DUMPSARENA

# AWS Certified Machine Learning Engineer - Associate

## Amazon AWS MLA-C01

**Version Demo**

**Total Demo Questions: 33**

**Total Premium Questions: 334**

**Buy Premium PDF**

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

# Topic Break Down

| Topic | No. of Questions |
|---|---|
| Topic 1, Data Preparation for Machine Learning | 83 |
| Topic 2, ML Model Development | 80 |
| Topic 3, Deployment and Orchestration of ML Workflows | 112 |
| Topic 4, ML Solution Monitoring, Maintenance, and Security | 58 |
| Topic 5, Mix Questions | 1 |
| Total | 334 |

QUESTION NO: 1

A logistics company has installed in-vehicle cameras for basic monitoring of its drivers. The company wants to improve driver safety by identifying distractions that could lead to accidents.

Which solution will meet this requirement with the LEAST operational effort?

**A.** Use Amazon Rekognition eye gaze direction detection to monitor driver behavior and identify distractions.

**B.** Use Amazon SageMaker AI to customize an AI model to monitor driver behavior and identify distractions.

**C.** Integrate a third-party driver monitoring system with Amazon Rekognition to monitor driver behavior and identify distractions.

**D.** Use Amazon Comprehend to analyze text-based driver feedback and identify distractions.

**ANSWER: A**

**Explanation:**

Use Amazon Rekognition eye gaze direction detection to monitor driver behavior and identify distractions. Amazon Rekognition is a fully managed computer vision service, so the company can analyze frames or video-derived images from its existing in-vehicle cameras without building, training, tuning, hosting, or maintaining a custom machine learning model. Its face-detection response includes the `EyeDirection` face-detail attribute, which provides pitch and yaw measurements and a confidence score for the detected eye direction. These measurements can be evaluated over time to identify behavior consistent with visual distraction, such as repeatedly looking substantially away from the road. Rekognition also returns face landmarks and attributes such as eye openness, which can support the implementation of practical driver-monitoring logic using managed API output. This approach minimizes operational overhead because AWS operates the underlying computer vision infrastructure and model, while the company focuses on defining alert thresholds, processing camera frames, and integrating results into its safety workflow. The EyeDirection attribute is documented in the Amazon Rekognition `DetectFaces` API response and face-detail model. See the Amazon Rekognition DetectFaces API Reference and the FaceDetail API Reference.

**QUESTION NO: 2**

A company is developing an application that reads animal descriptions from user prompts and generates images based on the information in the prompts. The application reads a message from an Amazon Simple Queue Service (Amazon SQS) queue. Then the application uses Amazon Titan Image Generator on Amazon Bedrock to generate an image based on the information in the message.

Finally, the application removes the message from SQS queue.

Which IAM permissions should the company assign to the application's IAM role? (Select TWO.)

**A.** Allow the bedrock:InvokeModel action for the Amazon Titan Image Generator resource.

**B.** Allow the bedrock:Get* action for the Amazon Titan Image Generator resource.

**C.** Allow the sqs:ReceiveMessage action and the sqs:DeleteMessage action for the SQS queue resource.

**D.** Allow the sqs:GetQueueAttributes action and the sqs:DeleteMessage action for the SQS queue resource.

**E.** Allow the sagemaker:PutRecord* action for the Amazon Titan Image Generator resource.

**ANSWER: A C**

**Explanation:**

The application must be authorized to invoke Amazon Titan Image Generator and to consume and acknowledge work items from the Amazon SQS queue. `bedrock:InvokeModel` is the IAM action that permits inference requests to an Amazon Bedrock foundation model, including Amazon Titan Image Generator. The policy should scope this permission to the applicable Titan model resource or inference profile as supported by the model invocation method being used.

For the queue-processing workflow, `sqs:ReceiveMessage` permits the application to retrieve messages from the queue. Receiving a message does not permanently remove it; SQS uses a receipt handle to identify that received message. After image generation succeeds, `sqs:DeleteMessage` allows the application to delete the processed message by using that receipt handle. Granting these two SQS actions on the specific queue supports the required receive-process-delete pattern while following least-privilege principles.

See the Amazon Bedrock IAM policy examples for model invocation at Amazon Bedrock User Guide and the required permissions for SQS actions at Amazon SQS Developer Guide.

## QUESTION NO: 3

A company has trained an ML model that is packaged in a container. The company will integrate the model with an existing Python web application. The company needs to host the model on AWS by using Kubernetes.

The company does not want to manage the control plane and must provision the resources in a repeatable manner. The infrastructure must be provisioned by using Python.

Which solution will meet these requirements?

**A.** Use AWS CloudFormation to provision Amazon EC2 instances in multiple Availability Zones. Set up a Kubernetes cluster. Host the model container on the Kubernetes cluster.

**B.** Use the AWS CLI to provision an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Store the image in an Amazon Elastic Container Registry (Amazon ECR) repository. Host the model container on the EKS cluster.

**C.** Use the AWS Cloud Development Kit (AWS CDK) to provision an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Store the image in an Amazon Elastic Container Registry (Amazon ECR) repository. Host the model container on the EKS cluster.

**D.** Use AWS CloudFormation to provision an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Store the image in an Amazon Elastic Container Registry (Amazon ECR) repository. Host the model container on the EKS cluster.

## ANSWER: C

**Explanation:**

Use the AWS Cloud Development Kit (AWS CDK) to provision an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Store the image in an Amazon Elastic Container Registry (Amazon ECR) repository. Host the model container on the EKS cluster. Amazon EKS is a managed Kubernetes service in which AWS manages the Kubernetes control plane, including its availability and scalability. This directly fulfills the requirement to use Kubernetes without the company operating the control plane itself.

AWS CDK supports Python as a first-class language for defining infrastructure as code. The team can therefore describe the EKS cluster, networking, IAM permissions, and related resources in Python source files, synthesize them into AWS CloudFormation templates, and deploy the same configuration consistently across environments. This provides the required repeatable provisioning workflow while aligning with the existing Python expertise. Amazon ECR is AWS's managed container image registry and is an appropriate location for the packaged model image. Kubernetes workloads on EKS can then pull that image from ECR and run it as a deployment or service for integration with the web application.

See the Amazon EKS User Guide and the AWS CDK Python developer guide.

## QUESTION NO: 4

A company needs to run a batch data-processing job on Amazon EC2 instances. The job will run during the weekend and will take 90 minutes to finish running. The processing can handle interruptions. The company will run the job every weekend for the next 6 months.

Which EC2 instance purchasing option will meet these requirements MOST cost-effectively?

**A.** Spot Instances

**B.** Reserved Instances

**C.** On-Demand Instances

**D.** Dedicated Instances

**ANSWER: A**

**Explanation:**

Spot Instances are the most cost-effective choice because the workload is batch-oriented, has a defined but flexible execution schedule, and can tolerate interruption. Amazon EC2 Spot uses unused EC2 capacity and can offer substantial discounts compared with On-Demand pricing. In exchange, AWS can interrupt Spot Instances when it needs the capacity back, making Spot particularly appropriate for fault-tolerant, interruption-resilient processing such as batch jobs, data analysis, and distributed workloads.

The stated ability to handle interruptions is the decisive requirement. The job should be designed to save progress or retry unfinished work when capacity is interrupted. For example, a job scheduler or a service such as AWS Batch can request Spot capacity and reschedule work when an instance is reclaimed. Because the processing runs only for approximately 90 minutes each weekend, the company pays only for the compute time used rather than committing to capacity for the full six-month period. This aligns the purchasing model with intermittent usage while maximizing available cost savings.

