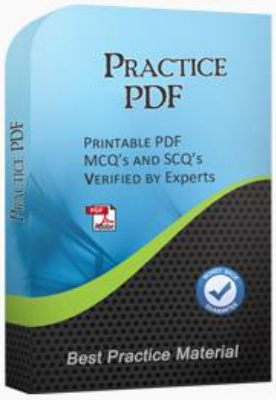
Pass4sureQuiz

Pass4sureQuiz

- Online Tool, Convenient, easy to study.
- Instant Online Access
- Supports All Web Browsers
- Practice Online Anytime
- Test History and Performance Review
- Supports Windows / Mac / Android / iOS, etc.

- Installable Software Application
- Simulates Real Exam Environment
- Builds Exam Confidence
- Supports MS Operating System
- Two Modes For Practice
- Practice Offline Anytime



- Printable PDF Format
- Prepared by IT Experts
- Instant Access to Download
- Study Anywhere, Anytime
- 365 Days Free Updates
- Free PDF Demo Available

Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.

365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.

Money Back Guarantee

Full refund if you fail the corresponding exam in 90 days after purchasing. And Free get any another product.

Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

http://www.pass4surequiz.com/

The best Pass-Sure Quiz materials help you pass exam fast and easily.

Exam : **MLA-C01**

Title : AWS Certified Machine Learning Engineer - Associate

Vendor : Amazon

Version : DEMO

NO.1 An ML engineer wants to run a training job on Amazon SageMaker AI. The training job will train a neural network by using multiple GPUs. The training dataset is stored in Parquet format.
The ML engineer discovered that the Parquet dataset contains files too large to fit into the memory of the SageMaker AI training instances.
Which solution will fix the memory problem?

A. Attach an Amazon Elastic Block Store (Amazon EBS) Provisioned IOPS SSD volume to the instance. Store the files in the EBS volume.

B. Repartition the Parquet files by using Apache Spark on Amazon EMR. Use the repartitioned files for the training job.

C. Change the instance type to Memory Optimized instances with sufficient memory for the training job.

D. Use the SageMaker AI distributed data parallelism (SMDDP) library with multiple instances to split the memory usage.

Answer: B

Explanation:
The issue is caused by oversized Parquet files that cannot be efficiently read into memory during training. The most effective and scalable solution is to repartition the dataset into smaller Parquet files.
AWS best practices for large-scale ML training recommend optimizing data layout, not simply increasing memory. By using Apache Spark on Amazon EMR, the ML engineer can repartition the Parquet files into smaller chunks that can be streamed and processed efficiently by SageMaker training jobs.
Attaching EBS volumes (Option A) increases storage capacity but does not solve in-memory constraints.
Changing to memory-optimized instances (Option C) increases cost and does not address long-term scalability. SMDDP (Option D) distributes gradients and computation, not dataset file sizes.
Therefore, repartitioning the Parquet files is the correct solution.

NO.2 An ML engineer wants to use, prepare, and load data from Amazon S3 for analytics. The ML engineer must run an extract, transform, and load (ETL) job to discover the schema of the data and to store the metadata.
Which solution will meet these requirements with the LEAST manual effort?

A. Use AWS Glue to run the ETL job. Use the job to discover the schema and to store the associated metadata in the AWS Glue Data Catalog.

B. Create an Amazon SageMaker Data Wrangler flow to run the ETL job. Use the job to discover the schema and to store the associated metadata in an S3 bucket.

C. Create an ETL pipeline by using Amazon Athena integrated with AWS Step Functions. Use the pipeline to run the ETL job to discover the schema and to store the associated metadata in an S3 bucket.

D. Launch an Amazon EC2 instance that includes the scikit-learn library to run the ETL job. Use the job to discover the schema and to store the associated metadata in Amazon Redshift.

Answer: A

Explanation:
Option A is correct because AWS Glue is the AWS-native managed ETL service built specifically to discover schema, run ETL jobs, and store metadata in the AWS Glue Data Catalog. AWS

documentation states that Glue crawlers can automatically discover and catalog new or updated data sources, and that the Data Catalog automatically captures and manages schema metadata. This directly matches the requirement to run an ETL job on data in Amazon S3, discover the schema, and store the metadata with the least manual effort.
AWS Glue is also the lowest-effort answer because the service is managed and purpose-built for this workflow. The Glue Data Catalog serves as a persistent metadata repository, and AWS documents that crawlers infer schema information and integrate it into the catalog automatically. That means the ML engineer does not need to build custom schema inference logic or manually maintain metadata storage. This is exactly the kind of manual work the question is trying to avoid.
The other options are not as good. SageMaker Data Wrangler is primarily for visual data preparation and feature engineering, not for running a managed ETL-plus-catalog workflow with schema stored in a metadata catalog. Athena with Step Functions would require assembling more custom orchestration and still does not naturally replace the Glue Data Catalog workflow. Launching an EC2 instance introduces the highest operational overhead and does not align with the requirement for least manual effort. Therefore, the best verified AWS-docs answer is A, because AWS Glue combines ETL, schema discovery, and metadata cataloging in one managed service.

