# Amazon

(AWS Certified Machine Learning - Specialty)

AWS Certified Machine Learning - Specialty (MLS-C01)



Total: **370 Questions**

Link:

**Question: 1**

A large mobile network operating company is building a machine learning model to predict customers who are likely to unsubscribe from the service. The company plans to offer an incentive for these customers as the cost of churn is far greater than the cost of the incentive.
The model produces the following confusion matrix after evaluating on a test dataset of 100 customers:

| n= 100 | PREDICTED CHURN | PREDICTED CHURN |
|---|---|---|
| | Yes | No |
| ACTUAL Churn Yes | 10 | 4 |
| Actual No | 10 | 76 |

Based on the model evaluation results, why is this a viable model for production?

A.The model is 86% accurate and the cost incurred by the company as a result of false negatives is less than the false positives.

B.The precision of the model is 86%, which is less than the accuracy of the model.

C.The model is 86% accurate and the cost incurred by the company as a result of false positives is less than the false negatives.

D.The precision of the model is 86%, which is greater than the accuracy of the model.

**Answer: C**

**Explanation:**

There are more FP's than FN's, however the costs of FN's are far larger than that of FP's. So:numberof(FP) > numberof(FN), costperunit(FP) << costperunit(FN). This itself could suggest that totalcosts(FP) < totalcosts(FN), but would be somewhat subjective, since it is not stated how far the unitary costs are.What is suggested, however, is that the model is indeed viable (question asks WHY the model is viable, and not WHETHER it's viable). If the model didn't exist, there would be no way that there are FP's or FN's, but churns would still exist, which have the same cost as FN's.So it means the total costs with FP's must be less than the total costs with FN's (churns).

**Question: 2**

A Machine Learning Specialist is designing a system for improving sales for a company. The objective is to use the large amount of information the company has on users' behavior and product preferences to predict which products users would like based on the users' similarity to other users.
What should the Specialist do to meet this objective?

A.Build a content-based filtering recommendation engine with Apache Spark ML on Amazon EMR

B.Build a collaborative filtering recommendation engine with Apache Spark ML on Amazon EMR.

C.Build a model-based filtering recommendation engine with Apache Spark ML on Amazon EMR

D.Build a combinative filtering recommendation engine with Apache Spark ML on Amazon EMR

**Answer: B**

**Explanation:**

The correct answer is B, building a collaborative filtering recommendation engine with Apache Spark ML on Amazon EMR. Here's why:

The problem describes a scenario where predicting user preferences is based on the similarity of their

behavior and preferences to other users. This aligns perfectly with the principles of collaborative filtering. Collaborative filtering algorithms leverage the collective wisdom of the user base to make recommendations. They find users who have similar tastes and preferences and then recommend items that those similar users have liked or purchased.

Option A, content-based filtering, focuses on the attributes of the items themselves. It recommends items similar to those a user has liked in the past, regardless of other users' preferences. This doesn't fit the problem's emphasis on user similarity.

Option C, model-based filtering, is a broader term that can encompass collaborative filtering, but the specific phrasing is less precise than "collaborative filtering." Furthermore, model-based approaches in collaborative filtering context still leverage user-item interaction data.

Option D, combinative filtering, is not a standard or well-defined term in recommendation systems. While recommendation systems can combine different approaches, it's not the core principle being applied here.

Apache Spark ML on Amazon EMR is an excellent choice for implementation. Spark ML provides scalable machine learning libraries well-suited for processing large datasets of user behavior and product preferences.

Amazon EMR provides a managed Hadoop framework that simplifies the deployment and management of Spark clusters, making it easier to build and scale the recommendation engine. Given the potentially large dataset, using a distributed computing framework like Spark on EMR is highly beneficial.Therefore, building a collaborative filtering engine using Spark ML on EMR is the most appropriate and effective solution for the given objective.

Relevant links:

**Collaborative Filtering:**https://en.wikipedia.org/wiki/Collaborative_filtering
**Apache Spark MLlib:**https://spark.apache.org/mllib/
**Amazon EMR:**https://aws.amazon.com/emr/

## Question: 3

A Mobile Network Operator is building an analytics platform to analyze and optimize a company's operations using Amazon Athena and Amazon S3.
The source systems send data in .CSV format in real time. The Data Engineering team wants to transform the data to the Apache Parquet format before storing it on Amazon S3.
Which solution takes the LEAST effort to implement?

A.Ingest .CSV data using Apache Kafka Streams on Amazon EC2 instances and use Kafka Connect S3 to serialize data as Parquet

B.Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Glue to convert data into Parquet.