AWS documents Spot Instances as suitable for flexible, fault-tolerant workloads and explains their interruption behavior and pricing model in the Amazon EC2 Spot Instances User Guide. See also the Amazon EC2 Spot Instances overview.

**QUESTION NO: 5**

An ML model is deployed in production. The model has performed well and has met its metric thresholds for months.

An ML engineer who is monitoring the model observes a sudden degradation. The performance metrics of the model are now below the thresholds.

What could be the cause of the performance degradation?

**A.** Lack of training data

**B.** Drift in production data distribution

**C.** Compute resource constraints

**D.** Model overfitting

**ANSWER: B**

**Explanation:**

Drift in production data distribution is the most likely cause when a model that has operated successfully for months suddenly falls below its established performance thresholds. Data drift occurs when the statistical properties of live inference inputs differ from those of the data used to train or baseline the model. For example, customer behavior may change, a new product category may be introduced, seasonal effects may become significant, or an upstream process may alter the format, range, or frequency of feature values. As these changes accumulate, the relationships learned during training can become less representative of the production environment, reducing prediction quality.

In Amazon SageMaker, Model Monitor can establish baseline constraints from training data and continuously compare production endpoint data against those baselines. It can identify data-quality violations, including shifts in feature distributions, and publish monitoring results that can trigger investigation or a retraining workflow. Monitoring both input data quality and model-quality metrics helps distinguish a degraded model from changing real-world conditions and supports prompt remediation through updated data collection, feature validation, and retraining. See the Amazon SageMaker Model Monitor documentation and data quality monitoring documentation.

**QUESTION NO: 6**

A construction company is using Amazon SageMaker AI to train specialized custom object detection models to identify road damage. The company uses images from multiple cameras. The images are stored as JPEG objects in an Amazon S3 bucket.

The images need to be pre-processed by using computationally intensive computer vision techniques before the images can be used in the training job. The company needs to optimize data loading and pre-processing in the training job. The solution cannot affect model performance or increase compute or storage resources.

Which solution will meet these requirements?

**A.** Use SageMaker AI file mode to load and process the images in batches.

**B.** Reduce the batch size of the model and increase the number of pre-processing threads.

**C.** Reduce the quality of the training images in the S3 bucket.

**D.** Convert the images into RecordIO format and use the lazy loading pattern.

**ANSWER: D**

**Explanation:**

Convert the images into RecordIO format and use the lazy loading pattern. RecordIO packages many image records into larger sequentially readable files, avoiding the high per-object request overhead that occurs when a training job repeatedly accesses a large number of individual JPEG objects in Amazon S3. This improves input throughput and lets the training workload spend less time waiting on data retrieval.

Lazy loading complements this format by reading records only when they are needed by the training pipeline rather than loading the full dataset into memory at startup. The computer vision transformations can therefore be performed as records are consumed, with memory and input processing used efficiently throughout training. This approach improves the data pipeline without reducing image fidelity, changing the model's training behavior, or requiring larger training instances or additional persistent storage capacity. RecordIO is a supported input format for SageMaker AI image workloads, and its use is intended to provide efficient image data ingestion. AWS documents RecordIO support for image training at SageMaker AI Image Classification. The RecordIO file format is also documented by the MXNet project at MXNet RecordIO FAQ.

## QUESTION NO: 7

An ML engineer is designing an AI-powered traffic management system. The system must use near real-time inference to predict congestion and prevent collisions.

The system must also use batch processing to perform historical analysis of predictions over several hours to improve the model. The inference endpoints must scale automatically to meet demand.

Which combination of solutions will meet these requirements? (Select TWO.)

**A.** Use Amazon SageMaker real-time inference endpoints with automatic scaling based on ConcurrentInvocationsPerInstance.

**B.** Use AWS Lambda with reserved concurrency and SnapStart to connect to SageMaker endpoints.

**C.** Use an Amazon SageMaker Processing job for batch historical analysis. Schedule the job with Amazon EventBridge.

**D.** Use Amazon EC2 Auto Scaling to host containers for batch analysis.

**E.** Use AWS Lambda for historical analysis.

**ANSWER: A C**

**Explanation:**

Use Amazon SageMaker real-time inference endpoints with automatic scaling based on ConcurrentInvocationsPerInstance. A SageMaker real-time endpoint is designed to serve low-latency, synchronous inference requests, which fits a traffic-management application that needs predictions as events occur. SageMaker endpoint autoscaling uses Application Auto

Scaling to adjust the number of endpoint instances in response to demand. In particular, the `SageMakerVariantInvocationsPerInstance` predefined metric corresponds to the `ConcurrentInvocationsPerInstance` metric for applicable endpoint types, enabling target-tracking policies that maintain capacity as traffic changes. This provides managed, elastic online inference without operating the serving fleet directly.

Use an Amazon SageMaker Processing job for batch historical analysis. Processing jobs run user-provided processing code in managed compute infrastructure and are appropriate for data-processing and model-evaluation workloads that analyze accumulated prediction data over an extended period. An EventBridge schedule can start the processing workflow on a defined cadence, such as every few hours, so the historical analysis occurs automatically. Together, managed real-time endpoints provide responsive operational predictions, while scheduled Processing jobs provide the recurring batch analysis needed to generate insights and improve the model. See the SageMaker endpoint autoscaling documentation and the SageMaker Processing jobs documentation.

## QUESTION NO: 8

A company is creating an Amazon SageMaker Pipeline that trains and evaluates a binary classification model. The pipeline must register a model only when its F1 score meets a specified minimum value.

Which actions should the company take to meet these requirements? (Choose two.)

**A.** Use a SageMaker Processing step to calculate the F1 score and produce an evaluation report.

**B.** Use a Condition step to compare the F1 score with the required threshold before model registration.

**C.** Use an Amazon CloudWatch alarm to change the model package approval status.

**D.** Use an Amazon SageMaker Debugger rule to register the model after training.

**E.** Use an AWS Lambda function to replace the evaluation dataset after model registration.

## ANSWER: A B

**Explanation:**

Use a SageMaker Processing step to evaluate the trained model and write the F1 score to an evaluation report. A processing job can load the model and evaluation dataset, calculate the required metric, and produce a JSON property file in Amazon S3 for downstream pipeline steps.

Use a Condition step to compare the F1 score in the evaluation report with the required threshold. The condition can route execution to a RegisterModel step only when the score meets the threshold. If the condition is not met, the pipeline can end or route to an alternate remediation path. This creates an automated and repeatable model-quality gate. See the Amazon SageMaker Pipeline steps documentation and the SageMaker Pipelines condition step documentation.

## QUESTION NO: 9

An ML engineer is training a simple neural network model. The ML engineer tracks the performance of the model over time on a validation dataset. The model's performance improves substantially at first and then degrades after a specific number of epochs.

Which solutions will mitigate this problem? (Choose two.)

**A.** Enable early stopping on the model.

**B.** Increase dropout in the layers.

**C.** Increase the number of layers.

**D.** Increase the number of neurons.

**E.** Investigate and reduce the sources of model bias.

**ANSWER: A B**

**Explanation:**

The pattern described is overfitting: training continues beyond the point at which the model has learned patterns that generalize well to the held-out validation data. **Enable early stopping on the model.** is appropriate because early stopping monitors a chosen validation metric, such as validation loss or validation accuracy, during training. Once that metric no longer improves for a configured number of epochs, training can stop and the weights from the best validation epoch can be retained. This prevents unnecessary additional training after generalization has peaked. TensorFlow documents the callback behavior, including the `patience` setting and restoration of the best weights, at TensorFlow EarlyStopping.

