DUMPSARENA

AWS Certified Machine Learning - Specialty

Amazon AWS AWS-Certified-Machine-Learning-Specialty-MLS-C01

Version Demo

Total Demo Questions: 22

Total Premium Questions: 221

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Engineering	51
Topic 2, Exploratory Data Analysis	34
Topic 3, Modeling	67
Topic 4, Machine Learning Implementation and Operations	69
Total	221

QUESTION NO: 1

A medical imaging company wants to train a computer vision model to detect areas of concern on patients' CT scans. The company has a large collection of unlabeled CT scans that are linked to each patient and stored in an Amazon S3 bucket. The scans must be accessible to authorized users only. A machine learning engineer needs to build a labeling pipeline.

Which set of steps should the engineer take to build the labeling pipeline with the LEAST effort?

A. Create a workforce with AWS Identity and Access Management (IAM). Build a labeling tool on Amazon EC2 Queue images for labeling by using Amazon Simple Queue Service (Amazon SQS). Write the labeling instructions.

B. Create an Amazon Mechanical Turk workforce and manifest file. Create a labeling job by using the built-in image classification task type in Amazon SageMaker Ground Truth. Write the labeling instructions.

C. Create a private workforce and manifest file. Create a labeling job by using the built-in bounding box task type in Amazon SageMaker Ground Truth. Write the labeling instructions.

D. Create a workforce with Amazon Cognito. Build a labeling web application with AWS Amplify. Build a labeling workflow backend using AWS Lambda. Write the labeling instructions.

ANSWER: C

Explanation:

Creating a private workforce and manifest file, then creating an Amazon SageMaker Ground Truth labeling job with the built-in bounding box task type, is the lowest-effort managed approach. SageMaker Ground Truth provides a ready-made labeling workflow and labeling user interface, removing the need to develop, host, and operate a custom application or task-distribution system. A manifest file identifies the Amazon S3 data objects to label and can associate each scan with the required input metadata. The built-in bounding box task is appropriate because the objective is to identify localized areas of concern within each scan; labelers can draw boxes around those regions to produce training annotations for an object-detection computer vision model.

A private workforce is appropriate for sensitive medical imaging data because it limits task access to specifically invited and authorized workers rather than exposing the scans to a public crowd. Ground Truth supports private workforces through Amazon Cognito and provides access controls for labeling jobs and their data. Clear labeling instructions establish a consistent definition of areas of concern and how bounding boxes should be drawn, improving annotation quality. For implementation details, see the SageMaker Ground Truth private workforce documentation and the bounding box labeling task documentation.

QUESTION NO: 2

A company needs to label 500,000 product images for an image classification model. The company already has 5,000 accurately labeled images. Manual labeling is expensive, and the company wants human labelers to review only images for which an automated model is not sufficiently confident.

Which actions should the company take in Amazon SageMaker Ground Truth? (Choose two.)

A. Create an automated data labeling job by using the existing labeled images as the initial training dataset.

B. Set a confidence threshold that routes low-confidence labels to human workers.

C. Use Amazon SageMaker batch transform to create image labels without a trained model.

D. Configure an Amazon SageMaker endpoint with multiple production variants for each product category.

E. Convert the image files to Apache Parquet format before creating the labeling job.

ANSWER: A B

Explanation:

Create an automated data labeling job by using the existing labeled images as the initial training dataset allows Ground Truth to train a machine learning model and use it to label additional images. This reduces the amount of manual work compared with assigning every image directly to human workers.

Set a confidence threshold that routes low-confidence labels to human workers ensures that uncertain predictions receive human review while high-confidence predictions can be accepted automatically. Ground Truth can iteratively use human-labeled results to improve the labeling model and reduce the number of records that require manual annotation. This approach is appropriate when an initial labeled dataset is available and labeling quality must be maintained. See the Amazon SageMaker Ground Truth automated data labeling documentation.

QUESTION NO: 3

When submitting Amazon SageMaker training jobs using one of the built-in algorithms, which common parameters MUST be specified? (Choose three.)

A. The training channel identifying the location of training data on an Amazon S3 bucket.

B. The validation channel identifying the location of validation data on an Amazon S3 bucket.

C. The IAM role that Amazon SageMaker can assume to perform tasks on behalf of the users.

D. Hyperparameters in a JSON array as documented for the algorithm used.

E. The Amazon EC2 instance class specifying whether training will be run using CPU or GPU.

F. The output path specifying where on an Amazon S3 bucket the trained model will persist.

ANSWER: A C E F

Explanation:

The question's "Choose three" instruction is incomplete: four listed items correspond to required elements of an Amazon SageMaker `CreateTrainingJob` request for a standard built-in-algorithm training job. "The training channel identifying the location of training data on an Amazon S3 bucket." supplies the training input through `InputDataConfig`; built-in algorithms require the appropriate training channel and S3 data location. "The IAM role that Amazon SageMaker can assume to perform tasks on behalf of the users." is required as `RoleArn`, allowing SageMaker to access the specified input data and write artifacts. "The Amazon EC2 instance class specifying whether training will be run using CPU or GPU." is part of the required `ResourceConfig`, which defines the training instance type and count. "The output path specifying where on an Amazon S3 bucket the trained model will persist." is required in `OutputDataConfig` so SageMaker has an S3 destination for the model artifacts created by training. These settings establish the data source, permissions, compute environment, and artifact destination necessary to launch and retain a training job. See the CreateTrainingJob API reference and the SageMaker training workflow documentation.

