Databricks Notebook Activity
- D. ADF with Data Flow Activity

Explanation:

Answer: B

- Option A is incorrect because the serverless solution of Azure Functions are excellent means for short-running (less than 15 mins) processes. Since your transformation processes are "complicated and long-running", this is not an option in this case.
- Option B is CORRECT because the most effective (and also cost-effective) way of solving the problem is to add a Custom Activity to your ADF pipeline and add the transformation logic in the format of a custom Python script.
- Option C is incorrect because while being very powerful, creation of the Databricks infrastructure takes up time, and using it can be expensive. It is a perfect choice for distributed data processing at scale but with high cost implications.
- Option D is incorrect because Data Factory's Data Flow activity is very useful in cases when simple transformations (like mapping) are needed. For complicated, more sophisticated transformation scenarios enhance ADF flows with custom code extensions.

Diagram - Azure Data Factory with Custom Activity



Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-data-ingest-adf
- https://docs.microsoft.com/en-us/azure/machine-learning/concept-data-ingestion

Question 18

Domain: Implement responsible machine learning

You have successfully trained your ML model and now it is ready to be deployed as a real-time inferencing web service. You've just started writing the script which configures the deployment including the model to deploy, the compute environment etc.

Related to this task, which two of the following statements are true?

- A. You define the path to your registered model in the inference_config; call model to make predictions in the run() section of the entry script; set the compute memory size in deployment_config

- B. You define the path to your model in the entry script; add reference to to your entry script into inference_config; define the size of the compute in deployment_config
- C. You define the environment and the compute size in the inference_config; define the path to your model in the init() section of the entry script.

- D. The inference_config links the model script and the run environment together
- E. It is the run() section of the entry script which loads a registered model from a designated folder

Explanation:

Answers: B and D

- Option A is incorrect because the path to the registered model (or models) you want to deploy must be defined in the entry script (in the init() section) rather than in the inference_config. The rest of the statement is true.
- Option B is CORRECT because it is the entry script which is used to define the path to the model(s) to be deployed; the inference configuration links the entry script and its runtime configuration together, and the deployment configuration describes all the attributes of the deployment's target compute.
- Option C is incorrect because all the compute parameters (including number of cores, size of memory, autoscaling limits etc.) must be set in the deployment_config. The rest of the statement is true.
- Option D is CORRECT because it is actually the inference_config which is used to connect the script to be run with its run environment. Sample inferencec_config:

```
              inference_config = InferenceConfig(runtime= "python",

                                  entry_script=script_file, # entry script

                                  conda_file=env_file) # reference to environment
definition
```

- Option E is incorrect because it is the init() section of the entry script which is used to load a registered model in the initialization phase of the service deployment. The run() function handles requests while the model is running.

Reference:

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

Question 19

Domain: Manage Azure resources for machine learning

You are about building a machine learning environment in order to train models for processing a large number of jpg files. The files are stored in Azure Blob Storage. You need to find the best, most effective way to access from and use the files in your ML workspace.

What is the recommended way of linking and accessing data from your workspace?

- A. Register as a tabular dataset and access it by downloading
- B. Register as tabular dataset and access it by mounting
- C. Register as a file dataset and access it by downloading
- D. Register it as a file dataset and access it by mounting

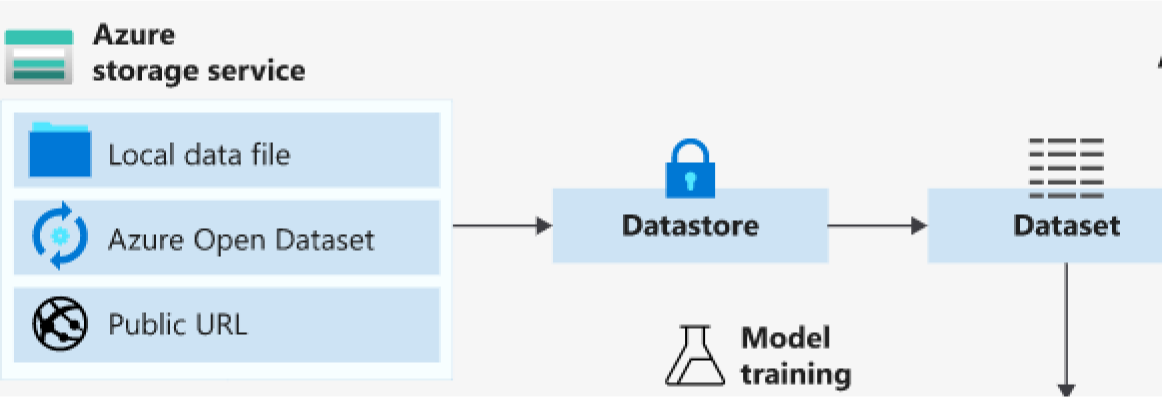
Explanation:

Answer: D

- Option A is incorrect because tabular datasets are used for structure data which can directly be loaded to dataframe structures. For jpg files, the file dataset type should be used. In addition, downloading the files is unnecessary and resource-intensive, use mounting instead.
- Option B is incorrect because tabular datasets are used for structure data which can directly be loaded to dataframe structures. Accessing the files via mounting is correct.
- Option C is incorrect because downloading the files is unnecessary and resource-intensive, the recommended method of accessing large amounts of remote data is by mounting their storage to the compute. Using datasets of type “file” is correct.
- Option D is CORRECT because a file dataset should be used for unstructured data (like images stored as files) and it is recommended to access the files by mounting them to the compute, in order to avoid unnecessary data movement from the storage. Data needed during the training will be transferred automatically, on demand.

Diagram:

Link storage account to ML workspace.



Reference:

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

Question 20

Domain: Manage Azure resources for machine learning

While creating a machine learning workspace, several associated resources are also created by Azure, which make your work with the workspace more convenient.

Which of the following items is not an associated resource and needs to be created manually when needed?

- A. Azure Container Registry
- B. Azure Data Lake Storage
- C. Azure Storage Account
- D. Azure Application Insights

Explanation:

Answer: B

- Option A is incorrect because Azure Container Registry with docker containers is at your disposal for deploying your model, without any manual provisioning.
- Option B is CORRECT because as the default (automatically created) storage for a workspace is an Azure Storage Account. If you want to use Data Lake Storage Gen2, for example because of its capability to manage hierarchical namespaces, you need to create one manually.
- Option C is incorrect because an Azure Storage Account is automatically created on creating the workspace. It is the default datastore for the workspace. It is a flat, non-hierarchical storage.
- Option D is incorrect because an Application Insights is also created on creating a workspace in order to store monitoring data related to your models.

Reference:

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

Question 21

Domain: Manage Azure resources for machine learning

You are a data scientist at your company and you are assigned to several ML projects. Therefore, you need to be able to create ML experiments and to run them on compute resources. There are different roles that can be used for performing tasks on workspaces.

Which predefined role fits best for your tasks?

- A. Data-scientist
- B. Reader
- C. Contributor
- D. Owner

Explanation:

Answer: C

- Option A is incorrect because Azure ML workspace comes with three default roles: Owner, Contributor, Reader. There is no such a predefined role as Data-scientist.
- Option B is incorrect because the Reader is one of the three default roles which are created while a workspace is created. As its name suggests, it only provides read privileges to workspace objects, i.e. it isn’t sufficient for data scientists who want to create and run experiments.
- Option C is CORRECT because the Contributor role provides users with the ability to create experiments, attach computes, run experiments and deploy web services, but also enables creating/deleting compute resources. It is stated that you need to work on predefined computes, this is not the best option for your role.
- Option D is incorrect because the Owner role grants full access to the workspace, which, in a large organization, should be limited to certain users, not to be exposed for those who only need user-level privileges.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-assign-roles#create-custom-role

Question 22

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 PFIExplainer to help you understand why this specific transaction has been classified as "suspicious".