**Increase dropout in the layers.** is also a valid regularization measure. During training, dropout randomly sets a configured fraction of layer inputs to zero. This reduces reliance on individual neurons and co-adapted features, encouraging the network to learn more robust representations that generalize better to validation data. Dropout is active during training and is normally inactive during inference, so predictions use the full learned network. The implementation and training behavior are described in the TensorFlow Dropout layer documentation.

## QUESTION NO: 10

An ML engineer is developing a loan approval model. The company needs to assess whether the training dataset contains bias before training and whether the trained model produces biased predictions after training.

Which SageMaker capabilities should the ML engineer use? (Choose two.)

**A.** Use SageMaker Clarify pretraining bias analysis on the training dataset.

**B.** Use SageMaker Clarify post-training bias analysis on the model predictions.

**C.** Use SageMaker Inference Recommender to calculate bias metrics.

**D.** Use SageMaker Debugger to identify demographic bias in the dataset.

**E.** Use Amazon Textract to calculate loan approval fairness metrics.

**ANSWER: A B**

**Explanation:**

Use Amazon SageMaker Clarify pretraining bias analysis to evaluate potential bias in the training dataset. Pretraining bias metrics assess whether labels or feature distributions differ across specified facets, such as demographic groups, before a model is trained. This helps identify data issues that could influence model behavior.

Use Amazon SageMaker Clarify post-training bias analysis to evaluate bias in model predictions. Post-training metrics evaluate predictions in relation to outcomes and facet groups, allowing the engineer to identify disparities introduced or amplified by the trained model.

SageMaker Model Monitor can monitor data and model quality in production, but it does not replace Clarify bias analysis for the stated pretraining and post-training fairness assessments. See the SageMaker Clarify post-training bias documentation and the SageMaker Clarify pretraining bias documentation.

## QUESTION NO: 11

A hospital is using an ML model to validate x-ray results. The hospital runs a nightly batch inference job. The hospital needs to produce a daily report about model data quality and model performance.

Which solution will meet these requirements?

**A.** Schedule a monitoring job in Amazon SageMaker Model Monitor. Generate the monitoring results for the model and data.

**B.** Create an Amazon CloudWatch dashboard that includes the metrics for processing steps in the nightly batch inference job. Compare the baseline resource metrics. Share the dashboard link.

**C.** Use AWS Glue DataBrew to create a custom recipe job that uses the Numerical Statistics data quality check for the model file. Generate the results.

**D.** Create a SageMaker AI pipeline that includes a QualityCheck step to run monitoring jobs. Generate the monitoring results for the model and the data.

**ANSWER: A**

**Explanation:**

Schedule a monitoring job in Amazon SageMaker Model Monitor. Generate the monitoring results for the model and data. SageMaker Model Monitor is designed specifically to evaluate production ML workloads for both data quality and model quality. A data quality monitoring job compares the input data used for inference with a baseline and produces a constraints-violation report when feature distributions, missing values, or other characteristics drift beyond the defined thresholds. This makes it appropriate for detecting changes in the nightly x-ray inference input data.

For model performance, SageMaker Model Monitor can run model quality monitoring when ground-truth labels become available. It compares predictions with those labels and calculates model-quality metrics, enabling the hospital to track whether predictive performance is meeting expectations over time. Monitoring jobs can be scheduled to align with the nightly batch inference cadence, and their reports are written to Amazon S3 for daily review, retention, or downstream reporting workflows. This directly addresses the need for recurring reports covering both incoming-data quality and model performance rather than only infrastructure or job-execution metrics. See the Amazon SageMaker Model Monitor documentation and the model quality monitoring guide.

**QUESTION NO: 12**

An ML company wants to monitor and analyze the API calls that its AWS resources make. The company has created an AWS CloudTrail log file that logs to an Amazon S3 bucket. The company has also created an organization in AWS Organizations to manage permissions across accounts.

The company needs to enable log file validation to ensure the integrity of its log files.

Which solution will meet these requirements?

**A.** Enable CloudTrail log file integrity validation.

**B.** Create a multi-Region trail in CloudTrail.

**C.** Create a trail in CloudTrail for the organization.

**D.** Enable Amazon CloudWatch Logs delivery.

**ANSWER: A**

**Explanation:**

Enable CloudTrail log file integrity validation. This CloudTrail feature is specifically designed to provide verifiable evidence that delivered CloudTrail log files have not been modified, deleted, or otherwise altered after CloudTrail generated them. When validation is enabled on a trail, CloudTrail periodically creates digest files and stores them in the trail's Amazon S3 bucket. These digest files contain cryptographic hashes that form a signed chain, allowing the company to use the AWS CLI `validate-logs` command to verify the integrity of log files over a specified time range. The validation process can identify whether log files are intact and whether any expected files are missing.

The trail already delivers API activity records to Amazon S3, which is the required destination for CloudTrail log file validation. AWS Organizations can govern accounts and support centralized audit practices, but the requirement here is specifically cryptographic validation of the delivered CloudTrail records. Enabling this feature directly fulfills that integrity requirement and supports reliable security investigations, audit evidence, and compliance controls across the company's AWS environment.

For implementation details, see the AWS CloudTrail log file integrity validation documentation and the AWS CLI validate-logs command reference.

## QUESTION NO: 13

Case Study

A company is building a web-based AI application by using Amazon SageMaker. The application will provide the following capabilities and features: ML experimentation, training, a

central model registry, model deployment, and model monitoring.

The application must ensure secure and isolated use of training data during the ML lifecycle. The training data is stored in Amazon S3.

The company needs to use the central model registry to manage different versions of models in the application.

Which action will meet this requirement with the LEAST operational overhead?

**A.** Create a separate Amazon Elastic Container Registry (Amazon ECR) repository for each model.

**B.** Use Amazon Elastic Container Registry (Amazon ECR) and unique tags for each model version.

**C.** Use the SageMaker Model Registry and model groups to catalog the models.

**D.** Use the SageMaker Model Registry and unique tags for each model version.

## ANSWER: C

**Explanation:**

Use the SageMaker Model Registry and model groups to catalog the models. Amazon SageMaker Model Registry is a managed SageMaker capability designed specifically to centralize the cataloging, versioning, approval, and lifecycle management of ML model artifacts. A model package group represents a collection of related models, such as all iterations of a model for a particular use case. Each registered model package in that group is automatically assigned a version, allowing teams to maintain a reliable history of model versions without building and operating custom version-management processes.

The registry also stores model-related metadata, including inference specifications, model metrics, lineage information, and approval status. This gives the organization a consistent governance point for deciding which version is suitable for deployment. Model packages can progress through approval states, and approved versions can be used directly in SageMaker deployment workflows. Because this functionality is natively managed by SageMaker and integrates with training, pipelines, deployment, and monitoring, it provides the requested central model registry with minimal operational effort. For implementation details, see the Amazon SageMaker Model Registry documentation and the Amazon SageMaker model package group documentation.

## QUESTION NO: 14 - (HOTSPOT)

A company wants to host an ML model on Amazon SageMaker. An ML engineer is configuring a continuous integration and continuous delivery (CI/CD) pipeline in AWS CodePipeline to deploy the model. The pipeline must run automatically when new training data for the model is uploaded to an Amazon S3 bucket.

Select and order the pipeline ' s correct steps from the following list. Each step should be selected one time or not at all. (Select and order three.)