QUESTION NO: 4

A Machine Learning Specialist is using an Amazon SageMaker notebook instance in a private subnet of a corporate VPC. The ML Specialist has important data stored on the Amazon SageMaker notebook instance's Amazon EBS volume, and needs to take a snapshot of that EBS volume. However, the ML Specialist cannot find the Amazon SageMaker notebook instance's EBS volume or Amazon EC2 instance within the VPC.

Why is the ML Specialist not seeing the instance visible in the VPC?

A. Amazon SageMaker notebook instances are based on the EC2 instances within the customer account, but they run outside of VPCs.

B. Amazon SageMaker notebook instances are based on the Amazon ECS service within customer accounts.

C. Amazon SageMaker notebook instances are based on EC2 instances running within AWS service accounts.

D. Amazon SageMaker notebook instances are based on AWS ECS instances running within AWS service accounts.

ANSWER: C

Explanation:

Amazon SageMaker notebook instances are based on EC2 instances running within AWS service accounts. Although a notebook instance can be configured to access resources in a customer VPC and subnet, SageMaker manages the underlying compute infrastructure. The underlying Amazon EC2 instance and its attached Amazon EBS volume are therefore not resources that appear in the customer's EC2 console or as customer-owned instances in the VPC. SageMaker creates and manages networking components that provide the notebook instance with VPC connectivity, allowing it to communicate with permitted VPC resources such as data stores and endpoints without exposing the service-managed host for direct EC2 administration.

This management boundary is intentional: customers administer the notebook instance through the SageMaker console, API, and SageMaker-specific configuration settings rather than through EC2 instance and EBS-volume operations. SageMaker documentation describes notebook instances as managed ML compute instances and explains how VPC configuration controls their access to customer resources. See the Amazon SageMaker notebook instance documentation and the SageMaker notebook VPC access documentation.

QUESTION NO: 5

A data science team is planning to build a natural language processing (NLP) application. The application's text preprocessing stage will include part-of-speech tagging and key phase extraction. The preprocessed text will be input to a custom classification algorithm that the data science team has already written and trained using Apache MXNet.

Which solution can the team build MOST quickly to meet these requirements?

A. Use Amazon Comprehend for the part-of-speech tagging, key phase extraction, and classification tasks.

B. Use an NLP library in Amazon SageMaker for the part-of-speech tagging. Use Amazon Comprehend for the key phase extraction. Use AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier.

C. Use Amazon Comprehend for the part-of-speech tagging and key phase extraction tasks. Use Amazon SageMaker built-in Latent Dirichlet Allocation (LDA) algorithm to build the custom classifier.

D. Use Amazon Comprehend for the part-of-speech tagging and key phase extraction tasks. Use AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier.

ANSWER: D

Explanation:

Use Amazon Comprehend for the part-of-speech tagging and key phase extraction tasks. Use AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier. This is the fastest solution because Amazon Comprehend is a fully managed NLP service that provides both syntax analysis and key phrase extraction through prebuilt APIs. Its syntax analysis returns token-level details, including each token's part of speech and associated score, so the team does not need to develop, train, or operate a separate part-of-speech tagging model. The DetectKeyPhrases API similarly identifies significant phrases directly from the input text. The resulting preprocessing outputs can then be supplied to the existing classifier workflow. Because the classifier has already been implemented and trained with Apache MXNet, an AWS Deep Learning Container provides a ready-to-use, managed container environment with the required deep learning framework support. Running that container in Amazon SageMaker lets the team deploy or continue operating its custom MXNet inference code without rewriting the model for a SageMaker built-in algorithm or creating a new classification model. This combines managed NLP preprocessing with rapid deployment of the team's existing custom implementation. See the Amazon Comprehend syntax analysis documentation and the Amazon SageMaker Deep Learning Containers documentation.

QUESTION NO: 6

A data scientist wants to use Amazon Forecast to build a forecasting model for inventory demand for a retail company. The company has provided a dataset of historic inventory demand for its products as a .csv file stored in an Amazon S3 bucket. The table below shows a sample of the dataset.

timestamp	item_id	demand	category	lead_time
2019-12-14	uni_000736	120	hardware	90
2020-01-31	uni_003429	98	hardware	30
2020-03-04	uni_000211	234	accessories	10

How should the data scientist transform the data?

A. Use ETL jobs in AWS Glue to separate the dataset into a target time series dataset and an item metadata dataset. Upload both datasets as .csv files to Amazon S3.

B. Use a Jupyter notebook in Amazon SageMaker to separate the dataset into a related time series dataset and an item metadata dataset. Upload both datasets as tables in Amazon Aurora.