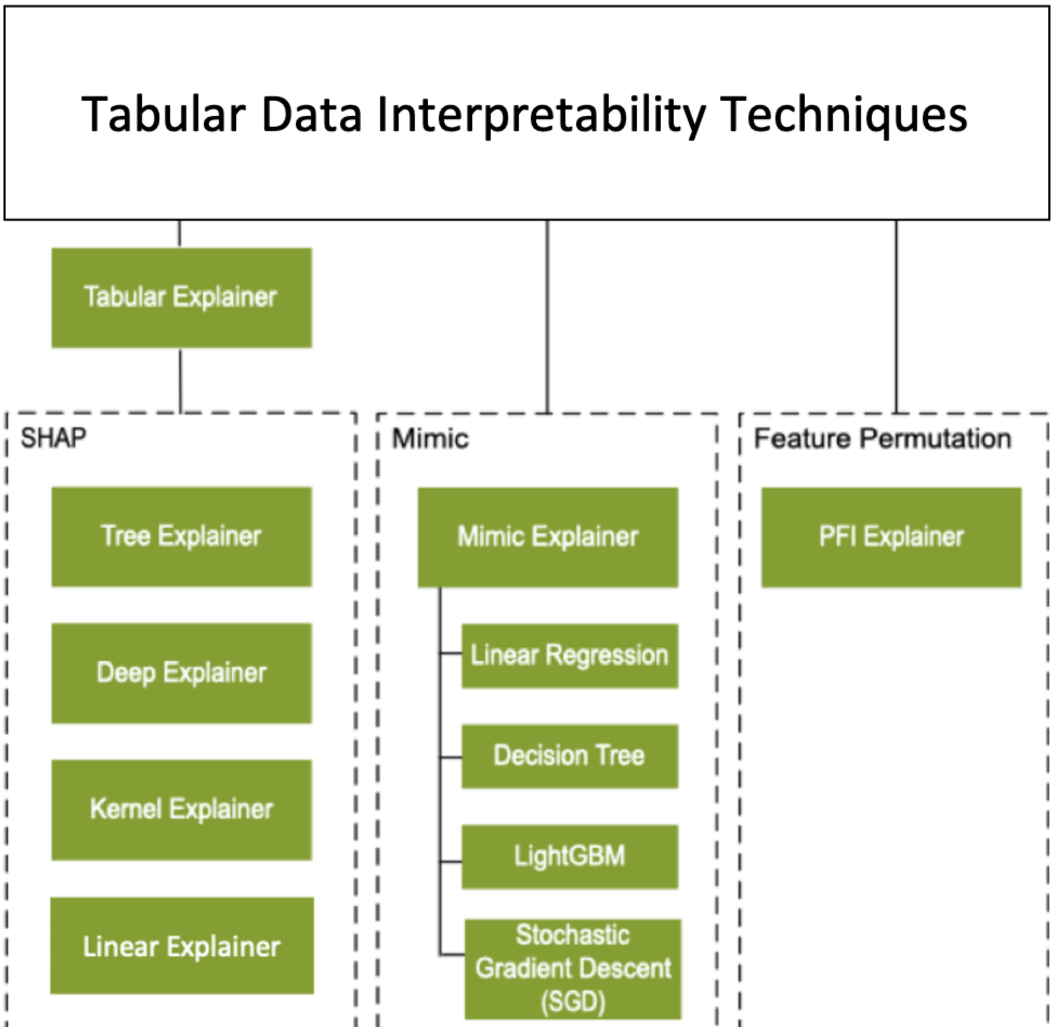
Does it serve your purpose?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because PFIExplainer doesn't support interpreting local feature importance. Mimic or Tabular explainer should be used instead.
- Option B 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. Choosing PFI cannot be used in this case.



Reference:

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

Question 23

Domain: Manage Azure resources for machine learning

For your machine learning experiments, you need to set up a multi node compute cluster for your training runs. You need to access your compute cluster from several resources, with the same access privileges.

Which is the most convenient and secure way you should choose?

- A. Include your credentials in your code when accessing the compute target
- B. Attach a user-assigned managed identity to your compute resource
- C. Attach a system-assigned managed identity to your compute resource
- D. Either system- or user-assigned managed identities fit the purpose

Explanation:

Answer: B

- Option A is incorrect because embedding credentials in any code must be avoided in all cases. Azure provides several ways of managing credentials, permissions without exposing them.
- Option B is CORRECT because Azure provides the managed identities feature to eliminate the need for managing credentials manually. You can either select system-assigned or user-assigned type, but only the user-assigned managed entities provide the reusability with the least administration. A user-assigned managed identity needs to be created as a separate entity in the Azure AD and can be attached to the compute cluster via the Advanced settings.
- Option C is incorrect because a system-assigned managed entity is linked directly to a single resource, in this case to a single compute cluster and cannot be used for accessing multiple resources. Its life-cycle starts when the resource is created and it ends when it is deleted.
- Option D is incorrect because only user-assigned managed entities serve your purpose. System-assigned managed entities are dedicated to a single resource and cannot be reused.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-attach-compute-studio#managed-identity

Question 24

Domain: Deploy and operationalize machine learning solutions

Azure ML enables you to interpret your model's results during training by adding a tabular explainer to your experiment script. Your model is now trained, you want to deploy it, and you also want to know how it behaves under real circumstances and you want to generate explanations also during inferencing, together with predictions. You want to embed a scoring explainer in your code.

Which steps do you need to execute, in what sequence?

1. Create a scoring explainer
2. Add the scoring explainer to your model's entry script
3. Create a tabular explainer
4. Deploy your model and the tabular explainer together
5. Register scoring explainer as a model
6. Deploy your model and the scoring explainer together

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

Explanation:

Answer: B

- Option A is incorrect because in order to create a scoring explainer, you need to create a “standard” explainer object (in this case: tabular) first.
- Option B is CORRECT because in order to generate feature data at inference time, you need to create an explainer (e.g. tabular), wrap it into a scoring explainer, register it as a model, add it to your model’s entry script (so that it can be invoked), and deploy it together with your model. Then it’s ready for use.
- Option C is incorrect because the scoring explainer is actually a wrapper around an explainer (e.g. tabular). What you need to deploy is the scoring explainer, as a scoring model. Deploying the tabular explainer is unnecessary.
- Option D is incorrect because the scoring explainer is actually a wrapper around an explainer (e.g. tabular). What you need to deploy is the scoring explainer, as a scoring model. Deploying the tabular explainer is unnecessary.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability-aml#interpretability-at-inference-time
- https://docs.microsoft.com/en-us/python/api/azureml-interpret/azureml.interpret.scoring.scoring_explainer?view=azure-ml-py

Question 25

Domain: Run experiments and train models

During your machine learning experiments, you need to use the PyTorch framework for training models. One of your colleagues tells you that you can save manual work by using Azure's pre-configured environment for PyTorch and he comes up with the following code:

```
# connect to workspace

from azureml.core import Workspace, Environment

ws = Workspace.from_config()

...

# set environment

my_env = Environment.get(workspace=ws,name="PyTorch-1.1-CPU")

...
```

Does this code do its job as intended?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because Azure actually provides you with a bunch of pre-built ("curated") environments for typical ML scenarios. You can use them easily, by simply using them in your "Environment" definition. The name of the curated environments start with the reserved prefix "AzureML". Therefore, ML won't recognize "PyTorch-1.1-CPU" as a valid curated environment.
- Option B is CORRECT because Azure actually provides a lot of pre-defined environments for typical ML scenarios, their name must start with the "AzureML" prefix, which is missing from the script, hence the script won't work as expected. "AzureML-PyTorch-1.1-CPU" is the correct name.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-environments
- https://docs.microsoft.com/en-us/azure/machine-learning/resource-curated-environments

Question 26

Domain: Run experiments and train models

You are storing your training data (jpg image files) in an Azure Blob Storage that you have registered as a datastore and a file dataset. You want to run your training script which can access your files from this datastore.

How should you do that?

- A. Include list of your file names in your training script
- B. Create a script parameter ('--data-folder') and pass the reference to your datastore to the training script within this parameter
- C. Store your script in the same location as your data is located
- D. Define a global parameter for the path name of your data and let your script use this for accessing the files.

Explanation:

Answer: B

- Option A is incorrect because the training script will load the input data from a datastore passed to it as a reference within an estimator parameter. Coding the list of files in the script is not a valid solution.
- Option B is CORRECT because if you want to use a datastore in experiments scripts, you have to pass a reference to the datastore as an input parameter for the script (via the estimator). The training script then can use data at the referenced location as local files.
- Option C is incorrect because reference to the location of the data must be passed as a parameter to the training script.
- Option D is incorrect because configuring the runs to access data in a datastore can be done by passing a reference to the data as an input parameter for the script. Global parameters are not the way of solving the problem.

Example:

```
# configure an estimator

data_ref =
blob_ds.path('input_data/training_files').as_download(path_on_compute='
training_data')

my_estimator = SKLearn(source_directory='experiment_folder',

      entry_script='training_script.py'

      compute_target='local',

      script_params = {'--data_folder': data_ref})

# submit script with my_estimator

...

# how to pass datastore reference to a training script

…

# define parameter for the script:

parser.add_argument('--data-folder', type=str, dest='data_folder',
help='data folder reference')
```

Reference:

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

Question 27

Domain: Implement responsible machine learning

You have to deploy your model as a web service to AKS. After starting the deployment process, you get the following error message:

Couldn't Schedule because the kubernetes cluster didn't have available resources after trying for 00:05:00

Which is not a possible solution for the problem?

- A. Add more nodes to your AKS cluster
- B. Decrease the resource requirements of your service
- C. Review the model path in your scoring script and re-register your model
- D. Change the SKU of your nodes

Explanation:

Answer: C

- Option A is incorrect because the reason behind this message is the lack of AKS resources during deployment. One possible solution is to allocate additional compute nodes.
- Option B is incorrect because the reason behind this message is the lack of AKS resources during deployment. One possible solution is to review the resource requirement of your service and optimize it in order to save resources.
- Option C is CORRECT because the error message occurs when the deployment of the service fails because of the lack of certain resources. Reviewing/changing the model path won't help in this case.
- Option D is incorrect because the reason behind this message is the lack of AKS resources during deployment. Selecting a different SKU, optimized to your workload might be a possible solution.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment?tabs=azcli
- https://docs.microsoft.com/en-us/azure/data-explorer/manage-cluster-choose-sku

Question 28

Domain: Implement responsible machine learning

You are working for a large pharmaceutical company, and their research laboratory generates a huge amount of test results daily. You, as a data scientist is tasked to build an ML solution to ingest the batches of data and generate predictions which can be further used by the research engineers. You want to build a machine learning pipeline deployed as a web service, which can be fed with the daily batches of data.

Which of the following should you consider as best practice while building your pipeline?