C.Ingest .CSV data using Apache Spark Structured Streaming in an Amazon EMR cluster and use Apache Spark to convert data into Parquet.

D.Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Kinesis Data Firehose to convert data into Parquet.

**Answer: B**

**Explanation:**

The question asks for the solution that requires the least effort to implement for transforming real-time CSV data into Parquet format and storing it on S3.

Here's why option B (Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Glue to convert data into Parquet) is the best choice:

**Managed Services Minimization of Effort:** AWS Glue is a fully managed extract, transform, and load (ETL) service. This drastically reduces the operational overhead compared to managing infrastructure like EC2 instances (option A) or EMR clusters (option C).

**Kinesis Data Streams for Real-time Ingestion:** Kinesis Data Streams is designed for real-time data ingestion, making it a suitable choice for handling the stream of CSV data.

**Glue's Built-in Parquet Conversion:** AWS Glue has built-in capabilities to read data from Kinesis Data Streams, transform it, and write it to S3 in Parquet format. This is typically done using Glue's DynamicFrames which provide schema evolution capabilities and make it easier to deal with the schema changes that may occur in CSV files. This eliminates the need for custom code to handle the conversion process itself.

**Reduced Coding and Configuration:** Using Glue involves less coding compared to options A and C. You mainly define the data source (Kinesis), the transformation logic using Glue's visual interface or Spark code, and the target location in S3.

**Kinesis Firehose Limitations:** While Kinesis Data Firehose (option D) can ingest data and write it to S3, its transformation capabilities are more limited than Glue's. Firehose supports basic transformations using Lambda functions, but for complex transformations like CSV-to-Parquet conversion, it's less efficient than using Glue's ETL capabilities. Option D would require a more complex Lambda function for the conversion. Also, Firehose does not natively support schema inference/handling the way Glue does with DynamicFrames which makes dealing with CSV schemas easier.

Options A and C require more effort due to managing infrastructure (EC2 and EMR), configuring Kafka Streams or Spark, and writing custom code for data transformation.

Therefore, option B provides the simplest and most managed approach to achieve the desired outcome with the least effort.

**Supporting Links:**

**AWS Glue:**https://aws.amazon.com/glue/
**Amazon Kinesis Data Streams:**https://aws.amazon.com/kinesis/data-streams/
**Apache Parquet:**https://parquet.apache.org/
**AWS Glue DynamicFrames:**https://docs.aws.amazon.com/glue/latest/dg/aws-glue-api-common.html

## Question: 4

A city wants to monitor its air quality to address the consequences of air pollution. A Machine Learning Specialist needs to forecast the air quality in parts per million of contaminates for the next 2 days in the city. As this is a prototype, only daily data from the last year is available.
Which model is MOST likely to provide the best results in Amazon SageMaker?

A.Use the Amazon SageMaker k-Nearest-Neighbors (kNN) algorithm on the single time series consisting of the full year of data with a predictor_type of regressor.

B.Use Amazon SageMaker Random Cut Forest (RCF) on the single time series consisting of the full year of data.

C.Use the Amazon SageMaker Linear Learner algorithm on the single time series consisting of the full year of data with a predictor_type of regressor.

D.Use the Amazon SageMaker Linear Learner algorithm on the single time series consisting of the full year of data with a predictor_type of classifier.

**Answer: C**

**Explanation:**

Here's a detailed justification for why option C is the most likely to provide the best results, along with

explanations of why the other options are less suitable in this context:

**Why Option C (Amazon SageMaker Linear Learner with regressor) is the best choice:**

**Regression Task:** The problem explicitly states the need to forecast air quality in parts per million (ppm), which is a continuous numerical value. This makes it a regression problem, where the goal is to predict a continuous output.

**Linear Learner Suitability:** Linear Learner is a good starting point for regression tasks, especially when dealing with a relatively small dataset (one year of daily data). It's computationally efficient and can often provide a reasonable baseline model.

**Simplicity and Interpretability:** With limited data, a more complex model might overfit. Linear Learner is simpler and less prone to overfitting, making it a practical choice for a prototype. Also, Linear Learner is more interpretable than other complex models, which helps debug and extract insights quickly.

**Time Series Application:** Although Linear Learner is not specifically designed for Time Series it can still be used with one-year data by carefully structuring the input features. For instance, the previous day's air quality can be one of the input features. With only one year data, it is difficult to perform time series decomposition, trend estimation, or seasonality modeling for more specialized methods.