NO.3 A company has a Retrieval Augmented Generation (RAG) application that uses a vector database to store embeddings of documents. The company must migrate the application to AWS and must implement a solution that provides semantic search of text files. The company has already migrated the text repository to an Amazon S3 bucket.
Which solution will meet these requirements?
A. Use an AWS Batch job to process the files and generate embeddings. Use AWS Glue to store the embeddings. Use SQL queries to perform the semantic searches.
B. Use a custom Amazon SageMaker notebook to run a custom script to generate embeddings. Use SageMaker Feature Store to store the embeddings. Use SQL queries to perform the semantic searches.
C. Use the Amazon Kendra S3 connector to ingest the documents from the S3 bucket into Amazon Kendra. Query Amazon Kendra to perform the semantic searches.
D. Use an Amazon Textract asynchronous job to ingest the documents from the S3 bucket. Query Amazon Textract to perform the semantic searches.
Answer: C
Explanation:
Amazon Kendra is an AI-powered search service designed for semantic search use cases. It allows ingestion of documents from an Amazon S3 bucket using the Amazon Kendra S3 connector. Once the documents are ingested, Kendra enables semantic searches with its built-in capabilities, removing the need to manually generate embeddings or manage a vector database. This approach is efficient, requires minimal operational effort, and meets the requirements for a Retrieval Augmented Generation (RAG) application.

NO.4 A company has trained an ML model in Amazon SageMaker. The company needs to host the model to provide inferences in a production environment.
The model must be highly available and must respond with minimum latency. The size of each request will be between 1 KB and 3 MB. The model will receive unpredictable bursts of requests during the day. The inferences must adapt proportionally to the changes in demand.
How should the company deploy the model into production to meet these requirements?

A. Create a SageMaker real-time inference endpoint. Configure auto scaling. Configure the endpoint to present the existing model.

B. Deploy the model on an Amazon Elastic Container Service (Amazon ECS) cluster. Use ECS scheduled scaling that is based on the CPU of the ECS cluster.

C. Install SageMaker Operator on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Deploy the model in Amazon EKS. Set horizontal pod auto scaling to scale replicas based on the memory metric.

D. Use Spot Instances with a Spot Fleet behind an Application Load Balancer (ALB) for inferences. Use the ALBRequestCountPerTarget metric as the metric for auto scaling.

Answer: A

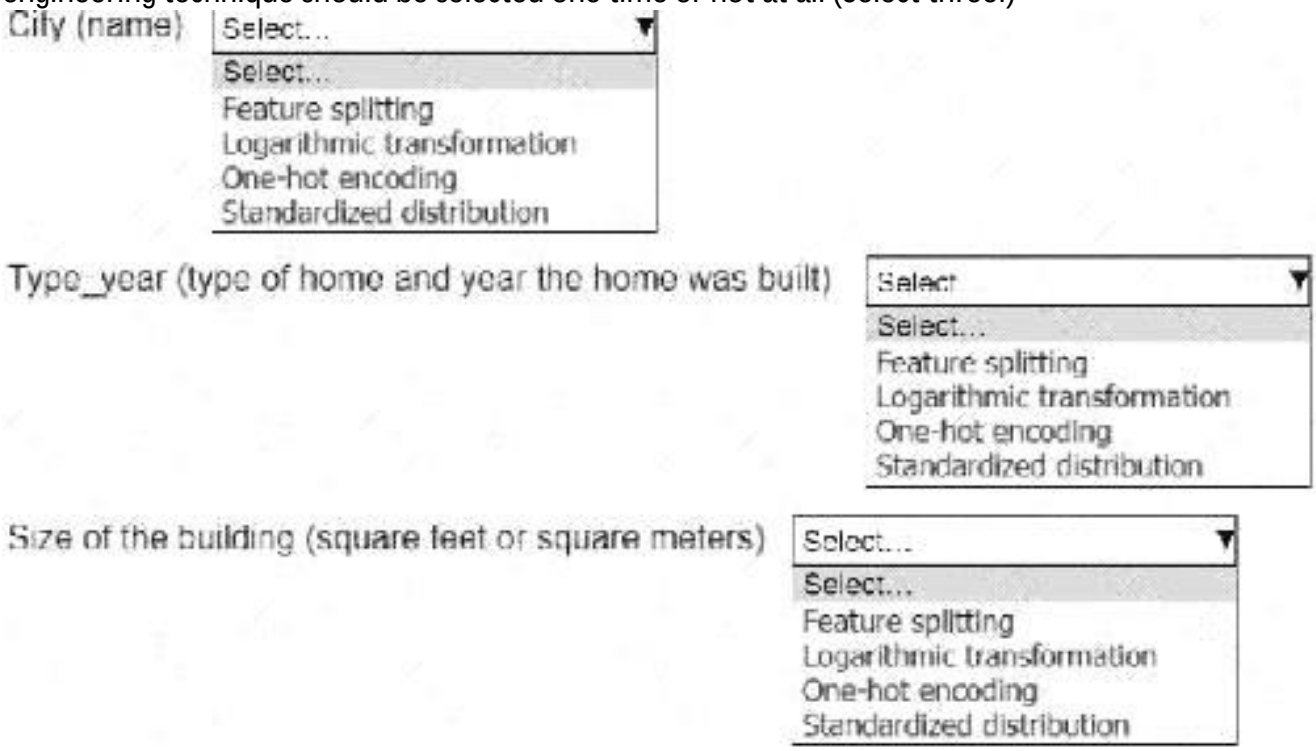
Explanation:

Amazon SageMaker real-time inference endpoints are designed to provide low-latency predictions in production environments. They offer built-in auto scaling to handle unpredictable bursts of requests, ensuring high availability and responsiveness. This approach is fully managed, reduces operational complexity, and is optimized for the range of request sizes (1 KB to 3 MB) specified in the requirements.

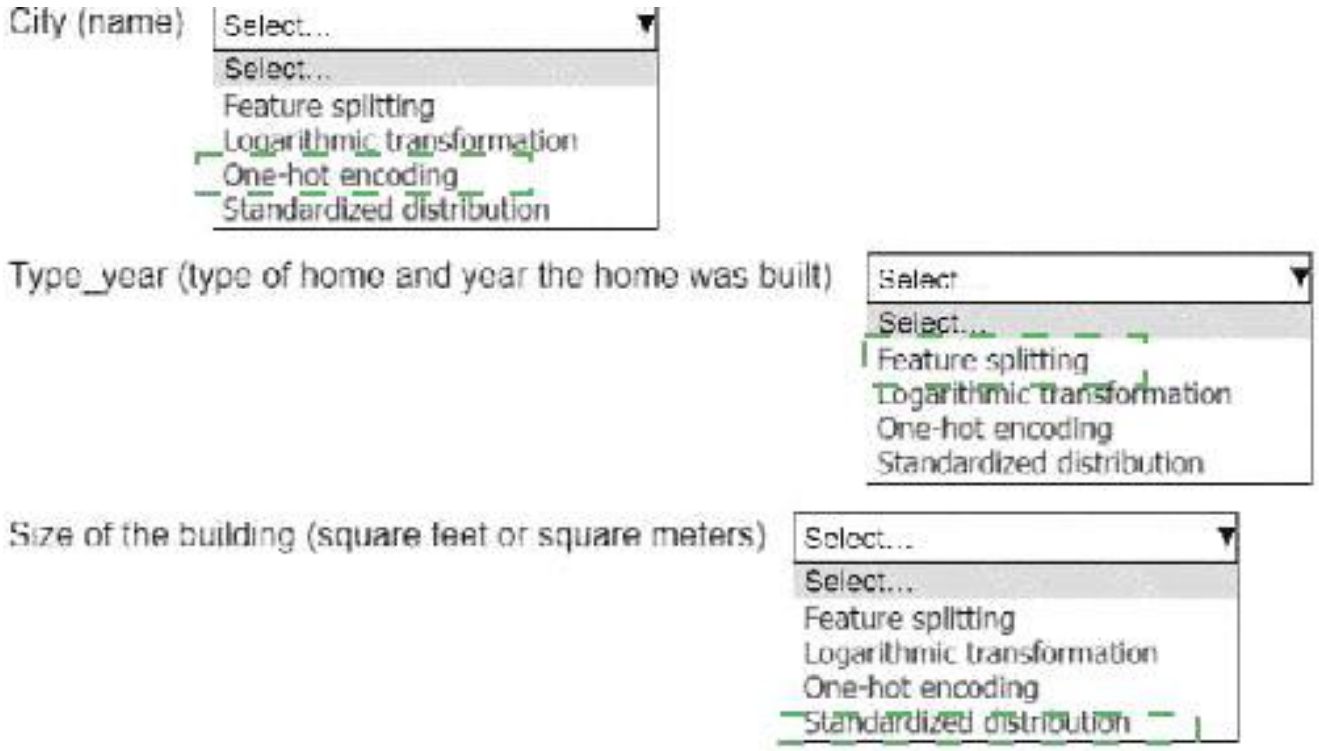
NO.5 An ML engineer is working on an ML model to predict the prices of similarly sized homes. The model will base predictions on several features The ML engineer will use the following feature engineering techniques to estimate the prices of the homes:

* 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:



Explanation:
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 City (name): One-hot encoding Why? The " City " is a categorical feature (non-numeric), so one-hot encoding is used to transform it into a numeric format. This encoding creates binary columns for each unique category (e.g., cities like " New York " or " Los Angeles "), which the model can interpret.
Type_year (type of home and year the home was built): Feature splitting Why? " Type_year " combines two pieces of information into one column, which could confuse the model.
Feature splitting separates this column into two distinct features: " Type of home " and " Year built, " enabling the model to process each feature independently.
Size of the building (square feet or square meters): Standardized distribution Why? Size is a continuous numerical variable, and standardization (scaling the feature to have a mean of 0 and a standard deviation of 1) ensures that the model treats it fairly compared to other features, avoiding bias from differences in feature scale.
By applying these feature engineering techniques, the ML engineer can ensure that the input data is correctly formatted and optimized for the model to make accurate predictions.