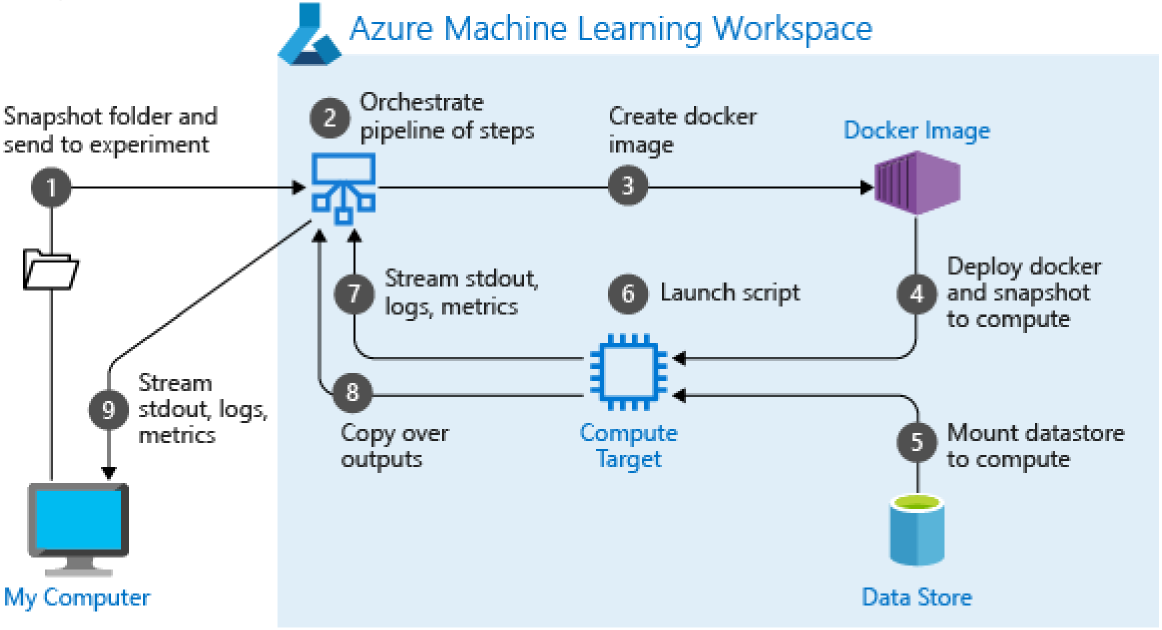
- A. The pipeline steps must use the same compute target
- B. Scripts for each pipeline step should be kept in separate folders
- C. Pipeline steps should be configured not to allow reuse
- D. When publishing the pipeline, interactive authentication is the recommended way of authentication

Explanation:

Answer: B

- Option A is incorrect because the compute target to be used can be configured for each pipeline step, although it is a common practice to set it at the pipeline level.
- Option B is CORRECT because by keeping the scripts of each pipeline step in a separate source directory helps saving resources, because if anything changes in a particular step's folder, only the affected step will be re-run. Although nothing prevents you from keeping all your code in one single folder, this is not recommended.
- Option C is incorrect because allowing to reuse the results of previous runs can radically reduce the resource need and execution time of the pipeline. Allow reuse whenever it is applicable.
- Option D is incorrect because interactive authentication (InteractiveLoginAuthentication) should be used only for development/testing purposes. For production scenario use ServicePrincipalAuthentication.

Diagram - Working with pipelines:


Azure Machine Learning Workspace
Snapshot folder and send to experiment
1
2 Orchestrate pipeline of steps
Create docker image
3
Docker Image
Stream stdout, logs, metrics
7
6 Launch script
4 Deploy docker and snapshot to compute
9 Stream stdout, logs, metrics
8
Copy over outputs
Compute Target
5 Mount datastore to compute
My Computer
Data Store

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/tutorial-pipeline-batch-scoring-classification#create-the-pipeline-step
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-your-first-pipeline

Question 29

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 are using the following script to deploy your model:

```
# deploy model
inference_config = InferenceConfig(runtime= "python",
                                   entry_script=script_file,
                                   conda_file=env_file)
deployment_config = AciWebservice.deploy_configuration(cpu_cores = 1,
memory_gb = 4)
service_name = "fraud-detection-service"
service = Model.deploy(ws, service_name, [model], inference_config,
deployment_config)
service.wait_for_deployment(True)
print(service.state)
```

Running the deployment script results in the service state “Failed”. You have a look at your scoring script and you suspect that something is wrong with getting data in the run() function:

```
# scoring script

...

def init():

    global model

    # Get the path to the deployed model file and load it

    model_path = Model.get_model_path('fraud_detection_model')

    model = joblib.load(model_path)

# Called when a request is received

def run(raw_data):

    # Get the input data as a numpy array

    data = np.array(json.loads(raw_data)['data'])

    # Get a prediction from the model

    predictions = model.predict(data)

    # Get the classnames for predictions

    classnames = ['non-fraud', 'fraud']

    predicted_classes = []
    for prediction in predictions:

        predicted_classes.append(classnames[prediction])

    # Return the predictions as JSON

    return json.dumps(predicted_classes)
```

Is this the best way to localize the error?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because you experienced the error while deploying the service, which means that the deployment process failed. It typically occurs when the get_model_path() function which is called during the deployment while initializing the model cannot locate the model's path for some reason. Therefore, you should look at the init() function first, for example this line:

  ```
      model_path = Model.get_model_path('fraud_detection_model')
  ```
- Option B is CORRECT because the error occurred during the deployment of the service, which means that it is not a data-related problem (if it was, it would occur during inference phase, while running the service). Since the deployment has failed, the problem is not inference-related. You'd better check the init() function.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python#understanding-service-state
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment?tabs=azcli#service-launch-fails

Question 30

Domain: Run experiments and train models

You are developing your training scenarios in Azure's notebook environment. You want to see the progress of your training runs by monitoring performance metrics and possible errors in detail.

Which is the best method should you use?

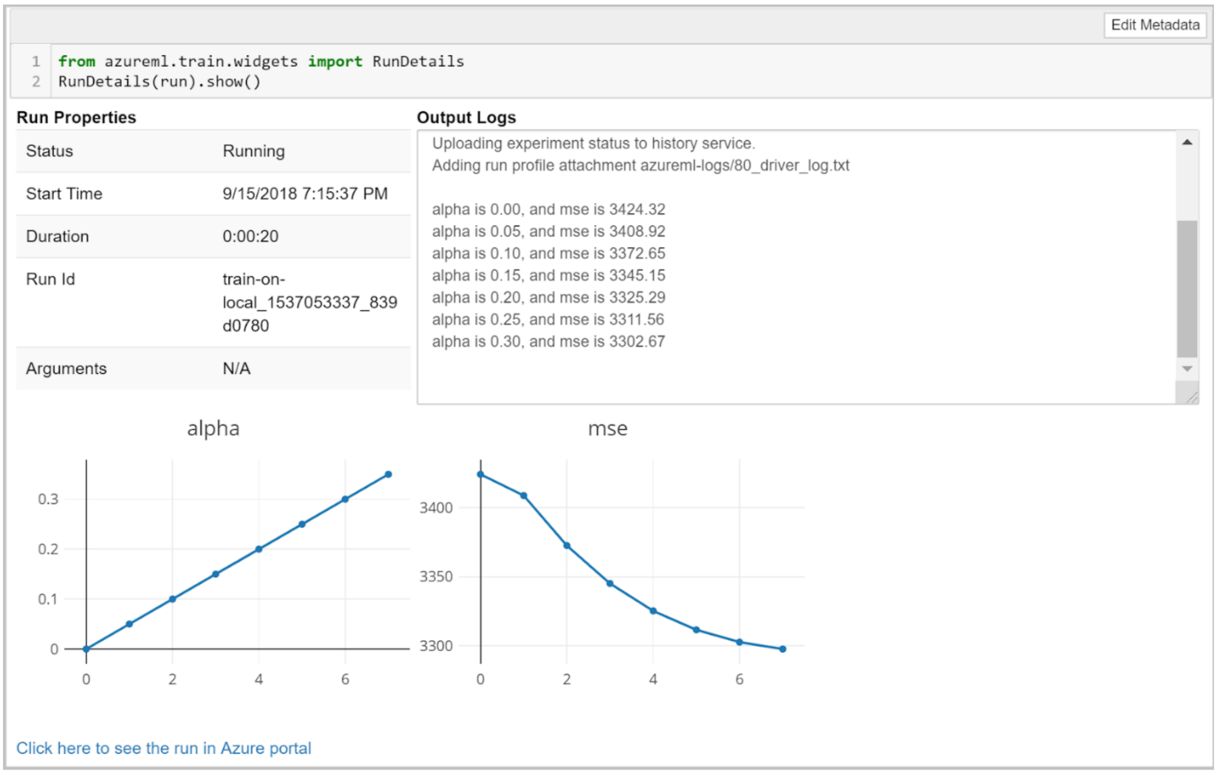
- A. Use the RunDetails class from the Jupiter widgets
- B. Include the run.get_metrics() in your script
- C. Use the get_status() of the run
- D. Use the run.wait_for_completion(show_output = True) in your script

Explanation:

Answer: A

- Option A is CORRECT because in Azure's notebook environment, the most comfortable way of monitoring the progress of a run is using the Jupiter RunDetails widget. The widget is running asynchronously and updates its output regularly (10-15 seconds) until the run completes.
- Option B is incorrect because the get_metrics() method of the Run object is used to retrieve the metrics logged on a run. It gives results after the run has completed.
- Option C is incorrect because get_status() returns the latest status of the run, like "Running", "Failed" etc. It doesn't provide detailed in-progress information.
- Option D is incorrect because this results in displaying the run's result only after its completion. It is not suitable for watching the progress.