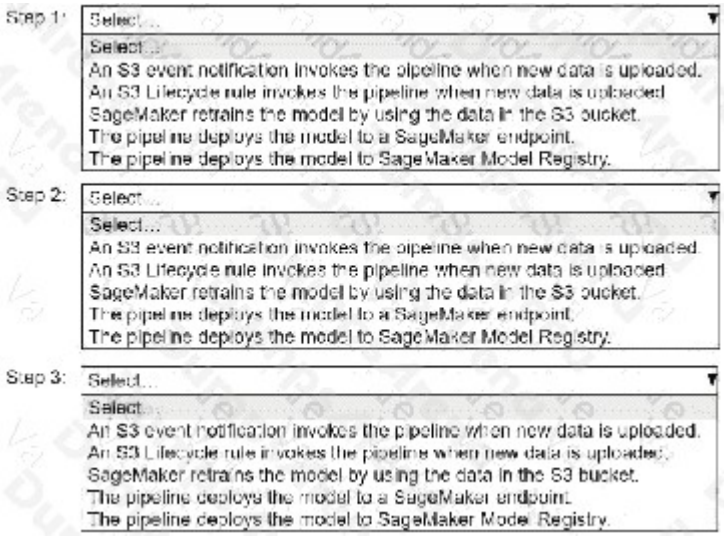
â€¢ An S3 event notification invokes the pipeline when new data is uploaded.

â€¢ S3 Lifecycle rule invokes the pipeline when new data is uploaded.

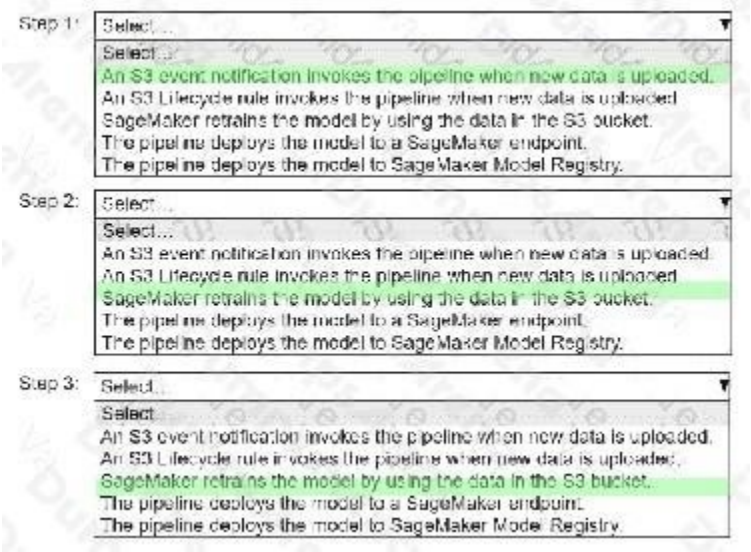
â€¢ SageMaker retrains the model by using the data in the S3 bucket.

â€¢ The pipeline deploys the model to a SageMaker endpoint.

â€¢ The pipeline deploys the model to SageMaker Model Registry.




**ANSWER:**




**Explanation:**

The correct workflow starts when an S3 object-created event occurs for newly uploaded training data. Amazon S3 event notifications are designed to react to object activity and can initiate downstream automation, including an event-driven process that starts a CodePipeline workflow. This makes the pipeline run automatically whenever new data arrives, rather than relying on a scheduled process. AWS documents the available S3 event types, including object-created events, in the Amazon S3 Event Notifications guide.

After the pipeline is triggered, SageMaker retrains the model by reading the updated data stored in the S3 bucket. SageMaker training jobs use training datasets from S3 and produce model artifacts that represent the newly trained version of the model. Incorporating this training stage into the delivery workflow ensures that each qualifying data update can produce a refreshed model artifact. The SageMaker training documentation describes how training jobs use input data and generate model artifacts.

Finally, the pipeline deploys the retrained model to a SageMaker endpoint. A SageMaker endpoint hosts the model behind a managed HTTPS inference service, allowing applications to submit real-time inference requests to the updated model. Performing this deployment as the final pipeline stage creates an automated path from new training data to a serving model. SageMaker endpoint deployment behavior is described in the SageMaker real-time endpoints documentation. Thus, the required order is event notification, retraining, then endpoint deployment.

**QUESTION NO: 15**

A company is planning to create several ML prediction models. The training data is stored in Amazon S3. The entire dataset is more than 5 TB in size and consists of CSV, JSON, Apache Parquet, and simple text files.

The data must be processed in several consecutive steps. The steps include complex manipulations that can take hours to finish running. Some of the processing involves natural language processing (NLP) transformations. The entire process must be automated.

Which solution will meet these requirements?

**A.** Process data at each step by using Amazon SageMaker Data Wrangler. Automate the process by using Data Wrangler jobs.

**B.** Use Amazon SageMaker notebooks for each data processing step. Automate the process by using Amazon EventBridge.

**C.** Process data at each step by using AWS Lambda functions. Automate the process by using AWS Step Functions and Amazon EventBridge.

**D.** Use Amazon SageMaker Pipelines to create a pipeline of data processing steps. Automate the pipeline by using Amazon EventBridge.

**ANSWER: D**

**Explanation:**

Use Amazon SageMaker Pipelines to create a pipeline of data processing steps. Automate the pipeline by using Amazon EventBridge. SageMaker Pipelines is designed to orchestrate repeatable ML workflows composed of ordered, dependent steps. In this case, each transformation stage can be implemented as a SageMaker Processing step, which runs managed processing jobs suitable for long-running, compute-intensive data preparation workloads. Processing jobs can read input data from Amazon S3 and write transformed artifacts back to S3, allowing later steps to consume the preceding step's output. This supports a scalable multi-stage workflow rather than requiring the complete transformation to fit into a short-lived execution environment.

The processing container can include the libraries and code required for complex structured-data transformations and NLP operations. Pipeline step dependencies ensure that consecutive steps run in the required sequence, while SageMaker records execution status, inputs, outputs, and lineage for each pipeline execution. Amazon EventBridge can schedule a pipeline execution or invoke it in response to relevant events, providing the required end-to-end automation. This architecture separates orchestration from processing compute and is appropriate for large S3-resident datasets and workloads that can run for hours. See the Amazon SageMaker Pipelines documentation and the SageMaker Processing documentation.

**QUESTION NO: 16**

A company wants to improve the sustainability of its ML operations.

Which actions will reduce the energy usage and computational resources that are associated with the company's training jobs? (Choose two.)

**A.** Use Amazon SageMaker Debugger to stop training jobs when non-converging conditions are detected.

**B.** Use Amazon SageMaker Ground Truth for data labeling.

**C.** Deploy models by using AWS Lambda functions.

**D.** Use AWS Trainium instances for training.

**E.** Use PyTorch or TensorFlow with the distributed training option.

**ANSWER: A D**

**Explanation:**

Using Amazon SageMaker Debugger to stop training jobs when non-converging conditions are detected reduces waste from jobs that are unlikely to produce a useful model. Debugger can monitor tensors and training metrics through rules that detect

conditions such as vanishing or exploding gradients, overfitting, or lack of progress. A rule action can stop a job when such a condition is identified. Ending an unproductive training run early avoids consuming additional accelerator, CPU, memory, and storage resources, which also reduces the energy used by the workload.

Using AWS Trainium instances for training is also an energy- and resource-efficiency measure because Trainium is purpose-built for deep learning training. Amazon EC2 Trn instances use AWS Trainium accelerators and are designed to provide high training performance and price performance for supported ML frameworks and models. Completing the required training work more efficiently reduces the compute time and infrastructure capacity needed for a training workload. These actions directly address the resources consumed during model training and support more sustainable ML operations. See the Amazon SageMaker Debugger documentation and the Amazon EC2 Trn1 instance documentation.