NO.6 A company regularly receives new training data from the vendor of an ML model. The vendor delivers cleaned and prepared data to the company ' s Amazon S3 bucket every 3-4 days.
The company has an Amazon SageMaker pipeline to retrain the model. An ML engineer needs to implement a solution to run the pipeline when new data is uploaded to the S3 bucket.
Which solution will meet these requirements with the LEAST operational effort?
A. Create an S3 Lifecycle rule to transfer the data to the SageMaker training instance and to initiate training.
B. Create an AWS Lambda function that scans the S3 bucket. Program the Lambda function to initiate the pipeline when new data is uploaded.

C. Create an Amazon EventBridge rule that has an event pattern that matches the S3 upload. Configure the pipeline as the target of the rule.

D. Use Amazon Managed Workflows for Apache Airflow (Amazon MWAA) to orchestrate the pipeline when new data is uploaded.

Answer: C

Explanation:

Using Amazon EventBridge with an event pattern that matches S3 upload events provides an automated, low- effort solution. When new data is uploaded to the S3 bucket, the EventBridge rule triggers the SageMaker pipeline. This approach minimizes operational overhead by eliminating the need for custom scripts or external orchestration tools while seamlessly integrating with the existing S3 and SageMaker setup.

NO.7 A company uses an Amazon EMR cluster to run a data ingestion process for an ML model. An ML engineer notices that the processing time is increasing.

Which solution will reduce the processing time MOST cost-effectively?

A. Use Spot Instances to increase the number of primary nodes.

B. Use Spot Instances to increase the number of core nodes.

C. Use Spot Instances to increase the number of task nodes.

D. Use On-Demand Instances to increase the number of core nodes.

Answer: C

Explanation:

Amazon EMR clusters consist of primary, core, and task nodes, each with a distinct role. The primary node manages the cluster, core nodes store data and run tasks, and task nodes only run tasks without storing data.

AWS documentation recommends using task nodes for scaling compute capacity when workloads are compute-intensive, such as data ingestion and transformation pipelines.

To reduce processing time cost-effectively, AWS strongly advises using Spot Instances for task nodes. Spot Instances provide the same compute capacity as On-Demand Instances but at a significantly reduced cost, often up to 90% lower. Because task nodes do not store HDFS data, they can be safely interrupted without risking data loss.

Increasing the number of primary nodes is not supported by EMR and would not improve performance.

Increasing core nodes affects both storage and compute and is more expensive, especially when using On- Demand Instances. Option D is therefore the least cost-effective.

AWS EMR best practices explicitly state that scaling out with Spot task nodes is the preferred way to improve performance for transient, parallel workloads such as ETL, ingestion, and feature preparation.

Therefore, Option C is the most cost-effective and AWS-recommended solution.

NO.8 An ML engineer trained an ML model on Amazon SageMaker to detect automobile accidents from dosed- circuit TV footage. The ML engineer used SageMaker Data Wrangler to create a training dataset of images of accidents and non-accidents.

The model performed well during training and validation. However, the model is underperforming in production because of variations in the quality of the images from various cameras.

Which solution will improve the model ' s accuracy in the LEAST amount of time?

A. Collect more images from all the cameras. Use Data Wrangler to prepare a new training dataset.
B. Recreate the training dataset by using the Data Wrangler corrupt image transform. Specify the impulse noise option.
C. Recreate the training dataset by using the Data Wrangler enhance image contrast transform. Specify the Gamma contrast option.
D. Recreate the training dataset by using the Data Wrangler resize image transform. Crop all images to the same size.
Answer: B
Explanation:
The model is underperforming in production due to variations in image quality from different cameras. Using the corrupt image transform with the impulse noise option in SageMaker Data Wrangler simulates real-world noise and variations in the training dataset. This approach helps the model become more robust to inconsistencies in image quality, improving its accuracy in production without the need to collect and process new data, thereby saving time.

NO.9 An ML engineer is tuning an image classification model that performs poorly on one of two classes. The poorly performing class represents an extremely small fraction of the training dataset. Which solution will improve the model's performance?
A. Optimize for accuracy. Use image augmentation on the less common images.
B. Optimize for F1 score. Use image augmentation on the less common images.
C. Optimize for accuracy. Use SMOTE to generate synthetic images.
D. Optimize for F1 score. Use SMOTE to generate synthetic images.
Answer: B
Explanation:
This scenario describes a severely imbalanced classification problem. In such cases, accuracy is a misleading metric, because the model can achieve high accuracy by predicting only the majority class. AWS ML best practices recommend using F1 score (or precision/recall) when evaluating imbalanced datasets.
The F1 score balances false positives and false negatives, making it ideal for assessing minority-class performance.
For image data, image augmentation (rotations, flips, crops, color jitter) is the preferred technique to increase minority-class representation. SMOTE is designed for tabular data and is not suitable for image pixel data.
Therefore, the correct solution is to optimize for F1 score and apply image augmentation.
Thus, Option B is the correct and AWS-aligned answer.