Diagram - Notebook widget example output:



```
from azureml.train.widgets import RunDetails
RunDetails(run).show()
```

Run Properties

Status	Running
Start Time	9/15/2018 7:15:37 PM
Duration	0:00:20
Run Id	train-on-local_1537053337_839d0780
Arguments	N/A

Output Logs

```
Uploading experiment status to history service.
Adding run profile attachment azureml-logs/80_driver_log.txt

alpha is 0.00, and mse is 3424.32
alpha is 0.05, and mse is 3408.92
alpha is 0.10, and mse is 3372.65
alpha is 0.15, and mse is 3345.15
alpha is 0.20, and mse is 3325.29
alpha is 0.25, and mse is 3311.56
alpha is 0.30, and mse is 3302.67
```

Click here to see the run in Azure portal

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-view-training-logs
- https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.run%28class%29?preserve-view=true&view=azure-ml-py#get-status--

Question 31

Domain: Run experiments and train models

You have trained a binary classification model in order to build

You have built an inference solution to analyze real-time vibration data coming from turbines of a power plant. As a part of a predictive maintenance solution, the model's goal is to predict if a machinery is at risk of breaking down. In order to determine your model's performance, you used the Run object's log_confusion_matrix() method to log the result of the runs. After examining the confusion matrix data in the log, you conclude that the model is performing pretty well.

Which of the following relationships support your conclusion?

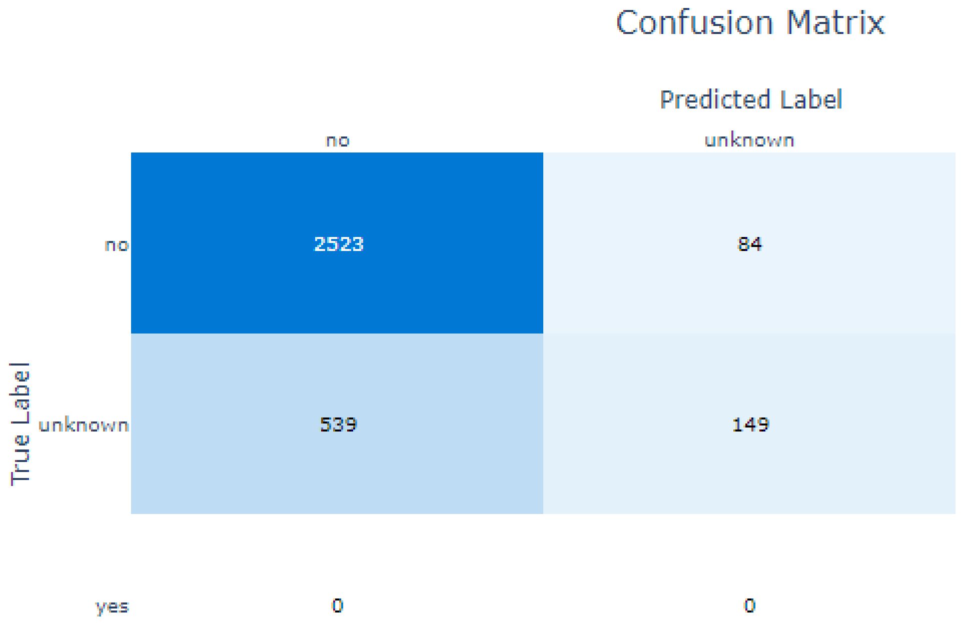
- A. TP is high; FP is low; TN is low; FN is high
- B. TP is low; FP is high; TN is high; FN is low
- C. TP is high; FP is low; TN high; FN low
- D. The confusion matrix cannot be used to support your conclusion

Explanation:

Answer: C

- Option A is incorrect because based on the confusion matrix, the model tends to classify cases "not at risk" when actually they are, hence the model is poor at predicting one of the classes.
- Option B is incorrect because based on the confusion matrix, the model tends to classify cases "at risk" when they are actually not, hence the model is poor at predicting one of the classes.
- Option C is CORRECT because, based on the confusion matrix, a classification model's performance is good if the number of true positives (TP) is higher than that of the false positives (FP), and the same is true for the true negatives (TN) vs. false negatives (FN).
- Option D is incorrect because it actually can be used. The confusion matrix visualizes the frequency of the correctly and incorrectly classified cases vs. the frequency of actual cases in each category. You can use it to draw initial conclusions about the model's performance. Other metrics like precision, recall should be used, as well.

Diagram - Confusion matrix example:



	no	unknown
no	2523	84
unknown	539	149
yes	0	0

Reference:

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

Question 32

Domain: Deploy and operationalize machine learning solutions

You have a large dataset of observations with a high number of features. You need to train a multiclass classification model with hyperparameter tuning in a time- and cost-effective way.

Which of the following decisions helps you to reduce training time and save cost?

- A. Use Grid sampling
- B. Use the Default Termination Policy
- C. Use Filter Based Feature Selection
- D. Disable overfitting

Explanation:

Answer: C

- Option A is incorrect because Grid sampling is used during parameter tuning. It is applicable for discrete model parameters, to sweep over the entire search space. It requires a lot of time, so use this method only if your budget allows for the exhaustive search.
- Option B is incorrect because the default termination policy is “no forced termination” during hyperparameter tuning, which means that the hyperparameter tuning service will let all training runs complete, i.e. it doesn’t serve saving time and cost.
- Option C is CORRECT because by using feature selection you include a process of applying statistical tests to inputs. The goal is to find the columns which are more predictive of the output. The Filter Based Feature Selection module provides several feature selection algorithms you can choose from. Reducing number of features can have remarkable effect on the training time of the model
- Option D is incorrect because overfitting occurs when a model fits the training data too well, and as a result, it can't accurately predict on new data. It is a kind of “model error”, not related with the training time. Overfitting is not something you can disable.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-select-algorithms#number-of-features
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters#grid-sampling
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters#no-termination-policy-default

Question 33

Domain: Deploy and operationalize machine learning solutions

Your task is to build a regression model for predicting demand for your car rental service for the following quarter. For scoring your model, you need to define a primary metric which can be used by the Scoring step. You decide to use the Logistic regression algorithm, and the *normalized_mean_absolute_error* as primary metric. Is it an appropriate decision?

- A. Yes, both the algorithm and the selected primary metric fit for your purpose
- B. No, because algorithm is correct, but the primary metric is not applicable for your case
- C. No, because neither the algorithm, nor the primary metric are applicable
- D. No, because the Log Reg algorithm is not applicable here; the primary metric is correct.

Explanation:

Answer: D

- Option A is incorrect because the selected algorithm is not applicable because Logistic Regression can only be used in classification scenarios. The selected metric is a valid option for regression tasks.
- Option B is incorrect because the selected algorithm is not applicable. The metric, however, is a valid option for regression tasks.
- Option C is incorrect because the selected algorithm is actually not applicable, however selection of normalized_mean_absolute_error as primary metric is a valid option for regression tasks.
- Option D is CORRECT because Logistic Regression, despite its name, is an algorithm for classification, not for regression.

Reference:

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

Question 34

Domain: Deploy and operationalize machine learning solutions

You are running experiments in Azure ML and you want to configure the model's hyperparameters. You need to define the sampling space for two parameters and you want Azure ML to use a sampling method which picks samples based on the result of previous runs, in order to improve the performance in the primary metric.

```
# define parameter space
from azureml.train.hyperdrive import BayesianParameterSampling, choice,
uniform
my_parameter_space = {
        'batch_size': choice(8, 16, 32, 64),
        "keep_probability": uniform(0.05, 0.1)
    }
param_sampling = BayesianParameterSampling(my_parameter_space)
...
```

Does the script above fulfills your requirement?

- A. Yes
- B. No

Explanation:

Answer: A

- Option A is CORRECT because the Bayesian parameter sampling is the right choice, and this sampling method supports choice, uniform, and quniform distributions. Therefore, the code is correct.
- Option B is incorrect because the code defines the parameter search space for two parameters, batch_size and keep_probability, with the Bayesian sampling method. Bayesian sampling improves sampling based on the result of previous runs, and it can be used with either choice or uniform methods.

Reference:

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

Question 35

Domain: Deploy and operationalize machine learning solutions

You have an inference web service deployed on Azure Kubernetes Service. In order to detect possible deterioration of your model's performance, you want to collect data during the operation. You need to use the Python SDK and you want to add the necessary statements to your code.

```
...
# script1
global dc_inputs, dc_predictions
dc_inputs = ModelDataCollector("best_model", designation="inputs",
feature_names=["f1", "f2", "f3"])
dc_predictions = ModelDataCollector("best_model",
designation="predictions", feature_names=["p1", "p2"])
------------
#script2
data = np.array(data)
result = model.predict(data)
dc_inputs.collect(data) #this call is saving our input data into Azure
Blob
dc_predictions.collect(result) #this call is saving our input data into
Azure Blob
------------------
#script3
aks_config =
AksWebservice.deploy_configuration(collect_model_data=True)
...
```

Which segment goes to which part of your code?

- A. script1 to init(); script2 to run(); add script3 to deployment config
- B. script1 to init(); script2 to run(); script3 is not needed
- C. script2 to init(); script1 to run(); script3 to deployment config
- D. script1 and script3 to init(); script2 to run()

Explanation:

Answer: A

- Option A is CORRECT because the ModelDatCollector objects need to be declared in the init() function, so that they can be invoked in the run(). Data collection has to be enabled outside the entry script, as part of the deployment configuration.
- Option B is incorrect because data collection is not enabled by default when deploying a model to AKS. You have to enable it so that you can collect model data, by setting collect_model_data=True.
- Option C is incorrect because script1, as a variable declaration, should go to the init() function, while script2 has to go to the run().
- Option D is incorrect because data collection must be enabled (script3) outside the entry script, within the deployment configuration.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-enable-data-collection
- https://docs.microsoft.com/en-us/azure/machine-learning/concept-model-management-and-deployment

Question 36

Domain: Run experiments and train models

Your team is working for a medical research center which researches skin diseases. The Center is in connection with several medical centers where images of actual cases are taken and collected. The image files are sent to your team weekly, and your task is ingesting them into machine learning algorithms. You are using Azure ML Designer to build ML pipelines.