**Why Option A (Amazon SageMaker k-Nearest-Neighbors (kNN) with regressor) is less suitable:**

**Data Density Requirements:** k-NN relies on finding similar data points in the feature space. With only a year of daily data, the density of data points might be insufficient for k-NN to make accurate predictions. **Curse of Dimensionality:** If additional features are added (e.g., lagged air quality values), the feature space becomes higher-dimensional, which can degrade k-NN performance. This makes it more vulnerable when using only one year data.

**Why Option B (Amazon SageMaker Random Cut Forest (RCF)) is unsuitable:**

**Anomaly Detection Purpose:** RCF is primarily designed for anomaly detection, not forecasting. While it can identify unusual patterns in a time series, it's not the appropriate algorithm for predicting future air quality values.

**Lack of Forecasting Capability:** Even if anomalies are detected, RCF doesn't inherently provide a way to predict future values of the time series.

**Why Option D (Amazon SageMaker Linear Learner with classifier) is incorrect:**

**Classification vs. Regression:** As explained previously, forecasting air quality in ppm is a regression problem, not a classification problem. A classifier predicts discrete categories or classes, not continuous values. Therefore, Linear Learner is not the right choice.

**Supporting Documentation:**

**Amazon SageMaker Linear Learner:**https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html
**Amazon SageMaker k-NN:**https://docs.aws.amazon.com/sagemaker/latest/dg/k-nearest-neighbors.html **Amazon SageMaker Random Cut Forest:**
https://docs.aws.amazon.com/sagemaker/latest/dg/randomcutforest.html

In summary, given the limited data and the requirement for a quick prototype, using Amazon SageMaker Linear Learner with a regressor is the most appropriate and efficient approach.

## Question: 5

A Data Engineer needs to build a model using a dataset containing customer credit card information How can the Data Engineer ensure the data remains encrypted and the credit card information is secure?

A.Use a custom encryption algorithm to encrypt the data and store the data on an Amazon SageMaker instance in a VPC. Use the SageMaker DeepAR algorithm to randomize the credit card numbers.

B.Use an IAM policy to encrypt the data on the Amazon S3 bucket and Amazon Kinesis to automatically discard credit card numbers and insert fake credit card numbers.

C.Use an Amazon SageMaker launch configuration to encrypt the data once it is copied to the SageMaker instance in a VPC. Use the SageMaker principal component analysis (PCA) algorithm to reduce the length of the credit card numbers.

D.Use AWS KMS to encrypt the data on Amazon S3 and Amazon SageMaker, and redact the credit card numbers from the customer data with AWS Glue.

**Answer: D**

**Explanation:**

The correct answer is D because it provides a comprehensive approach to securing sensitive credit card data throughout the machine learning pipeline, addressing both encryption and redaction.

Here's a detailed justification:

1. **Encryption at Rest and in Transit:** Using AWS KMS (Key Management Service) to encrypt data on both Amazon S3 (where the raw data likely resides) and Amazon SageMaker (where the model is trained) ensures data is protected both when stored (at rest) and during transit (between services). KMS provides robust encryption using customer-managed keys or AWS-managed keys, giving the data engineer control over the encryption process. This aligns with security best practices and compliance requirements for handling sensitive data. https://aws.amazon.com/kms/

2. **Data Redaction:** Redacting the credit card numbers using AWS Glue is crucial. Encryption alone might not be sufficient if the model inadvertently learns patterns from the actual credit card numbers themselves. Redaction (removing or masking the sensitive information) minimizes the risk of exposing the real credit card data, even if the model is compromised. AWS Glue is a serverless ETL (Extract, Transform, Load) service that can perform data transformation tasks like redaction efficiently.
   https://aws.amazon.com/glue/

3. **Compliance and Security Best Practices:** This combination of encryption and redaction aligns with common compliance standards like PCI DSS, which governs the handling of credit card data. It also follows the principle of least privilege by limiting the exposure of sensitive data to the model training process.

Why other options are incorrect:

**A:** While using a custom encryption algorithm and a VPC adds security, DeepAR is a time series forecasting algorithm and not suited for randomizing credit card numbers. Randomizing sensitive data directly compromises data integrity.

**B:** Using an IAM policy controls access to S3 but does not inherently encrypt the data. Automatically discarding and inserting fake credit card numbers corrupts the dataset and makes it unsuitable for model training.