NO.10 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:
In regression problems such as house price prediction, extreme values can significantly distort model learning.
In this dataset, the presence of a large mansion and an extremely small apartment represents clear outliers in the Square Meters feature. According to AWS Machine Learning best practices, outliers can disproportionately influence loss functions (such as mean squared error), leading to poor predictions for the majority of typical data points.
Removing these outliers helps the model focus on learning patterns that apply to the majority of houses and apartments, which aligns with the requirement to produce accurate predictions for typical properties. After removing outliers, applying a log transformation to the Square Meters feature further improves model performance by reducing skewness and stabilizing variance. Log transformations are commonly recommended in AWS and general ML documentation when numerical features span multiple orders of magnitude.
Option B is incorrect because normalization alone does not address the undue influence of extreme outliers.
Option C and D are incorrect because one-hot encoding is intended for categorical variables, not continuous numerical features such as square meters.
Therefore, removing outliers and applying a log transformation is the most statistically sound preprocessing approach.

NO.11 A financial company receives a high volume of real-time market data streams from an external provider. The streams consist of thousands of JSON records every second.
The company needs to implement a scalable solution on AWS to identify anomalous data points.
Which solution will meet these requirements with the LEAST operational overhead?
A. Ingest real-time data into Amazon Kinesis Data Streams. Use the built-in RANDOM_CUT_FOREST function in Amazon Managed Service for Apache Flink to process the data streams and to detect data anomalies.
B. Ingest real-time data into Amazon Kinesis Data Streams. Deploy an Amazon SageMaker AI endpoint for real-time outlier detection. Create an AWS Lambda function to detect anomalies. Use the data streams to invoke the Lambda function.
C. Ingest real-time data into Apache Kafka on Amazon EC2 instances. Deploy an Amazon SageMaker AI endpoint for real-time outlier detection. Create an AWS Lambda function to detect anomalies. Use the data streams to invoke the Lambda function.
D. Send real-time data to an Amazon Simple Queue Service (Amazon SQS) FIFO queue. Create an AWS Lambda function to consume the queue messages. Program the Lambda function to start an AWS Glue extract, transform, and load (ETL) job for batch processing and anomaly detection.
Answer: A
Explanation:
The key requirements are real-time processing, high throughput, and minimal operational overhead. Amazon Kinesis Data Streams is designed for ingesting thousands of events per second with low latency.
For anomaly detection on streaming data, Amazon Managed Service for Apache Flink provides a

built-in Random Cut Forest (RCF) function. RCF is an unsupervised anomaly detection algorithm that works well on numerical streaming data and does not require labeled training data.
This fully managed combination eliminates the need to deploy or maintain SageMaker endpoints, EC2 instances, or custom ML pipelines. Options B and C introduce unnecessary infrastructure and model management overhead. Option D is batch-oriented and unsuitable for real-time anomaly detection.
Therefore, using Kinesis Data Streams with Flink's built-in Random Cut Forest is the most scalable and low- overhead solution.

NO.12 An ML engineer normalized training data by using min-max normalization in AWS Glue DataBrew. The ML engineer must normalize production inference data in the same way before passing the data to the model.
Which solution will meet this requirement?

A. Apply statistics from a well-known dataset to normalize the production samples.

B. Keep the min-max normalization statistics from the training set and use them to normalize the production samples.

C. Calculate new min-max statistics from a batch of production samples and use them to normalize all production samples.

D. Calculate new min-max statistics from each production sample and use them to normalize all production samples.

Answer: B
Explanation:
AWS ML best practices state that data preprocessing applied during training must be applied identically during inference. For min-max normalization, this requires reusing the minimum and maximum values calculated from the training dataset.
If production data is normalized using different statistics, the feature distributions will differ from what the model learned, leading to degraded prediction accuracy. AWS documentation explicitly warns against recomputing normalization parameters on inference data.
Options A, C, and D introduce data leakage or inconsistent feature scaling. Option B ensures consistency between training and inference pipelines and preserves model integrity.
Therefore, Option B is the correct and AWS-aligned solution.

NO.13 Case study
An ML engineer is developing a fraud detection model on AWS. The training dataset includes transaction logs, customer profiles, and tables from an on-premises MySQL database. The transaction logs and customer profiles are stored in Amazon S3.
The dataset has a class imbalance that affects the learning of the model ' s algorithm. Additionally, many of the features have interdependencies. The algorithm is not capturing all the desired underlying patterns in the data.
Before the ML engineer trains the model, the ML engineer must resolve the issue of the imbalanced data.
Which solution will meet this requirement with the LEAST operational effort?