Which method should you use to ingest data?

- A. Create a file dataset from your data folder and drag the dataset to the canvas
- B. Drag the Import Data module to the canvas and set it to "File" type and link it to the source data folder
- C. Create a tabular dataset from your data folder and drag the dataset to the canvas
- D. Create a tabular dataset from your data folder, set the column headers to "All files have the same headers" and drag the dataset to the canvas

Explanation:

Answer: A

- Option A is CORRECT because if your data is contained in multiple unstructured (not table-like) files, the recommended way to ingest it in the pipeline is registering the source folder as a dataset, and use the dataset like any other modules in your pipeline. Type of the dataset must be set to "File".
- Option B is incorrect because the Iport Data module can cope with tabular data (CSV files, typically). It is not suitable for unstructured sources, hence "File" type cannot be set.
- Option C is incorrect because "Tabular" setting is not applicable for scenarios with unstructured sources. Use "Tabular" dataset type instead.
- Option D is incorrect because "Tabular" setting is not applicable for scenarios with unstructured sources. Use "Tabular" dataset type instead. Setting the column headers doesn't help.

Reference:

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

Question 37

Domain: Manage Azure resources for machine learning

In your machine learning workspace, you have a Compute Cluster of type Standard_D2s_v3 with 0/2 nodes, and a Compute Instance of type Standard_D2s_v3. You want to use these computes in the most cost-effective manner, i. e. you want to avoid being charged for unwanted costs.

Which best practice should you follow?

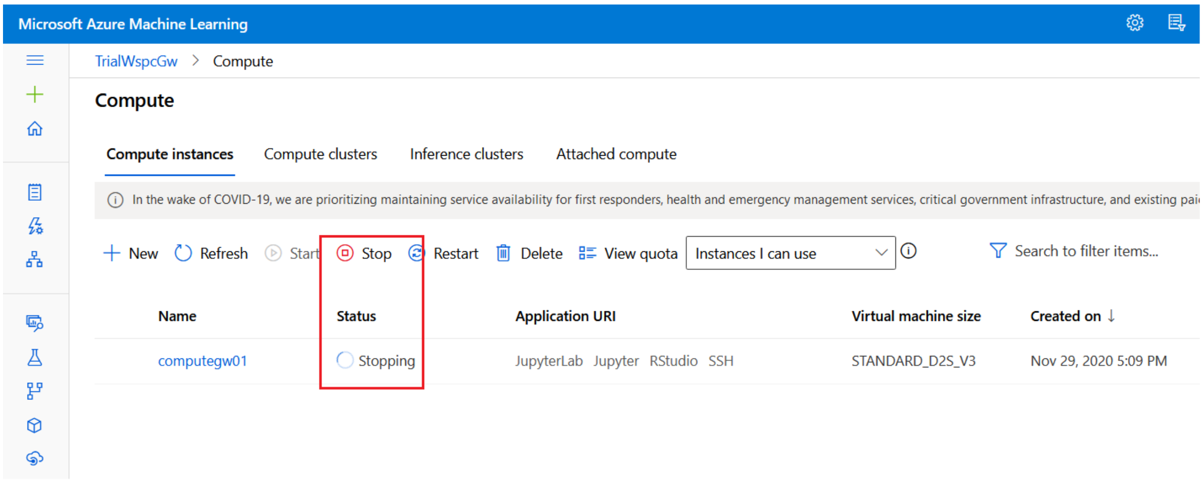
- A. Stop the Compute Instance while it is unused
- B. Stop the Compute Cluster while it is unused
- C. For the Compute Cluster, set the number of max nodes to 0
- D. Do nothing - your computes are stopped automatically when not used

Explanation:

Answer: A

- Option A is CORRECT because the compute instances are single VMs which must be managed manually. Once you start them, they remain in "Running" state until stopped (or deleted) manually. In order to save costs, stop them when they are not in use.
- Option B is incorrect because if the minimum number of nodes for the compute cluster is set to 0, the compute is automatically scales down to 0 nodes when running idle.
- Option C is incorrect because the compute cluster are spinning and scaling up when they are in use, depending on the workload. If the minimum number of nodes is set to 0, the cluster scales down to 0 nodes when not in use. The maximum number of nodes has no effect on costs.
- Option D is incorrect because while the compute cluster scales down automatically the minimum number of nodes, the compute instance is running generating cost even if it is idle.

Diagram - Managing compute instances



ⓘ In the wake of COVID-19, we are prioritizing maintaining service availability for first responders, health and emergency management services, critical government infrastructure, and existing pai

Name	Status	Application URI	Virtual machine size	Created on ↓
computegw01	Stopping	JupyterLab Jupyter RStudio SSH	STANDARD_D2S_V3	Nov 29, 2020 5:09 PM

Reference:

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

Question 38

Domain: Run experiments and train models

You already have the Python scripts for several steps (including data ingestion, data cleansing, dividing data into train and test sets etc.) of your machine learning tasks but you want to combine them into a consistent, repeatable flow. You want to make use of the orchestration services offered by Azure ML pipelines. You have defined 3 three steps, each of them referencing a piece of your Python code:

```
step1 = PythonScriptStep(name="train_step",
                         script_name="train.py",
                         compute_target=aml_compute,
                         source_directory=source_directory,
                         allow_reuse=False)
step2 = PythonScriptStep(name="compare_step",
                         script_name="compare.py",
                         compute_target=aml_compute_cluster2,
                         source_directory=source_directory,
                         allow_reuse=False)
step3 = PythonScriptStep(name="extract_step",
                         script_name="extract.py",
                         compute_target=aml_compute,
                         source_directory=source_directory,
                         runconfig=run_config)
```

Which parts of the codes should be changed to ensure optimal performance?

- A. allow_reuse should be set True for the 1st step of a pipeline

- B. source_directory should reference to different folders for each step
- C. compute _target should be the same compute for each step in a pipeline

- D. parameter runconfig should be set for each step

Explanation:

Answer: B

- Option A is incorrect because there is no such constraint for allow_reuse. This is an optional parameter with default value of True. It determines whether outputs of the step's previous run can be used by the following steps in order to save execution time. Either True or False can be correct.
- Option B is CORRECT because in order to optimize the behavior of the pipeline during runs, the recommended practice is to use separate folders for storing scripts and its dependent files for each step. These folders should be the source_directory for the steps. This way, the size of the snapshot created for the step can be reduced.
- Option C is incorrect because there is a compute target at pipeline level which is used by the steps unless specified otherwise at step level. Steps with individual compute requirements can define their own compute target.
- Option D is incorrect because the runconfig parameter can be used to specify additional requirements for the run, such as conda dependencies (e.g. 'scikit-learn'). When missing, a default runconfig will be created. See step parameters below:

PythonScriptStep(script_name, name=None, arguments=None, compute_target=None, runconfig=None, runconfig_pipeline_params=None, inputs=None, outputs=None,params=None, source_directory=None, allow_reuse=True, version=None, hash_paths=None)

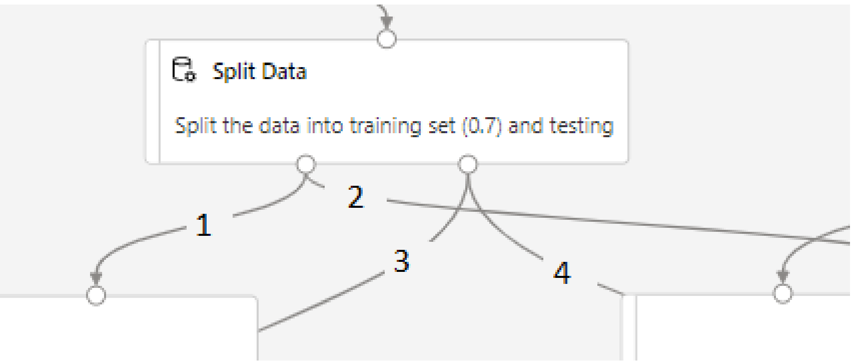
Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-create-your-first-pipeline
- https://github.com/MicrosoftLearning/DP100/blob/master/06A%20-%20Creating%20a%20Pipeline.ipynb
- https://github.com/MicrosoftLearning/DP100/blob/master/labdocs/Lab06A.md

Question 39

Domain: Run experiments and train models

You need to build an ML pipeline which takes an input dataset and trains two regression models so that you can compare their performance and, on the result of the comparison, you can decide which one to use for real time predictions. You know that the Split data module of the MD Designer is a great means of distributing data between the training subprocesses. The Split data module has 2 x 2 out data flows like this:



How do you need to connect the outputs of the Split data module?