## QUESTION NO: 17

An ML engineer has developed a binary classification model outside of Amazon SageMaker. The ML engineer needs to make the model accessible to a SageMaker Canvas user for additional tuning.

The model artifacts are stored in an Amazon S3 bucket. The ML engineer and the Canvas user are part of the same SageMaker domain.

Which combination of requirements must be met so that the ML engineer can share the model with the Canvas user? (Choose two.)

**A.** The ML engineer and the Canvas user must be in separate SageMaker domains.

**B.** The Canvas user must have permissions to access the S3 bucket where the model artifacts are stored.

**C.** The model must be registered in the SageMaker Model Registry.

**D.** The ML engineer must host the model on AWS Marketplace.

**E.** The ML engineer must deploy the model to a SageMaker endpoint.

## ANSWER: B C

**Explanation:**

The Canvas user must have permissions to access the S3 bucket where the model artifacts are stored because SageMaker Canvas needs access, through the user's configured IAM permissions and execution role, to retrieve the model artifact files. The applicable permissions must allow access to the relevant bucket and object prefix, and the permissions must also accommodate any applicable bucket policy or AWS KMS key policy when the artifacts are encrypted. This access is necessary for Canvas to use the shared model data.

The model must be registered in the SageMaker Model Registry because the registry provides the SageMaker-native mechanism for organizing, governing, and sharing model versions. A registered model package contains the model's artifact location and associated metadata, allowing an eligible Canvas user in the same SageMaker domain to discover and import the model for use in Canvas. Model Registry also supports lifecycle management and controlled access to registered model packages, rather than requiring the engineer to distribute artifacts manually.

A SageMaker endpoint is used to serve inference requests and is not needed merely to make a model available for Canvas tuning. Similarly, Marketplace publication is unrelated to internal sharing within a domain. See the Amazon SageMaker Canvas documentation for bringing your own models and the SageMaker Model Registry documentation.

## QUESTION NO: 18 - (HOTSPOT)

A company uses a training job on Amazon SageMaker AI to train a neural network. The job first trains a model and then evaluates the model's performance ag

test dataset. The company uses the results from the evaluation phase to decide if the trained model will go to production.

The training phase takes too long. The company needs solutions that can shorten training time without decreasing the model's final performance.

Select the correct solutions from the following list to meet the requirements for each description. Select each solution one time or not at all. (Select THREE.)

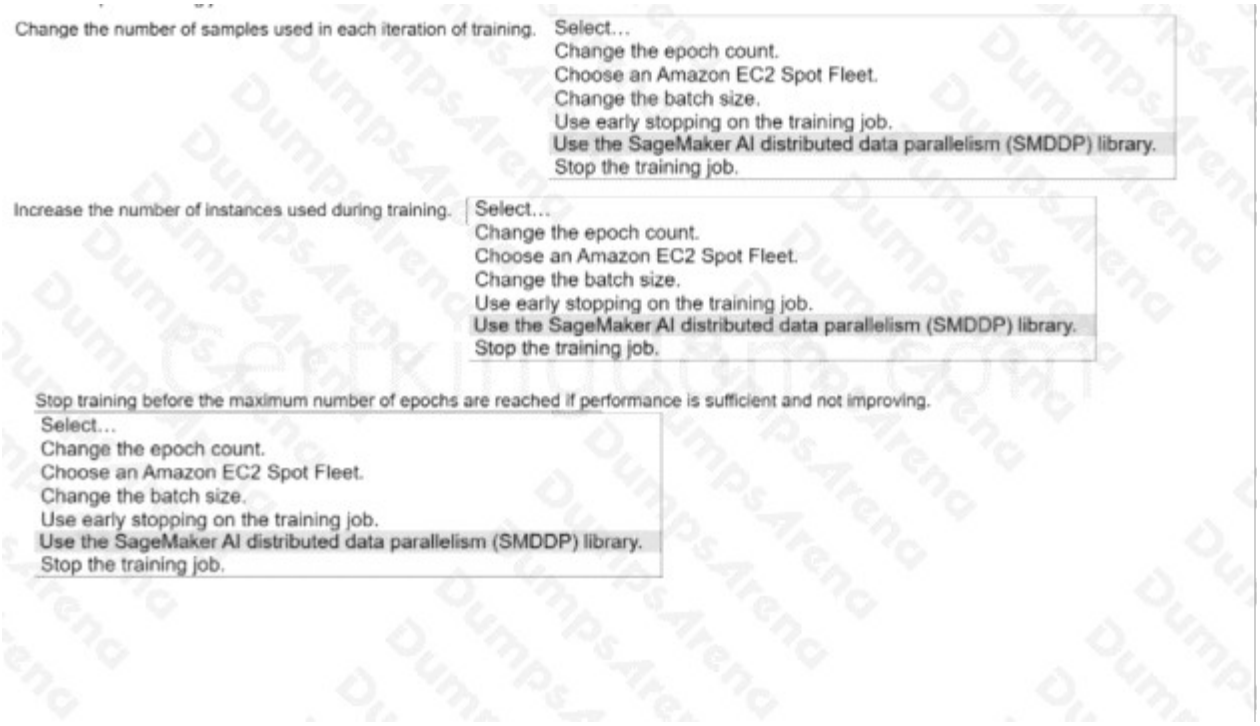
. Change the epoch count.

. Choose an Amazon EC2 Spot Fleet. Â· Change the batch size.

. Use early stopping on the training job.

Â· Use the SageMaker AI distributed data parallelism (SMDDP) library.

. Stop the training job.




**ANSWER:**

**Explanation:**

The correct selections address three distinct ways to reduce the elapsed time of neural-network training while retaining the ability to reach the required model quality. “Change the batch size” is the setting that controls the number of training samples processed in one iteration. Selecting an appropriate larger batch size can improve accelerator utilization and throughput, reducing the time needed to process an epoch. Batch size should be tuned with the available GPU memory and the training configuration so that convergence and validation quality remain acceptable. Amazon SageMaker guidance discusses batch size as an important training-performance consideration: Amazon SageMaker model training considerations.

“Use the SageMaker AI distributed data parallelism (SMDDP) library” is the correct mechanism for scaling training across multiple instances. The library partitions data across workers and synchronizes gradients efficiently during distributed training. This allows the same neural network to train with greater aggregate compute capacity and higher throughput, shortening wall-clock training time while maintaining the intended training process and model objective. AWS documents its use for distributed deep-learning workloads at Distributed data parallel training in Amazon SageMaker.

“Use early stopping on the training job” correctly handles the requirement to end training before the maximum epoch limit when validation performance is already sufficient and has stopped improving. Early stopping monitors an evaluation metric and stops further epochs after improvement plateaus according to configured criteria. This avoids spending compute on epochs that no longer provide useful gains, while retaining the best model state identified through validation. SageMaker explains this capability in its early stopping documentation.

## QUESTION NO: 19

An ML engineer develops a neural network model to predict whether customers will continue to subscribe to a service. The model performs well on training data. However, the accuracy of the model decreases significantly on evaluation data.

The ML engineer must resolve the model performance issue. Which solution will meet this requirement?