A. Use Amazon Athena to identify patterns that contribute to the imbalance. Adjust the dataset accordingly.

B. Use Amazon SageMaker Studio Classic built-in algorithms to process the imbalanced dataset.

C. Use AWS Glue DataBrew built-in features to oversample the minority class.

D. Use the Amazon SageMaker Data Wrangler balance data operation to oversample the minority class.

Answer: D

Explanation:

Problem Description:

The training dataset has a class imbalance, meaning one class (e.g., fraudulent transactions) has fewer samples compared to the majority class (e.g., non-fraudulent transactions). This imbalance affects the model's ability to learn patterns from the minority class.

Why SageMaker Data Wrangler?

SageMaker Data Wrangler provides a built-in operation called " Balance Data, " which includes oversampling and undersampling techniques to address class imbalances.

Oversampling the minority class replicates samples of the minority class, ensuring the algorithm receives balanced inputs without significant additional operational overhead.

Steps to Implement:

Import the dataset into SageMaker Data Wrangler.

Apply the " Balance Data " operation and configure it to oversample the minority class.

Export the balanced dataset for training.

Advantages:

Ease of Use: Minimal configuration is required.

Integrated Workflow: Works seamlessly with the SageMaker ecosystem for preprocessing and model training.

Time Efficiency: Reduces manual effort compared to external tools or scripts.

NO.14 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:

Early stopping halts training once the performance on the validation dataset stops improving. This prevents the model from overfitting, which is likely the cause of performance degradation after a certain number of epochs.

Dropout is a regularization technique that randomly deactivates neurons during training, reducing overfitting by forcing the model to generalize better. Increasing dropout can help mitigate the problem of performance degradation due to overfitting.

NO.15 A company stores time-series data about user clicks in an Amazon S3 bucket. The raw data consists of millions of rows of user activity every day. ML engineers access the data to develop their ML models.

The ML engineers need to generate daily reports and analyze click trends over the past 3 days by

using Amazon Athena. The company must retain the data for 30 days before archiving the data.
Which solution will provide the HIGHEST performance for data retrieval?

A. Keep all the time-series data without partitioning in the S3 bucket. Manually move data that is older than 30 days to separate S3 buckets.

B. Create AWS Lambda functions to copy the time-series data into separate S3 buckets. Apply S3 Lifecycle policies to archive data that is older than 30 days to S3 Glacier Flexible Retrieval.

C. Organize the time-series data into partitions by date prefix in the S3 bucket. Apply S3 Lifecycle policies to archive partitions that are older than 30 days to S3 Glacier Flexible Retrieval.

D. Put each day ' s time-series data into its own S3 bucket. Use S3 Lifecycle policies to archive S3 buckets that hold data that is older than 30 days to S3 Glacier Flexible Retrieval.

Answer: C

Explanation:

Partitioning the time-series data by date prefix in the S3 bucket significantly improves query performance in Amazon Athena by reducing the amount of data that needs to be scanned during queries. This allows the ML engineers to efficiently analyze trends over specific time periods, such as the past 3 days. Applying S3 Lifecycle policies to archive partitions older than 30 days to S3 Glacier Flexible Retrieval ensures cost- effective data retention and storage management while maintaining high performance for recent data retrieval.

NO.16 A customer call center uses Amazon Transcribe to convert hundreds of audio recordings of conversations between customers and support agents to text files. The call center wants to use the text files to train an ML model. To comply with industry regulations, the call center must remove customer names, addresses, and phone numbers from the training text files.
Which solution will meet these requirements with the LEAST development effort?

A. Use Amazon Bedrock Guardrails to process and redact personal information from the text files.

B. Use the AWS Glue Detect PII transform to remove personal information from the text files.

C. Store the text files in Amazon S3 buckets. Use S3 Object Lambda functions to redact personal information.

D. Configure an Amazon SageMaker Data Wrangler custom transformation to remove personal information from the text files.

Answer: B

Explanation:

Option B is correct because AWS Glue provides a built-in Detect PII transform that can detect, mask, or remove personally identifiable information with minimal custom development. AWS documentation says the Detect PII transform can process predefined AWS-managed PII entity types and supports actions such as removing or masking values. The examples in AWS docs explicitly mention sensitive entities such as phone numbers and addresses, which directly match the problem statement.

The question specifically asks for the least development effort. That wording makes AWS Glue Detect PII the strongest answer because it is a native transformation capability rather than a custom code-heavy workflow.

AWS also documents fine-grained sensitive data detection features that let you apply actions per entity type, improving usability and reducing the need to build custom parsing and redaction logic yourself. This is much easier than creating Lambda-based transformation code or custom text-cleaning logic inside another ML preprocessing tool.

The other options are less suitable. Amazon Bedrock Guardrails is not the standard AWS service documented for bulk ETL-style redaction of training text files in this context. S3 Object Lambda would require more custom engineering to inspect and redact each object. SageMaker Data Wrangler custom transformation would also involve extra implementation work compared with using a purpose-built Glue transform. Because the call center already has text output and simply needs regulated fields like names, addresses, and phone numbers removed before training, the AWS-native low-effort solution is AWS Glue Detect PII. Therefore, the best verified answer is B.