- A. 1 to Boosted Decision Tree module; 2 to Decision forest module; 3 and 4 to Train model
- B. 1 and 2 to Train model; 3 and 4 to Score model
- C. 1 and 2 to Score model; 3 and 4 to Train model
- D. 1 and 2 to Train model; 3 and 4 to Evaluate model

Explanation:

Answer: B

- Option A is incorrect because the Boosted Decision Tree and the Decision forest modules are the two algorithms to be used. They are “inputs” to the Train model modules; they themselves don’t have input connectors.
- Option B is CORRECT because Split data separates its input data into two distinct sets, which are typically used for training and testing (scoring). In this case, 70% of the rows go to the training set and the rest will be used for testing. The training data should go into the two Train model modules, while the rest of the data will be used by the Score model modules.
- Option C is incorrect because 70% of the data rows go to the training set (1, 2), which forms the inputs of the Train model modules rather than the Scoring.
- Option D is incorrect because the Evaluate model takes its input from the Score model, i.e. connecting 3 and 4 will not give the expected result.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/split-data
- https://github.com/Azure/MachineLearningDesigner/blob/master/articles/samples/regression-automobile-price-prediction-compare-algorithms.md

Question 40

Domain: Run experiments and train models

You have built an ML pipeline in order to make your machine learning process reproducible and automated. For quality assurance reasons, you show your script to one of your senior data scientist colleagues, who, after reviewing it, she suggests using an Estimator step in your pipeline.

Why did she suggest it?

- A. Because it helps to estimate the performance of the model by using the primary metric
- B. Because it helps to to simplify the tasks of specifying how a script executes
- C. Because it helps to seamlessly pass data between the pipeline steps
- D. Because it enhances the explainability of the model

Explanation:

Answer: B

- Option A is incorrect because the Estimator, despite how its name suggests, doesn’t “estimate” and it is not used in the context of model metrics. This is a configuration object.
- Option B is CORRECT because the Estimator combines the Run Configuration and a Script Run Configuration to a single object, making the configuration task easier and more straightforward. By including it in an Estimator Step, even complex configurations can easily be included in a reproducible pipeline process.
- Option C is incorrect because for passing data between pipeline steps, the PipelineData object must be used.
- Option D is incorrect because it is the model explainers (like Tabular, Mimic etc.) that help interpret the behavior of the model. Estimator is a configuration object used to define the run context of the model.

```
#create estimator object

 from azureml.train.estimator import Estimator

 est = Estimator(source_directory=source_directory,

                compute_target=cpu_cluster,

                entry_script='dummy_train.py',

                conda_packages=['scikit-learn'])

 # create estimator step using the estimator object

 from azureml.pipeline.steps import EstimatorStep

 est_step = EstimatorStep(name="Estimator_Train",

                         estimator=est,

                         estimator_entry_script_arguments=["--
datadir",

                          input_data.as_mount(), "--output", output],

                         runconfig_pipeline_params=None,

                         compute_target=cpu_cluster)
```

Reference:

- https://github.com/Azure/MachineLearningNotebooks/blob/master/how-to-use-azureml/machine-learning-pipelines/intro-to-pipelines/aml-pipelines-how-to-use-estimatorstep.ipynb
- https://docs.microsoft.com/en-us/python/api/azureml-pipeline-steps/azureml.pipeline.steps.estimator_step.estimatorstep?view=azure-ml-py

Question 41

Domain: Run experiments and train models

You have created an ML pipeline of eight steps, using Python SDK. While tuning the script of steps3 and 6, you submit the pipeline for execution several times. The scripts and data definitions of the other steps didn't change, but still you notice that all the steps rerun each time, and you experience long execution times. You suspect that the pipeline is not reusing the output of steps, and you decide to set allow_reuse to True, for each step.

Will it probably solve the problem?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because the allow_reuse parameter of the pipeline steps is set to True, by default. Setting it explicitly to True in your code won't change the situation, i.e. it won't solve the problem.
- Option B is CORRECT because if you want to ensure that steps only rerun when their underlying data definition or scripts change, put their scripts to separate folders for each step. Otherwise, you might experience unnecessary reruns.

Reference:

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

Question 42

Domain: Run experiments and train models

You have created an ML pipeline of eight steps, using Python SDK. While tuning the script of steps3 and 6, you submit the pipeline for execution several times. The scripts and data definitions of the other steps didn't change, but still you notice that all the steps rerun each time, and you experience long execution times. You decide to separate the scripts and other configuration files to different folders for each step and to set the source_directory parameters accordingly.

Will it probably solve the problem?

- A. Yes
- B. No

Explanation:

Answer: A

- Option A is CORRECT because if you experience unexpected reruns of pipeline steps whose underlying code didn't change, you should put the scripts and configuration items to separate folders for each step. This should solve the problem.
- Option B is incorrect because unexpected rerun of pipeline steps is a clear indicator of the problem caused by storing the codes of multiple steps in one, common location. Any time, any of the scripts change, all the steps referencing this source_directory will rerun, consuming extra time and resources. This practice should be avoided.

Reference:

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

Question 43

Domain: Deploy and operationalize machine learning solutions

While setting up your machine learning experiments, you need to ensure that the trained models will be appropriately scored with validation data. Azure ML SDK provides several methods to specify in your scripts how to determine the data to be used for validation.

Which is not a valid set of parameters for specifying validation method? Select two!

- A. Primary metric; training data
- B. Primary metric; training data; validation data
- C. Primary metric; validation data; number of cross-validations
- D. Primary metric; training data; validation set size
- E. Primary metric; training data; validation data; validation set size

Explanation:

Answers: C and E

- Option A is incorrect because the primary metric must be given in all cases. When only training data is provided, the default splitting rules will be applied.
- Option B is incorrect because the primary metric is always required. When both training and validation data are provided, these will be used, since no automatic splitting is needed.
- Option C is CORRECT because primary metric and training data are always required. Validation data either can be provided explicitly, or can be generated automatically (by default or explicit splitting rules).
- Option D is incorrect because when only training data is provided, with validation set size set manually, this value will be used to split data into training and validation subsets.
- Option E is CORRECT because it is a kind of redundancy because either validation data or training data with validation set size can be set. Setting both validation data and validation set size is wrong.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-cross-validation-data-splits#provide-validation-data
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-cross-validation-data-splits#set-the-number-of-cross-validations

Question 44

Domain: Manage Azure resources for machine learning

You are a data engineer at your company and you are assigned to several ML projects. You need to be able to create and run ML experiments, create and delete computes in several workspaces,

Which default role should you be assigned to?

- A. Data-engineer
- B. Reader
- C. Contributor
- D. Owner

Explanation:

Answer: C

- Option A is incorrect because Azure ML workspace comes with three default roles: Owner, Contributor, Reader. Data-engineer is not one of them. A role like this can be created as a custom role, but if you need a default role, the Contributor should be used.
- Option B is incorrect because the Reader is one of the three default roles which are created while a workspace is created. As its name suggests, it only provides read privileges to workspace objects, i.e. it isn't sufficient for data scientists who want to create and run experiments.
- Option C is CORRECT because in order to be able to manage ML experiments end-to-end, you need the Contributor role. This provides you with the ability to create experiments, attach computes, run experiments and deploy web services, without granting unnecessary privileges.
- Option D is incorrect because the Owner role grants full access to the workspace, which, in a large organization, should be limited to certain users, not to be exposed for those who only need user-level privileges.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-assign-roles#troubleshooting

Question 45

Domain: Implement responsible machine learning

You have successfully trained your ML model and haven't yet registered it in your workspace. Now you need to deploy it to the runtime environment as a real-time inference service. Azure ML offers several methods for deploying a model and you have to choose that best fits your scenario.

Which of the following methods can you use in this case?

- A. deploy method of the Model
- B. deploy_from_model method of the Webservice
- C. deploy_from_image method of the Webservice
- D. deploy method of the Webservice

Explanation:

Answer: D

- Option A is incorrect because while the deploy method of the Model is the easiest way to deploy a model to its runtime environment, it can only be used for registered models only. It would require a registration step so that it can be used.
- Option B is incorrect because the deploy_from_model method of the Webservice object can also be used to deploy a model as a webservice, however it does not register the model. Use it for models already registered in the workspace.
- Option C is incorrect because the deploy_from_image method of the Webservice object can be used if you already have an image to be deployed. It takes an Image as an input parameter instead of a Model. Since you have finished with the training of your model, you probably don't have an image yet.
- Option D is CORRECT because the deploy method of the Webservice object can be used to deploy a model even if it is not registered. The method completes both its registration and deployment as a real-time service. This is the solution for deploying unregistered models in one step.

Reference:

- https://docs.microsoft.com/en-us/python/api/azureml-core/azureml.core.webservice.webservice(class)?view=azure-ml-py
- https://github.com/MicrosoftLearning/DP100/blob/master/07A%20-%20Creating%20a%20Real-time%20Inferencing%20Service.ipynb

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 are using the following script to deploy your model:

```
# deploy model
inference_config = InferenceConfig(runtime= "python",
                                   entry_script=script_file,
                                   conda_file=env_file)
deployment_config = AciWebservice.deploy_configuration(cpu_cores = 1,
memory_gb = 4)
service_name = "fraud-detection-service"
service = Model.deploy(ws, service_name, [model], inference_config,
deployment_config)
service.wait_for_deployment(True)
print(service.state)
```

Running the deployment script results in the service state “Failed”. You have a look at your scoring script and you suspect that something is wrong with model path in the init() function:

```
# scoring script
...
def init():
    global model
    # Get the path to the deployed model file and load it
    model_path = Model.get_model_path('fraud_detection_model')
    model = joblib.load(model_path)
# Called when a request is received
def run(raw_data):
    # Get the input data as a numpy array
    data = np.array(json.loads(raw_data)['data'])
    # Get a prediction from the model
```

```
predictions = model.predict(data)

# Get the classnames for predictions

classnames = ['non-fraud', 'fraud']

predicted_classes = []

for prediction in predictions:

    predicted_classes.append(classnames[prediction])

# Return the predictions as JSON

return json.dumps(predicted_classes)
```

Is this the best track to spot the error?