**A.** Penalize large weights by using L1 or L2 regularization.

**B.** Remove dropout layers from the neural network.

**C.** Train the model for longer by increasing the number of epochs.

**D.** Capture complex patterns by increasing the number of layers.

## ANSWER: A

**Explanation:**

Penalize large weights by using L1 or L2 regularization. is the appropriate solution because strong training performance combined with substantially lower evaluation accuracy is a characteristic pattern of overfitting. The neural network has likely learned training-set-specific noise or overly detailed relationships that do not generalize to previously unseen customer records. Regularization modifies the training objective by adding a penalty associated with model weight magnitude, encouraging a simpler model with less reliance on any individual learned parameter. L2 regularization, often called weight decay, penalizes the squared magnitude of weights and generally produces smoother parameter values. L1 regularization penalizes absolute weight magnitude and can drive some weights to zero, reducing the effective complexity of the model. Both techniques constrain the network and help it learn patterns that are more likely to persist in evaluation and production data. The regularization strength should be tuned using validation results so that the model is sufficiently constrained without becoming too simple. This is a standard approach for improving generalization when a model exhibits overfitting. See the TensorFlow guide to overfitting and underfitting and the Google Machine Learning Crash Course discussion of regularization.

## QUESTION NO: 20 - (SIMULATION)

An ML engineer is working on an ML model to predict the prices of similarly sized homes. The model will base predictions on several features The ML engineer will use the following feature engineering techniques to estimate the prices of the homes:

• Feature splitting

• Logarithmic transformation

• One-hot encoding

• Standardized distribution

Select the correct feature engineering techniques for the following list of features. Each feature engineering technique should be selected one time or not at all (Select three.)

**ANSWER: See the explanation for the answer**

**Explanation:**

Select these three feature-engineering assignments:

**City (name) → One-hot encoding**
**Type_year (type of home and year the home was built) → Feature splitting**
**Size of the building (square feet or square meters) → Standardized distribution**

Do **not** select logarithmic transformation.

City is categorical and must be converted into numeric indicator columns. `Type_year` combines two independent attributes, so it should be separated into home type and construction year. Building size is numeric and should be standardized so that its scale does not disproportionately influence the model.

## QUESTION NO: 21 - (SIMULATION)

A company has built more than 50 models and deployed the models on Amazon SageMaker AI as real-time inference endpoints. The company needs to reduce the costs of the SageMaker AI inference endpoints. The company used the same ML framework to build the models. The company ' s customers require low-latency access to the models.

Select and order the correct steps from the following list to reduce the cost of inference and keep latency low. Select each step one time or not at all. (Select and order FIVE.)

· Create an endpoint configuration that references a multi-model container.

. Create a SageMaker AI model with multi-model endpoints enabled.

. Deploy a real-time inference endpoint by using the endpoint configuration.

. Deploy a serverless inference endpoint configuration by using the endpoint configuration.

· Spread the existing models to multiple different Amazon S3 bucket paths.

. Upload the existing models to the same Amazon S3 bucket path.

. Update the models to use the new endpoint ID. Pass the model IDs to the new endpoint.

**ANSWER: See the explanation for the answer**

**Explanation:**

**Select and order these five steps:**

1. **Upload the existing models to the same Amazon S3 bucket path.**
   Store the model artifacts under one S3 prefix so the multi-model endpoint can locate and dynamically load the requested model.
2. **Create a SageMaker AI model with multi-model endpoints enabled.**
   Use the shared ML framework container configured for multi-model endpoint (MME) operation.
3. **Create an endpoint configuration that references a multi-model container.**
   Configure the endpoint instance type and capacity for the MME model/container.
4. **Deploy a real-time inference endpoint by using the endpoint configuration.**
   A real-time endpoint provides the required low-latency inference. Do not select serverless inference for this requirement.

5. **Update the models to use the new endpoint ID. Pass the model IDs to the new endpoint.**
   The application sends the requested model identifier (the target model) with each request so SageMaker AI loads/routes to the appropriate model artifact.

Multi-model endpoints reduce cost by hosting many models that use the same framework on shared endpoint instances, rather than maintaining a separate real-time endpoint for each of the 50+ models.

## QUESTION NO: 22

An ML engineer is designing an AI-powered traffic management system. The system must use near real-time inference to predict congestion and prevent collisions.

The system must also use batch processing to perform historical analysis of predictions over several hours to improve the model. The inference endpoints must scale automatically to meet demand.

Which combination of solutions will meet these requirements? (Select TWO.)

**A.** Use Amazon SageMaker real-time inference endpoints with automatic scaling based on SageMakerVariantInvocationsPerInstance.

**B.** Use AWS Lambda with reserved concurrency and SnapStart to connect to SageMaker endpoints.

**C.** Use an Amazon SageMaker Processing job for batch historical analysis. Schedule the job with Amazon EventBridge.

**D.** Use Amazon EC2 Auto Scaling to host containers for batch analysis.

**E.** Use AWS Lambda for historical analysis.

## ANSWER: A C

**Explanation:**

Use Amazon SageMaker real-time inference endpoints with automatic scaling based on SageMakerVariantInvocationsPerInstance because real-time endpoints are designed to serve low-latency synchronous predictions. Application Auto Scaling can adjust the number of endpoint instances in response to a target-tracking policy using the SageMakerVariantInvocationsPerInstance metric. This allows the traffic-management application to retain responsive inference as demand varies, without manually provisioning endpoint capacity. The endpoint can therefore support near real-time congestion and collision prediction while scaling out and in according to observed request volume.

Use an Amazon SageMaker Processing job for batch historical analysis. Schedule the job with Amazon EventBridge because SageMaker Processing provides managed, on-demand compute for batch data-processing and analysis workloads. A processing job can examine prediction records accumulated over multiple hours, calculate historical metrics, and produce datasets or artifacts that support subsequent model-improvement workflows. Amazon EventBridge Scheduler or scheduled EventBridge rules can initiate the job on a defined cadence, providing a serverless operational mechanism for recurring analysis. This combination separates latency-sensitive online inference from longer-running offline analysis while using managed SageMaker capabilities for both workloads.

For implementation details, see the Amazon SageMaker real-time inference documentation and the SageMaker Processing documentation.

## QUESTION NO: 23

An ML engineer is preparing a dataset that contains medical records to train an ML model to predict the likelihood of patients developing diseases.

The dataset contains columns for patient ID, age, medical conditions, test results, and a "Disease" target column.

How should the ML engineer configure the data to train the model?

**A.** Remove the patient ID column.

**B.** Remove the age column.

**C.** Remove the medical conditions and test results columns.

**D.** Remove the "Disease" target column.

**ANSWER: A**

**Explanation:**

**Remove the patient ID column.** A patient ID is an identifier rather than a clinical characteristic that is expected to have a meaningful, generalizable relationship with the likelihood of disease. Including a unique or near-unique identifier as a model feature can allow the model to memorize training examples or learn accidental correlations specific to the training population. This can produce overly optimistic offline evaluation results while reducing performance for previously unseen patients in production.

For this supervised prediction problem, age, medical conditions, and test results should remain available as candidate input features because they describe patient characteristics and clinical observations that can support disease-risk prediction. The *Disease* column must also remain in the training dataset as the target label: the training process uses its known values to learn the relationship between the input features and the outcome to be predicted. The ID can still be retained separately for data governance, traceability, joining records, and associating predictions with the appropriate patient, but it should not be passed to the model as a predictive feature.

AWS SageMaker guidance emphasizes carefully preparing data and selecting useful features before model training. See the Amazon SageMaker best practices documentation and the AWS Well-Architected Machine Learning Lens.

**QUESTION NO: 24**

An ML engineer uses Amazon Athena to query training data in Amazon S3. New data arrives every day in prefixes that follow the format `s3://example-bucket/training/year=YYYY/month=MM/day=DD/`.

The ML engineer needs Athena queries to include new daily partitions without running partition-repair commands or updating the AWS Glue Data Catalog for each new day.

Which actions will meet these requirements? (Choose two.)