C. Use AWS Batch jobs to separate the dataset into a target time series dataset, a related time series dataset, and an item metadata dataset. Upload them directly to Forecast from a local machine.

D. Use a Jupyter notebook in Amazon SageMaker to transform the data into the optimized protobuf recordIO format. Upload the dataset in this format to Amazon S3.

ANSWER: A

Explanation:

Use ETL jobs in AWS Glue to separate the dataset into a target time series dataset and an item metadata dataset. Upload both datasets as .csv files to Amazon S3. Amazon Forecast organizes input into dataset types according to whether values vary over time or are fixed for each item. Historic inventory demand is the forecast target, so the target time series dataset must represent each product's demand observations over time, using the required item identifier, timestamp, and target-value fields. Product-level attributes that do not change for each observation, such as a product category or other descriptive product properties shown with the demand records, belong in an item metadata dataset. Separating these logical dataset types prevents static attributes from being duplicated as time-series observations and lets Forecast associate metadata with the corresponding item ID during training. AWS Glue is an appropriate managed ETL service for reading the source CSV data from Amazon S3, selecting and reshaping the relevant fields, and writing Forecast-compatible CSV outputs back to S3. Those S3 objects can then be used when importing datasets into the Forecast dataset group. AWS documents the target time series and item metadata dataset roles and required schemas at Amazon Forecast dataset groups and dataset types, and describes S3-based data import at Importing datasets into Amazon Forecast.

QUESTION NO: 7

A company has set up and deployed its machine learning (ML) model into production with an endpoint using Amazon SageMaker hosting services. The ML team has configured automatic scaling for its SageMaker instances to support workload changes. During testing, the team notices that additional instances are being launched before the new instances are ready. This behavior needs to change as soon as possible.

How can the ML team solve this issue?

A. Decrease the cooldown period for the scale-in activity. Increase the configured maximum capacity of instances.

B. Replace the current endpoint with a multi-model endpoint using SageMaker.

C. Set up Amazon API Gateway and AWS Lambda to trigger the SageMaker inference endpoint.

D. Increase the cooldown period for the scale-out activity.

ANSWER: D

Explanation:

Increasing the cooldown period for the scale-out activity is the appropriate solution because a scale-out cooldown controls how long Application Auto Scaling waits after adding capacity before it initiates another scale-out action. SageMaker endpoint instances require time to launch, download and load model artifacts, initialize containers, and pass health checks

before they can serve inference traffic. If the scale-out cooldown is too short, the scaling policy can continue to observe high demand while newly requested instances are still pending, causing it to launch further instances prematurely. A longer scale-out cooldown gives the new capacity sufficient time to become available and contribute to the endpoint's serving capacity before another scaling decision is made. The team should set this interval based on the endpoint's measured instance startup and model initialization time, while retaining enough responsiveness for expected traffic growth. SageMaker endpoint autoscaling is implemented through Application Auto Scaling, whose target-tracking policies support separately configured scale-out cooldown behavior. See the Amazon SageMaker endpoint autoscaling documentation and the Application Auto Scaling target tracking documentation.

QUESTION NO: 8

A data scientist is developing a churn model from customer account records. The data scientist plans to create a target-encoded feature for the customer region column by replacing each region with the historical proportion of customers in that region who churned.

The model has excellent validation results but performs poorly in production.

Which actions should the data scientist take to prevent target leakage? (Choose two.)

A. Split the dataset into training, validation, and test datasets before creating the target-encoded feature.

B. Calculate the target-encoding values by using only the training dataset.

C. Calculate the target-encoding values by using the combined training, validation, and test datasets.

D. Replace the churn target with a one-hot encoded representation before calculating region-level values.

E. Increase the number of folds in the validation dataset after calculating the target encoding.

ANSWER: A B

Explanation:

Split the dataset into training, validation, and test datasets before creating the target-encoded feature and **calculate the target-encoding values by using only the training dataset** prevent information from validation and test labels from influencing the model inputs. Target encoding uses the target variable itself to create a feature. If the churn rates are calculated from all records before the data is split, the validation and test examples indirectly reveal their label distribution to the training process. This produces overly optimistic offline metrics.

After calculating region-level churn rates from the training set, the data scientist should apply the resulting mapping to validation, test, and production records. Regions not observed during training should use a fallback value, such as the overall training-set churn rate. This preserves an evaluation workflow that represents unseen production data. See the Amazon SageMaker data preparation guidance.

QUESTION NO: 9

A machine learning specialist is developing a regression model to predict rental rates from rental listings. A variable named Wall_Color represents the most prominent exterior wall color of the property. The following is the sample data, excluding all other variables:

Property_ID	Wall_Color
1000	Red
1001	White
1002	Green

The specialist chose a model that needs numerical input data.

Which feature engineering approaches should the specialist use to allow the regression model to learn from the Wall_Color data? (Choose two.)