- [✓ A. Yes]
- B. No

Explanation:

Answer: A

- Option A is CORRECT because you experienced the error while deploying the service, which means that the deployment process failed. It typically occurs when the get_model_path() function which is called during the deployment while initializing the model cannot locate the model's path for some reason. Therefore, you should investigate this first.
- Option B is incorrect because the error occurred during the deployment of the service, which means that it is not a data-related problem (if it was, it would occur during inference phase, while running the service). Since the deployment has failed, the problem is not inference-related.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-deploy-and-where?tabs=python#understanding-service-state
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-troubleshoot-deployment?tabs=azcli#service-launch-fails

Question 47

Domain: Run experiments and train models

You are using the ML Studio for configuring your machine learning environment. You need to ingest data from a web location and an ML pipeline needs to be built for preparing data and training your model. Before you can run your first experiment, everything needs to be in place, every ML resource must be set up.

Which of the following actions don't you need to complete manually?

- A. Create Compute instance
- B. Create workspace
- C. Create Training cluster
- D. Create datastore

Explanation:

Answer: D

- Option A is incorrect because the compute instance is a fully configured workstation within your ML workspace. It is used to run the notebooks and which can also be used for smaller training workloads. You need to create one, manually.
- Option B is incorrect because creation of a machine learning workspace is the mandatory requirement for any machine learning experiment, pipeline etc. This is the very first step in configuring an ML work environment.
- Option C is incorrect because a training cluster is needed for running the training experiments. It must be configured manually before running the pipeline.
- Option D is CORRECT because as an associated resource, an Azure Blob storage is created automatically upon creation of the ML workspace. You don't need to create one manually. If your data resides in your company data stores (like in a Data Lake storage), you need to add them to the workspace, but the workspace has a blob store by default.

Reference:

- https://github.com/MicrosoftLearning/DP100/blob/master/labdocs/Lab01A.md

Question 48

Domain: Deploy and operationalize machine learning solutions

You are building an ML model, for which you want to find the optimal parameter setting which results in the best performing model. You decide to use the hyperparameter-tuning feature of Azure ML, i.e. use Hyperdrive in your experiments. Using Hyperdrive requires some specific conditions your script must fulfil.

Which components/settings are specific only for Hyperdrive experiments?

- A. Define ScriptConfig; create ScriptRunConfig
- B. Add script argument for hyperparameters; create an Estimator
- C. Add script argument for hyperparameters; Log primary metric
- D. Define training dataset; validation dataset

Explanation:

Answer: C

- Option A is incorrect because ScriptConfig and ScriptRunConfig are common configuration objects used for any ML experiment. They are not specific for Hyperdrive experiments.
- Option B is incorrect because adding a script argument for hyperparameters to be adjusted is specific for Hyperdrive indeed, estimators are commonly used in any experiment, as “wrappers” for ScriptConfig and ScriptRunConfig.
- Option C is CORRECT because if you want to tune model parameters using Hyperdrive, you must include a script argument for each parameter to be adjusted, as well as the primary performance metric (e.g. Accuracy) must be logged, so that Hyperdrive can evaluate the runs and it can select the best performer combination.
- Option D is incorrect because training and test/validation datasets are fundamental components of any ML experiment.

Reference:

- https://github.com/MicrosoftLearning/DP100/blob/master/08A%20-%20Tuning%20Hyperparameters.ipynb
- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters

Question 49

Domain: Deploy and operationalize machine learning solutions

As part of an IoT solution which collects a large amount of environmental data from various field sensors, including cameras, you have a forecasting ML model in production, deployed on an AKS cluster. Expecting that the performance of the image classification model is likely to degrade over time, you are going to implement an "early warning system" which triggers alarms when the performance metrics start to decline. For this purpose, you decide to enable data collection on your model, to be able to examine the incoming images with the model's inference results.

Does it help you achieve your goal?

- A. Yes
- B. No

Explanation:

Answer: B

- Option A is incorrect because while you can configure Azure ML's ModelDataCollector to collect and input data and predictions during model runs and store them in a blob storage, the feature isn't applicable for audio and video data, which is your focus in this scenario.
- Option B is CORRECT because when enabled, data collection can actually be used to collect model data, such as inputs and predictions, and collected data will be stored in the workspace's storage account. However, this feature isn't applicable in the case of large binary files like images and video data, which means that it won't fulfil your requirements.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-enable-data-collection
- https://docs.microsoft.com/en-us/python/api/azureml-monitoring/azureml.monitoring.modeldatacollector.modeldatacollector?view=azure-ml-py

Question 50

Domain: Deploy and operationalize machine learning solutions

You are working for a bank which has recently introduced an ML inference service to predict the churn of its customers. During the training phase, the model showed an excellent accuracy of 99.5% and its test accuracy was above 96%. However, in production use, for real cases the overall accuracy is around 60% which is rather low. It seems that the model tends to overfit data.

Which of the following actions are recommended practices?

- A. Decrease the number of features used for training; increase the model's complexity
- B. Decrease the number of observations used for training; add more features to the training data
- C. Get more observations for training; apply cross-validation
- D. Use regularization with hyperparameter tuning; add more features to the dataset

Explanation:

Answer: C

- Option A is incorrect because limiting the complexity of the algorithm is a feature in auto ML and serves to prevent models from being overly complicated and overfitting. The technique is usually applied for decision-tree algorithms to limit the depth of the trees.
- Option B is incorrect because the number of observations (data points) should be increased, while the number of features should be decreased in order to prevent models from overfitting.
- Option C is CORRECT because in general, getting more data for training the simplest and best possible way to prevent overfitting, which typically also increases accuracy. Cross validation, a technique of running the training process with several subsets of training data also helps avoiding overfitting.
- Option D is incorrect because the number of features should be decreased in order to prevent models from overfitting. Adding more features results in an adverse effect.

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/concept-manage-ml-pitfalls#best-practices-you-implement
- https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/multiclass-logistic-regression

Question 51

Domain: Implement responsible machine learning

You are working for an agricultural company which monitors its land areas using an IoT solution, including several types of environmental sensors, local weather stations, drones etc. These data sources gather field data continuously and send it to a central location where processing has to be done once a day (preferably in night hours) and predictions have to be generated for the following day. Your task is to build an ML solution to ingest the daily batches of data regularly, pre-process and feed it to an inference service. You decide to use Azure ML pipelines and run them at predefined intervals.

1. Set schedule recurrence
2. Create schedule
3. Enable schedule
4. Publish pipeline
5. Retrieve pipeline id

Which steps should you use in your script in what order?

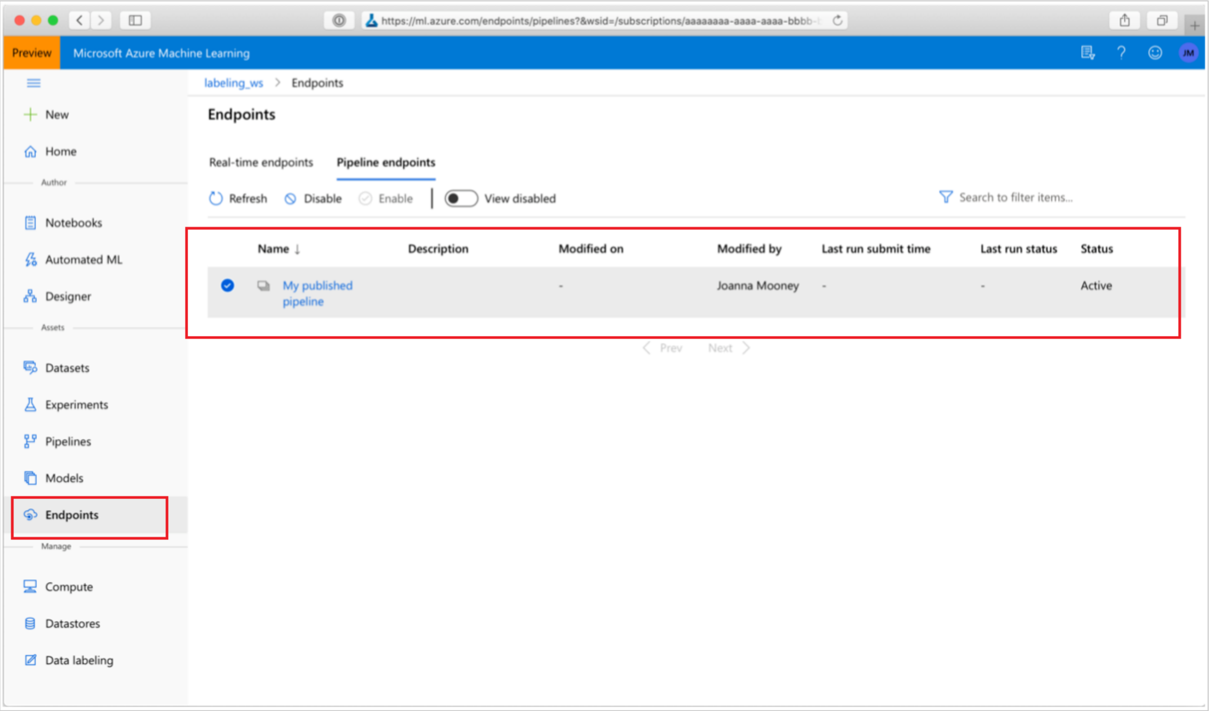
- A. 4, 5, 1, 3
- B. 4, 5, 1, 2
- C. 4, 5, 2, 1
- D. 4, 5, 2, 1, 3

Explanation:

Answer: B

- Option A is incorrect because the schedule must be submitted in order to become active. Once it is submitted, it becomes active automatically. Enabling/disabling have effect only on existing schedules.
- Option B is CORRECT because so that you can initiate the scheduled runs of a pipeline, the pipeline must be created and published first. Then you have to create a ScheduleRecurrence (trigger) object and, finally the Schedule itself, connecting the pipeline (via its id) and the trigger object must be created and run. You can watch the scheduled pipelines in the Endpoints section of the ML Studio.
- Option C is incorrect because the Schedule object connects the pipeline and a triggering event together. Therefore, the defining the schedule recurrence (i.e. the trigger) must precede submitting the schedule.
- Option D is incorrect because the scheduled pipeline gets into ‘Active’ status automatically. There is no need enabling them explicitly.