**A.** Enable partition projection for the Athena table.

**B.** Define partition projection properties and a storage location template that matches the S3 prefixes.

**C.** Run the MSCK REPAIR TABLE command after each daily data upload.

**D.** Create an AWS Glue crawler to run after each daily data upload.

**E.** Copy each daily partition into an Amazon Redshift table.

**ANSWER: A B**

**Explanation:**

Enable partition projection for the Athena table. Partition projection causes Athena to calculate partition locations and values at query time instead of requiring every partition to be registered in the AWS Glue Data Catalog. This reduces operational work for regularly structured partition layouts.

Define partition projection properties and a storage location template that matches the S3 prefix structure. The projection configuration must describe the valid year, month, and day values and map them to the expected S3 location template. Athena can then determine which partitions to read when queries contain partition predicates. See the Amazon Athena partition projection documentation.

**QUESTION NO: 25**

A company is developing a Retrieval Augmented Generation (RAG) application by using Amazon Bedrock. Source documents are stored in an Amazon S3 bucket and are updated frequently.

The company needs a managed solution that generates embeddings, stores vectors, and retrieves relevant document chunks for the application. The company wants to minimize custom code and infrastructure management.

Which actions will meet these requirements? (Choose two.)

**A.** Create an Amazon Bedrock knowledge base and configure the S3 bucket as a data source.

**B.** Configure Amazon OpenSearch Serverless as the vector store for the knowledge base.

**C.** Use Amazon S3 Select to perform semantic similarity searches on the documents.

**D.** Use Amazon Comprehend entity detection as the vector store.

**E.** Store document embeddings as Amazon CloudWatch metrics.

**ANSWER: A B**

**Explanation:**

Create an Amazon Bedrock knowledge base and configure Amazon S3 as the data source. Knowledge Bases for Amazon Bedrock can ingest supported documents from S3, split documents into chunks, generate embeddings by using a selected embedding model, and make relevant retrieved content available to the application.

Configure Amazon OpenSearch Serverless as the vector store for the knowledge base. OpenSearch Serverless provides a managed vector-search capability and avoids provisioning and operating OpenSearch cluster instances. Amazon Bedrock knowledge bases support OpenSearch Serverless as a vector store option.

Amazon S3 by itself is object storage and does not perform vector similarity search. Amazon Comprehend can extract insights from text but is not a managed vector database for RAG retrieval. See the Knowledge Bases for Amazon Bedrock documentation and the knowledge base setup documentation.

## QUESTION NO: 26

A company has developed a new ML model. The company requires online model validation on 10% of the traffic before the company fully releases the model in production. The company uses an Amazon SageMaker endpoint behind an Application Load Balancer (ALB) to serve the model.

Which solution will set up the required online validation with the LEAST operational overhead?

**A.** Use production variants to add the new model to the existing SageMaker endpoint. Set the variant weight to 0.1 for the new model. Monitor the number of invocations by using Amazon CloudWatch.

**B.** Use production variants to add the new model to the existing SageMaker endpoint. Set the variant weight to 1 for the new model. Monitor the number of invocations by using Amazon CloudWatch.

**C.** Create a new SageMaker endpoint. Use production variants to add the new model to the new endpoint. Monitor the number of invocations by using Amazon CloudWatch.

**D.** Configure the ALB to route 10% of the traffic to the new model at the existing SageMaker endpoint. Monitor the number of invocations by using AWS CloudTrail.

**ANSWER: A**

**Explanation:**

Use production variants to add the new model to the existing SageMaker endpoint. Set the variant weight to 0.1 for the new model. Monitor the number of invocations by using Amazon CloudWatch. Amazon SageMaker production variants support hosting multiple model variants behind one endpoint and routing requests among them according to assigned traffic weights. Assigning a weight of 0.1 to the new model directs approximately 10% of endpoint invocations to that variant, while the existing variant can retain the remaining traffic. This provides an online canary validation using real production requests without deploying and operating a separate endpoint or implementing custom traffic-routing logic at the load balancer layer.

SageMaker publishes endpoint and variant-level operational metrics to Amazon CloudWatch, including invocation counts, latency, and error-related metrics. These metrics allow the team to observe the new model's real-world behavior during the validation period. If the results meet release requirements, the team can update the production-variant weights incrementally until the new model receives all traffic. This approach uses managed SageMaker deployment controls and monitoring capabilities, minimizing operational work while directly satisfying the required 10% traffic split. See the Amazon SageMaker production variants documentation and SageMaker CloudWatch monitoring documentation.

## QUESTION NO: 27

A company uses Amazon SageMaker Studio to develop an ML model. The company has a single SageMaker Studio domain. An ML engineer needs to implement a solution that provides an automated alert when SageMaker AI compute costs reach a specific threshold.

Which solution will meet these requirements?

**A.** Add resource tagging by editing the SageMaker AI user profile in the SageMaker AI domain. Configure AWS Cost Explorer to send an alert when the threshold is reached.

**B.** Add resource tagging by editing the SageMaker AI user profile in the SageMaker AI domain. Configure AWS Budgets to send an alert when the threshold is reached.

**C.** Add resource tagging by editing each user's IAM profile. Configure AWS Cost Explorer to send an alert when the threshold is reached.

**D.** Add resource tagging by editing each user's IAM profile. Configure AWS Budgets to send an alert when the threshold is reached.

## ANSWER: B

**Explanation:**

Add resource tagging by editing the SageMaker AI user profile in the SageMaker AI domain. Configure AWS Budgets to send an alert when the threshold is reached. This approach provides both cost attribution and automated notification. Tags on the SageMaker AI Studio user profile allow the organization to organize and identify SageMaker AI usage by a meaningful business dimension, such as a project, team, or environment. To use a tag for billing analysis and budget filtering, the organization must activate the applicable user-defined cost allocation tag in the AWS Billing and Cost Management console; activated tags then become available in Cost Explorer and AWS Budgets after cost data is processed.

AWS Budgets is designed to evaluate actual or forecasted costs against a defined monetary threshold. A cost budget can be scoped to SageMaker AI costs and, when needed, filtered by the activated cost allocation tag. The budget can issue notifications through configured email or Amazon SNS subscribers when the defined threshold is reached. This directly meets the requirement for an automated alert rather than merely viewing or analyzing cost data. SageMaker AI tagging guidance is available in the Amazon SageMaker AI Developer Guide, and budget thresholds and notifications are documented in the AWS Budgets User Guide.

## QUESTION NO: 28

An ML engineer is building a model to predict house and apartment prices. The model uses three features: Square Meters, Price, and Age of Building. The dataset has 10,000 data rows. The data includes data points for one large mansion and one extremely small apartment.

The ML engineer must perform preprocessing on the dataset to ensure that the model produces accurate predictions for the typical house or apartment.

Which solution will meet these requirements?

**A.** Remove the outliers and perform a log transformation on the Square Meters variable.

**B.** Keep the outliers and perform normalization on the Square Meters variable.

**C.** Remove the outliers and perform one-hot encoding on the Square Meters variable.

**D.** Keep the outliers and perform one-hot encoding on the Square Meters variable.

**ANSWER: A**

**Explanation:**

Remove the outliers and perform a log transformation on the Square Meters variable. The single mansion and extremely small apartment are unusual observations relative to the 10,000-row population of typical properties. Such extreme values can disproportionately affect learned relationships between size, age, and price, particularly for regression training objectives that penalize large prediction errors heavily. Removing clearly exceptional observations, when the stated objective is accuracy for typical houses and apartments, helps the training distribution better represent the intended prediction population.