**C:** Encrypting the data only when copied to the SageMaker instance leaves the data vulnerable when it is initially stored in S3. PCA is for dimensionality reduction, not secure data masking, and reducing the length isn't an appropriate anonymization technique. PCA could also inadvertently expose the data even if the credit card information is reduced in length.

In summary, only option D provides a robust and compliant solution by encrypting the data both at rest and in transit and redacting the sensitive credit card numbers using AWS Glue.

**Question: 6**

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:**

The correct answer is C: Amazon SageMaker notebook instances are based on EC2 instances running within AWS service accounts.

Here's why:

Amazon SageMaker notebook instances, while appearing to operate within a customer's specified VPC, are actually managed using EC2 instances within an AWS-managed service account. This means that the underlying EC2 instance hosting the notebook doesn't reside directly within the customer's VPC from an EC2 management perspective. The notebook instance is accessible within the customer's VPC through network interfaces created by SageMaker.

Option A is incorrect because SageMaker notebook instances operate within the customer's VPC, not outside of it. They are configured within the VPC, and network access controls are applied through VPC security groups.

Option B is incorrect because SageMaker notebook instances are not based on Amazon ECS. They are directly based on EC2 instances, but the EC2 instances exist in AWS managed service accounts, not customer accounts.

Option D is incorrect for the same reasons as B. They use EC2, not ECS. While SageMaker can use ECS for other features (like training jobs), the notebook instances are not directly running on ECS.

The ML Specialist is looking for an EC2 instance in their AWS account's VPC. The actual EC2 instance running the notebook is in an AWS service account's VPC. The network connection from the notebook to the internet or other VPC resources goes through an elastic network interface (ENI) that AWS creates in the customer's VPC. The notebook appears to be inside the VPC, but the underlying EC2 instance and EBS volume are not directly visible or manageable by the customer through standard EC2/EBS interfaces in their own AWS account. Therefore, the ML Specialist won't be able to directly locate the underlying EC2 instance or EBS volume through the EC2 or EBS consoles in their AWS account.

Further research:

Amazon SageMaker Notebook Instances - Security: This official AWS documentation clearly explains the networking and security aspects of SageMaker notebook instances within VPCs.

How Amazon SageMaker Secures Notebook Instance Traffic: A blog post further clarifying the traffic and networking related to SageMaker notebook instances.

**Question: 7**

A Machine Learning Specialist is building a model that will perform time series forecasting using Amazon SageMaker. The Specialist has finished training the model and is now planning to perform load testing on the endpoint so they can configure Auto Scaling for the model variant.
Which approach will allow the Specialist to review the latency, memory utilization, and CPU utilization during the load test?

A.Review SageMaker logs that have been written to Amazon S3 by leveraging Amazon Athena and Amazon QuickSight to visualize logs as they are being produced.

B.Generate an Amazon CloudWatch dashboard to create a single view for the latency, memory utilization, and CPU utilization metrics that are outputted by Amazon SageMaker.

C.Build custom Amazon CloudWatch Logs and then leverage Amazon ES and Kibana to query and visualize the log data as it is generated by Amazon SageMaker.

D.Send Amazon CloudWatch Logs that were generated by Amazon SageMaker to Amazon ES and use Kibana to query and visualize the log data.

**Answer: B**

**Explanation:**

The correct answer is B: Generate an Amazon CloudWatch dashboard to create a single view for the latency, memory utilization, and CPU utilization metrics that are outputted by Amazon SageMaker.

Here's a detailed justification:

Amazon SageMaker automatically emits several key performance metrics to Amazon CloudWatch, including latency, CPU utilization, and memory utilization. These metrics are crucial for monitoring endpoint health, performance, and resource consumption, especially during load testing for Auto Scaling configuration. CloudWatch provides a centralized platform for collecting and visualizing these metrics.

Option B directly leverages this built-in integration. By creating a CloudWatch dashboard, the Machine Learning Specialist can quickly visualize the latency, CPU utilization, and memory utilization metrics generated by SageMaker endpoints in real-time. This provides a consolidated view for monitoring the endpoint's behavior under increasing load, facilitating informed decisions about Auto Scaling configurations. CloudWatch allows for customized graphs, alerting based on metric thresholds, and historical data analysis.

Option A is less efficient and potentially more costly. While Athena and QuickSight can be used to analyze logs, it requires additional configuration to extract relevant performance metrics from the logs and may not provide the near real-time visibility needed during load testing. SageMaker already provides metrics directly to CloudWatch.