A. Apply integer transformation and set Red = 1, White = 5, and Green = 10.

B. Add new columns that store one-hot representation of colors.

C. Replace the color name string by its length.

D. Create three columns to encode the color in RGB format.

E. Replace each color name by its training set frequency.

ANSWER: B D

Explanation:

“Add new columns that store one-hot representation of colors” correctly converts the nominal Wall_Color category into model-ready numeric features without assigning a false numeric ranking to colors. Each distinct color receives its own binary indicator column, allowing the regression model to estimate a separate relationship between each wall color and rental rate. This is a standard preprocessing technique for categorical variables when the possible values are known from the training data.

“Create three columns to encode the color in RGB format” is also valid because it maps each color to three numeric components: red, green, and blue intensity. This supplies numerical input while retaining meaningful information about visual similarity and color composition. For example, colors with similar RGB values can be represented as closer together in the feature space than unrelated colors. AWS SageMaker Data Wrangler supports categorical encodings such as one-hot encoding as part of feature transformation workflows, and AWS guidance identifies encoding categorical data as an important preprocessing step for machine learning models. See Amazon SageMaker Data Wrangler transformations and AWS Prescriptive Guidance on data preparation.

QUESTION NO: 10

A manufacturing company has structured and unstructured data stored in an Amazon S3 bucket. A Machine Learning Specialist wants to use SQL to run queries on this data.

Which solution requires the LEAST effort to be able to query this data?

A. Use AWS Data Pipeline to transform the data and Amazon RDS to run queries.

B. Use AWS Glue to catalogue the data and Amazon Athena to run queries.

C. Use AWS Batch to run ETL on the data and Amazon Aurora to run the queries.

D. Use AWS Lambda to transform the data and Amazon Kinesis Data Analytics to run queries.

ANSWER: B

Explanation:

Use AWS Glue to catalogue the data and Amazon Athena to run queries. AWS Glue Data Catalog provides a centralized, managed metadata repository for data stored in Amazon S3. A Glue crawler can automatically inspect supported files and data stores, infer schemas, identify partitions, and create or update catalog tables. This avoids building and operating custom ingestion, transformation, or database-loading infrastructure simply to make S3 data queryable.

Amazon Athena is a serverless interactive query service that uses standard SQL to analyze data directly in Amazon S3. Athena integrates with the AWS Glue Data Catalog, so once the relevant data definitions are cataloged, users can select the database and table definitions and run SQL queries without provisioning servers or loading the source data into a separate relational database. This managed integration offers the lowest operational effort for ad hoc analysis of data in S3, while allowing query results to be written back to S3. For formats and data types that Athena supports, this approach is especially suitable for quickly exploring structured, semi-structured, and appropriately prepared unstructured datasets.

See the Amazon Athena User Guide and the AWS Glue components overview.

QUESTION NO: 11

A company stores training data in an Amazon S3 bucket that is accessible only from a VPC. Amazon SageMaker training jobs run in private subnets and must access the training data without using an internet gateway or a NAT gateway. The company also needs to ensure that the training container cannot make outbound network calls.

Which actions should the company take? (Choose two.)

A. Create an Amazon S3 gateway VPC endpoint and configure the route tables for the training subnets.

B. Enable network isolation on the SageMaker training job.

C. Attach an internet gateway to each private subnet.

D. Assign public IP addresses to the SageMaker training instances.

E. Create an Amazon S3 interface endpoint in place of configuring S3 access through a gateway endpoint.

ANSWER: A B

Explanation:

Create an Amazon S3 gateway VPC endpoint and configure the route tables for the training subnets is correct because an S3 gateway endpoint provides private connectivity from VPC resources to Amazon S3. The training jobs can retrieve input data and write model artifacts without traversing an internet gateway or NAT gateway.

Enable network isolation on the SageMaker training job is also correct because it prevents the training container from making network calls. Network isolation provides an additional control against unintended outbound communication from code running inside the container. SageMaker can still use managed mechanisms to obtain declared input data and write output artifacts. See the Amazon S3 gateway endpoint documentation and the Amazon SageMaker network isolation documentation.

QUESTION NO: 12

A Machine Learning Specialist at a company sensitive to security is preparing a dataset for model training. The dataset is stored in Amazon S3 and contains Personally Identifiable Information (PII).

The dataset:

▪ Must be accessible from a VPC only. ▪ Must not traverse the public internet.

How can these requirements be satisfied?

A. Create a VPC endpoint and apply a bucket access policy that restricts access to the given VPC endpoint and the VPC.

B. Create a VPC endpoint and apply a bucket access policy that allows access from the given VPC endpoint and an Amazon EC2 instance.

C. Create a VPC endpoint and use Network Access Control Lists (NACLs) to allow traffic between only the given VPC endpoint and an Amazon EC2 instance.

D. Create a VPC endpoint and use security groups to restrict access to the given VPC endpoint and an Amazon EC2 instance

ANSWER: A

Explanation:

Create a VPC endpoint and apply a bucket access policy that restricts access to the given VPC endpoint and the VPC. An Amazon S3 gateway VPC endpoint provides private connectivity between resources in a VPC and Amazon S3 without requiring an internet gateway, NAT device, VPN connection, or public IP address. This satisfies the requirement that access to the PII dataset does not traverse the public internet.