A logarithmic transformation of Square Meters is then appropriate because property-size data is commonly right-skewed: a small number of very large properties occupy a much wider numeric range than ordinary homes. Applying a log transform compresses the upper range of the feature, reduces skew, and can make the relationship between size and price easier for many models to learn. This preprocessing preserves Square Meters as an ordered continuous measurement while reducing the influence of scale differences. Amazon SageMaker Data Wrangler supports both outlier-handling transforms and numerical transformations as part of a reproducible data-preparation flow. See the Amazon SageMaker Data Wrangler transformations documentation and the Data Wrangler data-flow documentation.

## QUESTION NO: 29

A company uses an ML model to recommend videos to users. The model is deployed on Amazon SageMaker AI. The model performed well initially after deployment, but the model's performance has degraded over time.

Which solution can the company use to identify model drift in the future?

**A.** Create a monitoring job in SageMaker Model Monitor. Then create a baseline from the training dataset.

**B.** Create a baseline from the training dataset. Then create a monitoring job in SageMaker Model Monitor.

**C.** Create a baseline by using a built-in rule in SageMaker Clarify. Monitor the drift in Amazon CloudWatch.

**D.** Retrain the model on new data. Compare the retrained model's performance to the original model's performance.

**ANSWER: B**

**Explanation:**

Create a baseline from the training dataset. Then create a monitoring job in SageMaker Model Monitor. This establishes a reference profile for the data characteristics that the model was trained to handle, such as feature distributions, data types, missing-value rates, and constraint violations. SageMaker Model Monitor can then run scheduled monitoring jobs against production inference data captured from the deployed endpoint and compare that data with the established baseline. Meaningful deviations can indicate data drift, which is a common cause of degraded model quality over time.

The baseline should be created first because it supplies the statistics and constraints that the monitoring schedule evaluates. The company can configure Model Monitor to publish monitoring results and violations to Amazon CloudWatch, where alarms and automated operational workflows can notify the team or initiate investigation and retraining processes. This provides an ongoing, managed approach for detecting changes in real-world input data rather than waiting until degraded business outcomes are discovered manually. AWS documents that SageMaker Model Monitor continuously monitors production models and can detect deviations from a baseline in live data. See the Amazon SageMaker Model Monitor documentation and the guidance for creating baselines.

## QUESTION NO: 30

An ML engineering team runs several Amazon SageMaker training jobs while evaluating different algorithms and hyperparameter values. The team must compare the metrics and configurations of the training jobs. The team also must trace each approved model artifact to the training job and input datasets that produced the model.

Which solutions will meet these requirements? (Choose two.)

**A.** Use Amazon SageMaker Experiments to track the training runs and metrics.

**B.** Use Amazon SageMaker ML Lineage Tracking to record relationships between datasets, jobs, and model artifacts.

**C.** Use Amazon CloudTrail to compare training metrics and hyperparameters.

**D.** Use Amazon CloudWatch Logs to register model versions and input datasets.

**E.** Use AWS Config to track relationships between model artifacts and training data.

**ANSWER: A B**

**Explanation:**

Use Amazon SageMaker Experiments to organize and compare the training runs. SageMaker Experiments can track trials and trial components that contain training-job parameters, input data locations, output artifacts, and metrics. This enables the team to compare results across algorithm and hyperparameter configurations in a centralized interface.

Use Amazon SageMaker ML Lineage Tracking to record relationships among datasets, training jobs, model artifacts, and deployed models. Lineage entities and associations provide traceability through the ML lifecycle and help the team determine which inputs and training activity produced a specific approved model. See the Amazon SageMaker Experiments documentation and the SageMaker ML Lineage Tracking documentation.

## QUESTION NO: 31

An ML engineer is training a simple neural network model. The model's performance improves initially and then degrades after a certain number of epochs.

Which solutions will mitigate this problem? (Select TWO.)

**A.** Enable early stopping on the model.

**B.** Increase dropout in the layers.

**C.** Increase the number of layers.

**D.** Increase the number of neurons.

**E.** Investigate and reduce the sources of model bias.

**ANSWER: A B**

**Explanation:**

This training pattern is characteristic of overfitting: the network learns useful general patterns at first, but after additional epochs it increasingly fits noise or training-specific details, causing validation performance to decline. **Enable early stopping on the model.** is appropriate because early stopping monitors a chosen metric, commonly validation loss, and ends training when that metric no longer improves for a configured number of epochs. It can also restore the best-performing model weights, preserving the model state from before validation performance deteriorated. **Increase dropout in the layers.** is also a suitable regularization measure. During training, dropout randomly sets a fraction of layer inputs to zero. This reduces co-adaptation among neurons and encourages the neural network to learn representations that generalize better to unseen data. In Amazon SageMaker, early stopping can terminate a training job based on objective-metric progress, and SageMaker's TensorFlow and PyTorch environments support implementing standard framework regularization techniques such as dropout. Together, these measures limit excessive training and reduce effective model complexity, directly addressing the observed epoch-dependent degradation. See the Amazon SageMaker early stopping documentation and the TensorFlow Dropout API documentation.

## QUESTION NO: 32

An ML engineer is training a simple neural network model. The models performance improves initially and then degrades after a certain number of epochs.

Which solutions will mitigate this problem? (Select TWO.)

**A.** Enable early stopping on the model.

**B.** Increase dropout in the layers.

**C.** Increase the number of layers.

**D.** Increase the number of neurons.

**E.** Investigate and reduce the sources of model bias.

**ANSWER: A B**

**Explanation:**

The described learning curve is characteristic of overfitting: after initially learning useful, generalizable patterns, the neural network continues optimizing for the training examples and its ability to perform well on unseen validation data declines. “Enable early stopping on the model.” mitigates this by monitoring a validation metric, such as validation loss, and ending training when that metric no longer improves. A typical implementation can also restore the weights from the best-performing epoch, preserving the model state with the strongest validation performance. AWS SageMaker AI supports early stopping for supported training workflows, and early stopping is a standard way to avoid unnecessary training after validation performance has plateaued or worsened.

“Increase dropout in the layers.” is also an appropriate regularization measure. During training, dropout randomly sets a configured fraction of input units to zero. This reduces co-adaptation among neurons and encourages the network to learn patterns that are more robust rather than memorizing particular training examples. The dropout rate must be tuned using validation results, because excessive dropout can prevent useful learning; nevertheless, increasing it from an insufficient level is a well-established way to reduce overfitting. These approaches directly address degradation caused by training too long or by excessive dependence on particular model paths. See the TensorFlow EarlyStopping documentation and the TensorFlow Dropout documentation.

**QUESTION NO: 33**

A company is updating an ML model that is hosted on a SageMaker real-time endpoint. The company needs to release the new model gradually and automatically roll back the deployment if latency or invocation error rates exceed acceptable thresholds.

Which actions will meet these requirements? (Choose two.)

**A.** Configure a blue/green deployment with canary traffic shifting.

**B.** Configure SageMaker deployment guardrails with Amazon CloudWatch alarms.

**C.** Configure a shadow variant to return the new model responses to end users.

**D.** Configure a multi-model endpoint to shift traffic between model versions.

**E.** Configure Amazon Route 53 weighted routing for the endpoint URL.

**ANSWER: A B**

**Explanation:**

Configure a blue/green deployment with canary traffic shifting. Canary traffic shifting initially routes a small portion of traffic to the new endpoint fleet before shifting the remaining traffic after a configured wait interval. This limits customer exposure while the company evaluates the new model under production load.

Configure SageMaker deployment guardrails with Amazon CloudWatch alarms. The alarms can monitor endpoint metrics such as latency and invocation errors during the update. If an alarm enters the ALARM state, SageMaker can automatically

roll back to the previous endpoint configuration. See the Amazon SageMaker deployment guardrails documentation and the SageMaker blue/green deployment documentation.