NO.17 A company collects customer data daily and stores it as compressed files in an Amazon S3 bucket partitioned by date. Each month, analysts process the data, check data quality, and upload results to Amazon QuickSight dashboards.
An ML engineer needs to automatically check data quality before the data is sent to QuickSight, with the LEAST operational overhead.
Which solution will meet these requirements?
A. Run an AWS Glue crawler monthly and use AWS Glue Data Quality rules to check data quality.
B. Run an AWS Glue crawler and create a custom AWS Glue job with PySpark to evaluate data quality.
C. Use AWS Lambda with Python scripts triggered by S3 uploads to evaluate data quality.
D. Send S3 events to Amazon SQS and use Amazon CloudWatch Insights to evaluate data quality.
Answer: A
Explanation:
AWS Glue Data Quality provides managed, declarative data quality checks with minimal configuration.
Combined with Glue crawlers, it enables automatic schema discovery and quality validation without custom code.
Option A uses native AWS services designed for this exact purpose, minimizing operational overhead.
Options B and C require custom code and maintenance. Option D is not designed for data validation.
AWS documentation explicitly recommends Glue Data Quality rules for scalable, automated data quality checks in analytics pipelines.
Therefore, Option A is the correct and AWS-aligned solution.

NO.18 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 is experimenting with consecutive training jobs.
How can the company MINIMIZE infrastructure startup times for these jobs?
A. Use Managed Spot Training.
B. Use SageMaker managed warm pools.
C. Use SageMaker Training Compiler.
D. Use the SageMaker distributed data parallelism (SMDDP) library.
Answer: B
Explanation:

When running consecutive training jobs in Amazon SageMaker, infrastructure provisioning can introduce latency, as each job typically requires the allocation and setup of compute resources. To minimize this startup time and enhance efficiency, Amazon SageMaker offers Managed Warm Pools.
Key Features of Managed Warm Pools:
Reduced Latency: Reusing existing infrastructure significantly reduces startup time for training jobs.
Configurable Retention Period: Allows retention of resources after training jobs complete, defined by the KeepAlivePeriodInSeconds parameter.
Automatic Matching: Subsequent jobs with matching configurations (e.g., instance type) can reuse retained infrastructure.
Implementation Steps:
Request Warm Pool Quota Increase: Increase the default resource quota for warm pools through AWS Service Quotas.
Configure Training Jobs:
Set KeepAlivePeriodInSeconds for the first training job to retain resources.
Ensure subsequent jobs match the retained pool ' s configuration to enable reuse.
Monitor Warm Pool Usage: Track warm pool status through the SageMaker console or API to confirm resource reuse.
Considerations:
Billing: Resources in warm pools are billable during the retention period.
Matching Requirements: Jobs must have consistent configurations to use warm pools effectively.
Alternative Options:
Managed Spot Training: Reduces costs by using spare capacity but doesn't address startup latency.
SageMaker Training Compiler: Optimizes training time but not infrastructure setup.
SageMaker Distributed Data Parallelism Library: Enhances training efficiency but doesn't reduce setup time.
By using Managed Warm Pools, the company can significantly reduce startup latency for consecutive training jobs, ensuring faster experimentation cycles with minimal operational overhead.
AWS Documentation: Managed Warm Pools
AWS Blog: Reduce ML Model Training Job Startup Time

NO.19 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:
AWS documentation recommends using RecordIO format with lazy loading to optimize data input pipelines for image-based training workloads. RecordIO is a binary data format that enables

sequential reads, reducing I
/O overhead and improving throughput during training.
By converting JPEG images into RecordIO format, the training job can read data more efficiently from Amazon S3. Lazy loading ensures that only the required data is loaded into memory when needed, which optimizes CPU utilization during computationally intensive preprocessing steps.
Option A (file mode) results in many small S3 GET requests, which can become a bottleneck for large image datasets. Option B changes training behavior and can negatively affect convergence and performance. Option C reduces image quality, which directly impacts model accuracy and violates the requirement.
AWS SageMaker documentation explicitly highlights RecordIO and lazy loading as best practices for high- performance image training pipelines, especially when preprocessing is CPU-intensive.
Therefore, Option D is the correct and AWS-aligned solution.

NO.20 A company ' s ML engineer has deployed an ML model for sentiment analysis to an Amazon SageMaker endpoint. The ML engineer needs to explain to company stakeholders how the model makes predictions.
Which solution will provide an explanation for the model ' s predictions?

A. Use SageMaker Model Monitor on the deployed model.

B. Use SageMaker Clarify on the deployed model.

C. Show the distribution of inferences from A/# testing in Amazon CloudWatch.

D. Add a shadow endpoint. Analyze prediction differences on samples.

Answer: B

Explanation:
SageMaker Clarify is designed to provide explainability for ML models. It can analyze feature importance and explain how input features influence the model ' s predictions. By using Clarify with the deployed SageMaker model, the ML engineer can generate insights and present them to stakeholders to explain the sentiment analysis predictions effectively.

NO.21 An ML engineer is building a logistic regression model to predict customer churn for subscription services.
The dataset contains two string variables: location and job_seniority_level.
The location variable has 3 distinct values, and the job_seniority_level variable has over 10 distinct values.
The ML engineer must perform preprocessing on the variables.
Which solution will meet this requirement?