The S3 bucket policy should explicitly limit requests using condition keys such as `aws:SourceVpce` for the specific endpoint ID and, where appropriate, `aws:SourceVpc` for the intended VPC. Applying an explicit deny for requests that do not originate through the approved endpoint is a strong control because it prevents access through other network paths, including public S3 endpoints. This ensures that model-training workloads running within the approved VPC can retrieve the dataset while the bucket remains unavailable through unapproved locations. AWS documents using VPC endpoint conditions in S3 bucket policies to restrict bucket access and describes gateway endpoints as private S3 connectivity for VPC resources. See Controlling access from VPC endpoints with Amazon S3 bucket policies and Gateway endpoints for Amazon S3.

QUESTION NO: 13

A Machine Learning Specialist has created a deep learning neural network model that performs well on the training data but performs poorly on the test data.

Which of the following methods should the Specialist consider using to correct this? (Choose three.)

A. Decrease regularization.

B. Increase regularization.

C. Increase dropout.

D. Decrease dropout.

E. Increase feature combinations.

F. Decrease feature combinations.

ANSWER: B C F

Explanation:

The observed pattern—strong training performance coupled with poor test performance—is characteristic of overfitting. The neural network has learned patterns specific to the training set that do not generalize reliably to unseen examples. **Increase regularization.** is appropriate because regularization adds a penalty for excessive model complexity, discouraging overly large weights and helping the model learn a smoother, more generalizable decision function. **Increase dropout.** is also appropriate: during training, dropout randomly omits a proportion of neuron activations, reducing co-adaptation among neurons and acting as a regularization technique that improves generalization. Finally, **Decrease feature combinations.** can reduce the effective complexity of the input representation and model. Feature crosses and numerous derived combinations can enable a model to memorize training-specific relationships; limiting them can reduce variance and make the learned relationships more robust on test data. These approaches all address the core issue by reducing the model's tendency to fit noise or overly specific training-set patterns. AWS describes regularization and dropout as techniques for reducing overfitting in deep-learning workflows. See the Amazon SageMaker Developer Guide and the Amazon Machine Learning Developer Guide.

QUESTION NO: 14

A Machine Learning Specialist is working for an online retailer that wants to run analytics on every customer

visit, processed through a machine learning pipeline. The data needs to be ingested by Amazon Kinesis Data

Streams at up to 100 transactions per second, and the JSON data blob is 100 KB in size.

What is the MINIMUM number of shards in Kinesis Data Streams the Specialist should use to successfully

ingest this data?

A. 1 shards

B. 10 shards

C. 100 shards

D. 1,000 shards

ANSWER: B

Explanation:

10 shards is correct because shard capacity must accommodate both the record rate and total incoming data volume. In provisioned mode, an Amazon Kinesis Data Streams shard supports up to 1 MB per second of write throughput and up to 1,000 records per second for writes. This workload generates 100 records per second, which is well below the record-rate limit of one shard. However, each record is 100 KB, so the aggregate incoming throughput is 100 records/second × 100 KB/record = 10,000 KB/second, or approximately 10 MB/second. The byte-throughput limit is therefore the limiting factor. Dividing the required 10 MB/second write throughput by the 1 MB/second capacity per shard yields a minimum requirement of 10 shards. This capacity calculation ensures the stream can accept the stated peak ingestion rate without exceeding per-shard write throughput. AWS documents the write limits for provisioned Kinesis Data Streams shards, including 1 MB/second and 1,000 records/second per shard, at Amazon Kinesis Data Streams key concepts. Guidance on calculating shard requirements is also available in the Kinesis Data Streams resharding documentation.

QUESTION NO: 15

A Data Scientist is developing a machine learning model to classify whether a financial transaction is fraudulent. The labeled data available for training consists of 100,000 non-fraudulent observations and 1,000 fraudulent observations.

The Data Scientist applies the XGBoost algorithm to the data, resulting in the following confusion matrix when the trained model is applied to a previously unseen validation dataset. The accuracy of the model is 99.1%, but the Data Scientist needs to reduce the number of false negatives.

		Predicted	
		0	1
Actual	0	99,966	34
	1	877	123

Which combination of steps should the Data Scientist take to reduce the number of false negative predictions by the model? (Choose two.)

A. Change the XGBoost eval_metric parameter to optimize based on Root Mean Square Error (RMSE).

B. Increase the XGBoost scale_pos_weight parameter to adjust the balance of positive and negative weights.

C. Increase the XGBoost max_depth parameter because the model is currently underfitting the data.

D. Change the XGBoost eval_metric parameter to optimize based on Area Under the ROC Curve (AUC).

E. Decrease the XGBoost max_depth parameter because the model is currently overfitting the data.

ANSWER: B D

Explanation:

Increasing the XGBoost `scale_pos_weight` parameter is appropriate because fraudulent transactions are the minority, positive class. This parameter increases the loss associated with misclassifying positive examples, causing training to place greater emphasis on detecting fraud. A commonly used starting value is the ratio of negative to positive examples, although the value should be tuned against validation results and business costs. Giving fraud examples more influence makes the classifier less likely to miss them, which directly supports reducing false negatives.