Reference - Monitor scheduled pipelines



labeling_ws > Endpoints

Endpoints

Real-time endpoints Pipeline endpoints

Name ↓	Description	Modified on	Modified by	Last run submit time	Last run status	Status
My published pipeline		-	Joanna Mooney	-	-	Active

Reference:

- https://docs.microsoft.com/en-us/azure/machine-learning/how-to-schedule-pipelines
- https://github.com/Azure/MachineLearningNotebooks/blob/master/how-to-use-azureml/machine-learning-pipelines/intro-to-pipelines/aml-pipelines-setup-schedule-for-a-published-pipeline.ipynb

Question 52

Domain: Manage Azure resources for machine learning

When creating a workspace <---1---> are also created.

In order to run your <---2---> you need to create <---3--->.

<---4---> help you manage your data during the training process.

You have to define <---5---> to set up the context where scoring of your model takes place.

Which is the right combination of terms to fill the statements above?

- A. compute resources; experiments; environments; datasets;associated resources
- B. compute resources; experiments; runs; datasets; datastores
- C. snapshots; experiments; compute resources; datasets; associated resources
- D. associated resources; experiments; compute resources; datasets; environments

Explanation:

Answer: D

- Option A is incorrect because compute resources (compute instance and compute clusters) need to be created manually, within an existing workspace.
- Option B is incorrect because compute resources (compute instance and compute clusters) need to be created manually, within an existing workspace. Datastores store connection information for accessing training data.
- Option C is incorrect because it is the associated resources which are created together workspaces. Snapshots are zipped folders used while submitting ML runs.
- Option D is CORRECT because creating a workspace also creates certain *other resources* like a storage account automatically; during your machine learning tasks you use *compute resources* to run your *experiments*; it is the *datasets* that make working with data easier; the context where the experiments run is the *environment*.

Reference:

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

Question 53

Domain: Manage Azure resources for machine learning

For your machine learning experiments, you need to get CSV data files from a web location and you need to use them as a dataset in your ML workspace. There are ten files to be imported and each of them contain different columns of a large table. Above the column header, each file has 6 rows containing unstructured data like dates, separator lines etc. You want to use ML Studio to complete the work. Beside others, you should set the following options:

1. Dataset type:
2. Column headers:
3. Skip rows / Skip n rows:

Which combination of settings should you use?

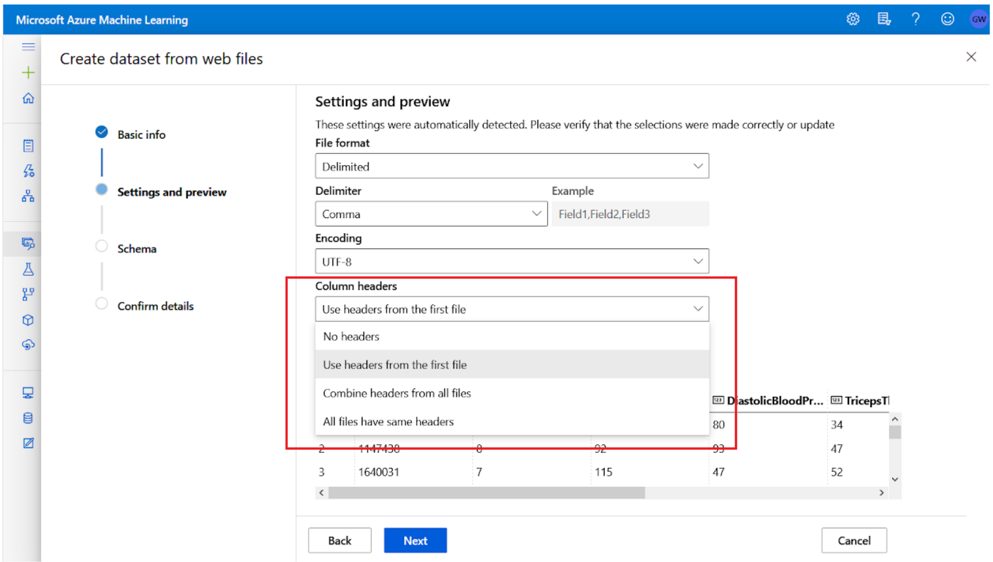
- A. 1 - File; 2 - Combine headers from all files; 3 - From all files / 6
- B. 1 - Tabular; 2 - All files have the same headers; 3 - From all files /6
- C. 1 - Tabular; 2 - Combine headers from all files; 3 - From all files / 6
- D. 1 - File; 2 - Combine headers from all files; 3 - From the first file / 5

Explanation:

Answer: C

- Option A is incorrect because File datasets are designed for unstructured training data, like images etc. For CVS sources, Tabular type should be selected.
- Option B is incorrect because column headers from all files must be selected because, as it is stated, all files hold different columns of the whole data structure.
- Option C is CORRECT because for structured data files, tabular dataset should be defined and, since the files contain vertical slices of a large table, column headers from all files have to be combined. The first relevant row (the column header) is located in row 7, i.e. Skip rows setting is 6.
- Option D is incorrect because File type datasets are used for unstructured data, and column headers from all files must be used.

Diagram - ML Studio



Microsoft Azure Machine Learning
GW

Create dataset from web files

Basic info
Settings and preview
Schema
Confirm details

Settings and preview

These settings were automatically detected. Please verify that the selections were made correctly or update

File format: Delimited

Delimiter: Comma

Example: Field1,Field2,Field3

Encoding: UTF-8

Column headers: Use headers from the first file

No headers
Use headers from the first file
Combine headers from all files
All files have same headers

				DiastolicBloodPr...	TricepsT
				80	34
2	1147438	8	92	93	47
3	1640031	7	115	47	52

Reference:

- https://github.com/MicrosoftLearning/DP100/blob/master/labdocs/Lab01A.md

Question 54

Domain: Manage Azure resources for machine learning

You have a batch inference model, size of around 2 GB. You need to deploy this model for batch inferencing tasks and you need to choose a compute for production use.

Which compute should you choose?

- A. Azure Container Instances
- B. Azure ML Compute Cluster
- C. HDInsight
- D. Local compute

Explanation:

Answer: B

- Option A is incorrect because ACI is recommended for development and testing purposes and it is suitable only for models less than 1 GB in size.
- Option B is CORRECT because Azure compute cluster is a cost-effective way to run experiments that need to handle large volumes of data. It is an option that can provide a containerized environment for the deployed models and it is recommended for service endpoints that need to periodically process batches of data.
- Option C is incorrect because HDInsight is a popular, Apache Spark based big-data platform which can be used as an attached compute for training models. It is not a compute target for inference inference scenarios.
- Option D is incorrect because local computes are suitable for development and testing, but they are not appropriate resources for batch inferencing on the basis of large amounts of data.

Reference:

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

Question 55

Domain: Implement responsible machine learning

Your company is an operator of wind farms in the North Sea. The wind turbines are equipped with several sensors which regularly provide data about the status of the machinery. You are going to introduce a predictive maintenance solution to reduce the failure rate of the machines. As part of the solution, you have trained a machine learning model and deployed it as a web service. The predictive maintenance application use this code to call the service with real-time data:

```
...
# data from turbines
x_new = [[0.8,2.8,4.0,3.0],
        [-0.2,3.8,3.9,2.1],
        [0.3,3.2,3.7,2.3]]

# convert to JSON
json_data = json.dumps({"data": x_new})

# Set the content type in the headers
request_header = { 'Content-Type':'application/json' }

# Call the service

response = requests.post(url = endpoint,
                      data = json_data,
                      headers = request_headers)
...
# Get the predictions from the JSON response
predictions = json.loads(response.json())
```

How can you retrieve the REST endpoint of the deployed service?

- A. service.deploy_configuration()
- B. endpoint = service.location
- C. endpoint = service.scoring_uri
- D. endpoint = service.endpoint

Explanation:

Answer: C

- Option A is incorrect because the deploy_configuration() method creates a deploy configuration for the webservice. It cannot be used to retrieve the endpoint of a deployed service.
- Option B is incorrect because the location attribute of the Webservice contains the Azure region where the Webservice is deployed to. It defaults to the location of the workspace.
- Option C is CORRECT because it is the scoring_uri attribute of the Webservice which contains the REST endpoints URL that has to be used in the post request. A scoring URI looks like this: http://104.214.29.155:80/api/v1/service/srv_turbine_data_prediction/score
- Option D is incorrect because endpoint is not a valid attribute of the Webservice object.

Reference:

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