A. Apply tokenization to location. Apply ordinal encoding to job_seniority_level.

B. Apply one-hot encoding to location. Apply ordinal encoding to job_seniority_level.

C. Apply binning to location. Apply standard scaling to job_seniority_level.

D. Apply one-hot encoding to location. Apply standard scaling to job_seniority_level.

Answer: B

Explanation:
Logistic regression requires numeric input features and is sensitive to how categorical variables are encoded.
AWS feature engineering best practices recommend one-hot encoding for low-cardinality categorical variables with no inherent order and ordinal encoding for categorical variables with a meaningful

order.
The location feature has only three distinct values and no ordinal relationship, making one-hot encoding the most appropriate method. This prevents the model from inferring a false numerical relationship between locations.
The job_seniority_level feature typically has an inherent order (for example: junior, mid-level, senior, lead).
Even with more than 10 categories, ordinal encoding preserves this natural hierarchy while keeping the feature dimensionality manageable.
Tokenization is used for unstructured text, not structured categorical variables. Standard scaling applies only to continuous numeric features and is not suitable for categorical string variables.
AWS documentation explicitly highlights using one-hot encoding for nominal features and ordinal encoding for ordered categorical features when preparing data for linear models such as logistic regression.
Therefore, Option B is the correct and AWS-aligned solution.

NO.22 A company wants to use Amazon SageMaker AI to host an ML model that runs on CPU for real-time predictions. The model has intermittent traffic during business hours and periods of no traffic after business hours.
Which hosting option will serve inference requests in the MOST cost-effective manner?

A. Deploy the model to a real-time endpoint with scheduled auto scaling.

B. Deploy the model to a SageMaker AI Serverless Inference endpoint with provisioned concurrency during business hours.

C. Deploy the model to an asynchronous inference endpoint with auto scaling to zero.

D. Deploy the model to a real-time endpoint and activate it only during business hours using AWS Lambda.

Answer: B
Explanation:
AWS recommends SageMaker Serverless Inference for workloads with intermittent or unpredictable traffic.
Serverless inference automatically scales compute resources to zero when idle, eliminating costs during periods with no traffic.
For business-hour traffic spikes, provisioned concurrency ensures low-latency responses while still avoiding the cost of continuously running instances. This model is especially cost-effective for CPU-based inference workloads.
Real-time endpoints incur costs even when idle, and asynchronous inference is designed for long-running jobs rather than low-latency predictions.
AWS documentation explicitly states that Serverless Inference is the most cost-effective option for intermittent real-time workloads.
Therefore, Option B is the correct choice.

NO.23 An ML engineer needs to deploy a trained model based on a genetic algorithm. Predictions can take several minutes, and requests can include up to 100 MB of data.
Which deployment solution will meet these requirements with the LEAST operational overhead?

A. Deploy on EC2 Auto Scaling behind an ALB.

B. Deploy to a SageMaker AI real-time endpoint.

C. Deploy to a SageMaker AI Asynchronous Inference endpoint.

D. Deploy to Amazon ECS on EC2.

Answer: C

Explanation:

SageMaker Asynchronous Inference is designed for long-running inference workloads and large payloads (up to 1 GB). Requests are queued, processed asynchronously, and results are written to Amazon S3.

Real-time endpoints have payload and timeout limits. EC2 and ECS require infrastructure management, increasing operational overhead.

AWS documentation explicitly recommends asynchronous inference for workloads with large inputs and long execution times.

Therefore, Option C is the correct and most efficient solution.

NO.24 An ML engineer is building an ML model in Amazon SageMaker AI. The ML engineer needs to load historical data directly from Amazon S3, Amazon Athena, and Snowflake into SageMaker AI. Which solution will meet this requirement?

A. Use AWS Glue DataBrew to import the data into SageMaker AI.

B. Build a pipeline in SageMaker Pipelines to process the data. Use AWS DataSync to load the processed data into SageMaker AI.

C. Create a feature store in SageMaker Feature Store. Use an Apache Spark connector to Feature Store to access the data.

D. Use SageMaker Data Wrangler to query and import the data.

Answer: D

Explanation:

AWS provides Amazon SageMaker Data Wrangler as a native tool for importing, transforming, and analyzing data from multiple sources directly into SageMaker Studio. Data Wrangler supports Amazon S3, Amazon Athena, and Snowflake as built-in data sources through managed connectors. Using Data Wrangler, ML engineers can query data from Athena using SQL, load structured files from S3, and securely connect to Snowflake without writing custom ingestion code. This approach significantly reduces development effort and aligns with AWS best practices for rapid ML experimentation.

Option A is incorrect because AWS Glue DataBrew is designed for data preparation but does not natively integrate with SageMaker training workflows. Option B introduces unnecessary complexity and is not intended for direct ML data loading. Option C focuses on feature storage, not raw historical data ingestion.

Therefore, SageMaker Data Wrangler is the correct solution.