Changing `eval_metric` to Area Under the ROC Curve (AUC) is also appropriate for this highly imbalanced classification problem. Accuracy is dominated by the abundant non-fraudulent class and can remain very high even when the model misses many fraudulent transactions. AUC instead measures how well the model ranks fraudulent observations above non-fraudulent ones across classification thresholds. Using AUC for validation and model selection provides a more meaningful signal of discrimination for rare-event detection; the selected model can then be operated at a threshold that prioritizes fraud recall and fewer missed fraud cases. XGBoost documents both `scale_pos_weight` and AUC evaluation metrics at XGBoost Parameters. AWS also discusses evaluation metrics and imbalanced data considerations in Amazon SageMaker metric definitions.

QUESTION NO: 16

A data scientist has explored and sanitized a dataset in preparation for the modeling phase of a supervised learning task. The statistical dispersion can vary widely between features, sometimes by several orders of magnitude. Before moving on to the modeling phase, the data scientist wants to ensure that the prediction performance on the production data is as accurate as possible.

Which sequence of steps should the data scientist take to meet these requirements?

A. Apply random sampling to the dataset. Then split the dataset into training, validation, and test sets.

B. Split the dataset into training, validation, and test sets. Then rescale the training set and apply the same scaling to the validation and test sets.

C. Rescale the dataset. Then split the dataset into training, validation, and test sets.

D. Split the dataset into training, validation, and test sets. Then rescale the training set, the validation set, and the test set independently.

ANSWER: B

Explanation:

Split the dataset into training, validation, and test sets. Then rescale the training set and apply the same scaling to the validation and test sets. This sequence prevents data leakage while ensuring that features with substantially different numeric ranges are represented consistently. A scaling transformation, such as standardization or min-max normalization, must be fitted only on the training data. For example, the training set's mean and standard deviation are used to standardize training features, and those exact learned values must then transform validation, test, and eventually production data.

This approach preserves the independence of the validation and test sets, so their performance measurements more accurately estimate how the model will behave on previously unseen production records. It also ensures that distance-based, gradient-based, and regularized models are not disproportionately influenced by features with larger numeric magnitudes. During deployment, the preprocessing artifact fitted on the training data should be retained and applied consistently to every incoming prediction request. AWS SageMaker supports this pattern by using processing steps and model pipelines to apply reproducible transformations across training and inference workflows. See Amazon SageMaker Pipelines documentation and scikit-learn guidance on consistent preprocessing.

QUESTION NO: 17

A company deploys an Amazon SageMaker model endpoint to process loan applications. The company must detect changes in the distribution of production input features and must identify degradation in prediction quality after actual loan outcomes become available.

Which solutions should the company implement? (Choose two.)

A. Configure an Amazon SageMaker Model Monitor data quality monitoring schedule.

B. Configure an Amazon SageMaker Model Monitor model quality monitoring schedule after supplying ground truth outcomes.

C. Enable Amazon SageMaker Debugger to compare endpoint predictions with future loan outcomes automatically.

D. Use an Amazon SageMaker processing job only at model training time.

E. Configure Amazon CloudTrail to calculate feature distribution statistics for endpoint requests.

ANSWER: A B

Explanation:

Configure an Amazon SageMaker Model Monitor data quality monitoring schedule is correct because data quality monitoring can compare production endpoint input data against baseline constraints and statistics. This can identify data drift, missing values, changes in feature distributions, and schema violations.

Configure an Amazon SageMaker Model Monitor model quality monitoring schedule after supplying ground truth outcomes is also correct. Model quality monitoring evaluates predictions against actual labels after they become available. For a loan application model, the company can provide the eventual outcome as ground truth and monitor metrics such as accuracy, precision, recall, or other business-appropriate measures. Amazon SageMaker Model Monitor supports separate monitoring capabilities for data quality and model quality. See the Amazon SageMaker Model Monitor documentation.

QUESTION NO: 18

A company has an ecommerce website with a product recommendation engine built in TensorFlow. The recommendation engine endpoint is hosted by Amazon SageMaker. Three compute-optimized instances support the expected peak load of the website.

Response times on the product recommendation page are increasing at the beginning of each month. Some users are encountering errors. The website receives the majority of its traffic between 8 AM and 6 PM on weekdays in a single time zone.

Which of the following options are the MOST effective in solving the issue while keeping costs to a minimum? (Choose two.)

A. Configure the endpoint to use Amazon Elastic Inference (EI) accelerators.

B. Create a new endpoint configuration with two production variants.

C. Configure the endpoint to automatically scale with the InvocationsPerInstance metric.

D. Deploy a second instance pool to support a blue/green deployment of models.

E. Reconfigure the endpoint to use burstable instances.

F. Configure scheduled autoscaling to add endpoint capacity before weekday peak hours and reduce capacity after the peak period.

ANSWER: C F

Explanation:

Configure the endpoint to automatically scale with the InvocationsPerInstance metric because Amazon SageMaker endpoint autoscaling can use this predefined metric to adjust the number of inference instances as demand changes. A target-tracking scaling policy maintains a chosen average number of requests per instance, adding capacity when request volume rises and removing it as traffic declines. This directly addresses elevated latency and request errors caused by insufficient inference capacity without requiring the company to run additional instances continuously.

Configure scheduled autoscaling to increase capacity before the predictable weekday business-hour traffic window and decrease it afterward. The traffic pattern is known in advance: usage is concentrated between 8 AM and 6 PM on weekdays in one time zone, with an identifiable beginning-of-month increase. Scheduled scaling ensures instances are provisioned ahead of anticipated demand, avoiding warm-up delays and protecting response time during the busy period. Scaling down after business hours minimizes charges for unused endpoint capacity. Combining scheduled scaling for predictable demand with InvocationsPerInstance target tracking for actual workload variation provides cost-efficient, responsive capacity management. See the Amazon SageMaker endpoint autoscaling documentation and the Application Auto Scaling scheduled scaling documentation.

QUESTION NO: 19

A Data Scientist needs to migrate an existing on-premises ETL process to the cloud The current process runs at regular time intervals and uses PySpark to combine and format multiple large data sources into a single consolidated output for downstream processing

The Data Scientist has been given the following requirements for the cloud solution

* Combine multiple data sources

* Reuse existing PySpark logic

* Run the solution on the existing schedule

* Minimize the number of servers that will need to be managed

Which architecture should the Data Scientist use to build this solution?

A. Write the raw data to Amazon S3 Schedule an AWS Lambda function to submit a Spark step to a persistent Amazon EMR cluster based on the existing schedule Use the existing PySpark logic to run the ETL job on the EMR cluster Output the results to a "processed" location m Amazon S3 that is accessible tor downstream use

B. Write the raw data to Amazon S3 Create an AWS Glue ETL job to perform the ETL processing against the input data Write the ETL job in PySpark to leverage the existing logic Create a new AWS Glue trigger to trigger the ETL job based on the existing schedule Configure the output target of the ETL job to write to a "processed" location in Amazon S3 that is accessible for downstream use.

C. Write the raw data to Amazon S3 Schedule an AWS Lambda function to run on the existing schedule and process the input data from Amazon S3 Write the Lambda logic in Python and implement the existing PySpartc logic to perform the ETL process Have the Lambda function output the results to a "processed" location in Amazon S3 that is accessible for downstream use

D. Use Amazon Kinesis Data Analytics to stream the input data and perform realtime SQL queries against the stream to carry out the required transformations within the stream Deliver the output results to a "processed" location in Amazon S3 that is accessible for downstream use

ANSWER: B

Explanation:

Write the raw data to Amazon S3, create an AWS Glue ETL job in PySpark, schedule it with an AWS Glue trigger, and write results to a processed S3 location is the best architecture because AWS Glue is a fully managed, serverless data-integration service designed for scheduled batch ETL workloads. Glue ETL jobs support Apache Spark and can be authored in PySpark, allowing the Data Scientist to reuse and adapt the existing transformation logic rather than redesigning the processing application. The job can read and join multiple large S3-based data sources, apply the required formatting and consolidation steps at distributed Spark scale, and persist the resulting dataset to a designated Amazon S3 prefix for downstream consumers.

An AWS Glue scheduled trigger can start the job at the required recurring intervals. Glue provisions, scales, and releases the underlying compute capacity for each job run, so the team does not need to maintain persistent servers or manage an EMR cluster. This directly satisfies the operational requirement to minimize server management while retaining a familiar PySpark programming model. AWS documentation describes Glue ETL jobs as serverless Spark workloads and documents scheduled triggers for time-based job execution. See the AWS Glue ETL jobs documentation and the AWS Glue triggers documentation.

QUESTION NO: 20

A company is converting a large number of unstructured paper receipts into images. The company wants to create a model based on natural language processing (NLP) to find relevant entities such as date, location, and notes, as well as some custom entities such as receipt numbers.

The company is using optical character recognition (OCR) to extract text for data labeling. However, documents are in different structures and formats, and the company is facing challenges with setting up the manual workflows for each document type. Additionally, the company trained a named entity recognition (NER) model for custom entity detection using a small sample size. This model has a very low confidence score and will require retraining with a large dataset.

Which solution for text extraction and entity detection will require the LEAST amount of effort?

A. Extract text from receipt images by using Amazon Textract. Use the Amazon SageMaker BlazingText algorithm to train on the text for entities and custom entities.

B. Extract text from receipt images by using a deep learning OCR model from the AWS Marketplace. Use the NER deep learning model to extract entities.

C. Extract text from receipt images by using Amazon Textract. Use Amazon Comprehend for entity detection, and use Amazon Comprehend custom entity recognition for custom entity detection.

D. Extract text from receipt images by using a deep learning OCR model from the AWS Marketplace. Use Amazon Comprehend for entity detection, and use Amazon Comprehend custom entity recognition for custom entity detection.

ANSWER: C

Explanation:

Extract text from receipt images by using Amazon Textract. Use Amazon Comprehend for entity detection, and use Amazon Comprehend custom entity recognition for custom entity detection. This approach uses managed AWS services for each stage of the workflow, minimizing the operational effort required to support receipts with varied layouts. Amazon Textract performs OCR and can extract text from scanned documents without the company having to build, train, deploy, or maintain an OCR model. Its document-analysis capabilities are designed for unstructured and semi-structured documents, making it suitable for input that varies in format.

Amazon Comprehend provides managed natural language processing for detection of built-in entity types. For organization-specific fields such as receipt numbers, Amazon Comprehend custom entity recognition supports training a custom entity model from labeled examples. This directly addresses the need to identify domain-specific entities while avoiding the engineering work of designing a custom deep learning NER architecture and its training pipeline. The managed services can be combined into a repeatable extraction flow: obtain text with Textract, detect standard entities with Comprehend, and apply the custom entity recognizer for receipt-specific values. See the Amazon Textract Developer Guide and Amazon Comprehend custom entity recognition documentation.

QUESTION NO: 21

A retail company is selling products through a global online marketplace. The company wants to use machine learning (ML) to analyze customer feedback and identify specific areas for improvement. A developer has built a tool that collects customer reviews from the online marketplace and stores them in an Amazon S3 bucket. This process yields a dataset of 40 reviews. A data scientist building the ML models must identify additional sources of data to increase the size of the dataset.

Which data sources should the data scientist use to augment the dataset of reviews? (Choose three.)

A. Emails exchanged by customers and the company's customer service agents

B. Social media posts containing the name of the company or its products

C. A publicly available collection of news articles

D. A publicly available collection of customer reviews

E. Product sales revenue figures for the company

F. Instruction manuals for the company's products

ANSWER: A B D

Explanation:

The most useful augmentation sources are **Emails exchanged by customers and the company's customer service agents**, **Social media posts containing the name of the company or its products**, and **A publicly available collection of customer reviews**. Each source contains customer-authored or customer-related natural-language content that can provide additional examples of product sentiment, recurring defects, feature requests, service issues, and other themes relevant to improving the retail experience.

Customer-service email threads are especially valuable because they often describe a customer's problem, the product involved, and the desired resolution in detail. Social-media content broadens coverage beyond marketplace reviewers and can reveal spontaneous opinions and emerging issues. A review corpus contributes text with a structure and linguistic

characteristics closely aligned to the existing marketplace-review dataset, which helps increase the volume and diversity of examples for tasks such as sentiment analysis, topic modeling, or custom text classification.

Before training, the data scientist should implement appropriate consent, privacy, access-control, and data-governance procedures, particularly for customer communications. The text should also be cleaned, deduplicated, and labeled according to the ML objective. Amazon Comprehend supports sentiment analysis and custom classification for text datasets: Amazon Comprehend sentiment analysis documentation. AWS guidance on preparing text data is available in the Amazon SageMaker Developer Guide.

QUESTION NO: 22

A Data Scientist is building a model to predict customer churn using a dataset of 100 continuous numerical features. The Marketing team has not provided any insight about which features are relevant for churn prediction. The Marketing team wants to interpret the model and see the direct impact of relevant features on the model outcome. While training a logistic regression model, the Data Scientist observes that there is a wide gap between the training and validation set accuracy.

Which methods can the Data Scientist use to improve the model performance and satisfy the Marketing team's needs? (Choose two.)

A. Add L1 regularization to the classifier

B. Add features to the dataset

C. Perform recursive feature elimination

D. Perform t-distributed stochastic neighbor embedding (t-SNE)

E. Perform linear discriminant analysis

ANSWER: A C

Explanation:

Add L1 regularization to the classifier is appropriate because a substantial difference between training and validation accuracy indicates that the model is likely overfitting. L1 regularization adds a penalty based on the absolute magnitude of logistic-regression coefficients. This constrains model complexity and can drive coefficients for less useful features exactly to zero. The resulting sparse model uses a smaller set of original input features, improving generalization while retaining straightforward interpretability: Marketing can examine the sign and magnitude of each retained feature's coefficient to understand its direct association with churn probability.

Perform recursive feature elimination also addresses both needs. Recursive feature elimination repeatedly fits an estimator, ranks features by model importance, and removes the least important features until a selected subset remains. Applied with logistic regression, it can reduce the original 100-feature set to the features that contribute most to prediction. Reducing irrelevant or redundant inputs can decrease overfitting and improve validation performance. Because the selected predictors remain the original business features rather than transformed components, the final logistic-regression coefficients can be directly communicated to Marketing as the estimated directional impact of each relevant feature. See the scikit-learn logistic regression documentation and its recursive feature elimination guidance.