Option C involves custom CloudWatch Logs and Elasticsearch Service (Amazon ES) with Kibana. While powerful for log analysis, setting up custom logging and integrating with ES/Kibana is an unnecessary complication for monitoring basic resource utilization. SageMaker already sends these metrics to CloudWatch, making this approach redundant.

Option D, similar to option C, utilizes Amazon ES and Kibana but relies on SageMaker logs, which necessitates parsing through logs to extract the required performance metrics. This adds unnecessary complexity when CloudWatch directly provides these metrics.

In summary, CloudWatch provides a straightforward, integrated, and efficient method for monitoring SageMaker endpoint performance metrics such as latency, CPU utilization, and memory utilization. The dashboard provides the necessary real-time visibility during load testing, supporting effective Auto Scaling configuration.

Supporting Resources:

Monitor Amazon SageMaker with Amazon CloudWatch
Amazon CloudWatch Dashboards

## Question: 8

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:**

The question asks for the solution that requires the least effort to query structured and unstructured data in an S3 bucket using SQL. Option B, using AWS Glue and Amazon Athena, is the most efficient approach. AWS Glue automatically discovers the schema of the data in S3 and creates a metadata catalog. This catalog allows Athena to then query the data directly using SQL without the need for complex ETL processes or data transformation. Athena is a serverless query service that uses standard SQL to analyze data stored in S3.

Options A, C, and D require more effort. Option A involves using AWS Data Pipeline to transform the data and then loading it into Amazon RDS. This involves creating a pipeline, which introduces more complexity. Option C involves using AWS Batch to run ETL jobs and then loading the data into Amazon Aurora. This involves setting up a Batch environment and creating ETL scripts. Option D involves using AWS Lambda to transform the data and then using Amazon Kinesis Data Analytics to run queries. While Kinesis Data Analytics can use SQL, it's more geared toward real-time streaming data. Using Lambda to transform the data would add considerable operational overhead.

Glue and Athena integrate seamlessly for querying data in S3. Athena directly works with the Glue Data Catalog. It avoids the complexities of data warehousing or complex transformations before querying. Thus making option B the least effort choice.

AWS Glue DocumentationAmazon Athena Documentation

## Question: 9

A Machine Learning Specialist is developing a custom video recommendation model for an application. The dataset used to train this model is very large with millions of data points and is hosted in an Amazon S3 bucket. The Specialist wants to avoid loading all of this data onto an Amazon SageMaker notebook instance because it would take hours to move and will exceed the attached 5 GB Amazon EBS volume on the notebook instance.
Which approach allows the Specialist to use all the data to train the model?

A.Load a smaller subset of the data into the SageMaker notebook and train locally. Confirm that the training code is executing and the model parameters seem reasonable. Initiate a SageMaker training job using the full dataset from the S3 bucket using Pipe input mode.

B.Launch an Amazon EC2 instance with an AWS Deep Learning AMI and attach the S3 bucket to the instance. Train on a small amount of the data to verify the training code and hyperparameters. Go back to Amazon SageMaker and train using the full dataset

C.Use AWS Glue to train a model using a small subset of the data to confirm that the data will be compatible with Amazon SageMaker. Initiate a SageMaker training job using the full dataset from the S3 bucket using Pipe

input mode.

D.Load a smaller subset of the data into the SageMaker notebook and train locally. Confirm that the training code is executing and the model parameters seem reasonable. Launch an Amazon EC2 instance with an AWS Deep Learning AMI and attach the S3 bucket to train the full dataset.

**Answer: A**

**Explanation:**

The best approach is **A. Load a smaller subset of the data into the SageMaker notebook and train locally. Confirm that the training code is executing and the model parameters seem reasonable. Initiate a SageMaker training job using the full dataset from the S3 bucket using Pipe input mode.**

Here's why:

**SageMaker's Intended Use:** SageMaker is designed for large-scale machine learning, and it's not efficient to manage large datasets on the notebook instance itself. The 5GB EBS volume confirms this.

**Local Subset Testing:** Training on a small subset on the notebook verifies the code's functionality, data loading, and initial hyperparameter settings without the delay of processing the entire dataset.

**SageMaker Training Jobs:** SageMaker Training Jobs are designed for large datasets. They handle data loading from S3 directly to the training instances efficiently.

**Pipe Input Mode:** Pipe mode is a streaming mechanism that allows SageMaker training instances to directly access data from S3 without requiring the entire dataset to be downloaded at once. This significantly reduces the memory footprint on the training instances and speeds up training.

https://docs.aws.amazon.com/sagemaker/latest/dg/model-access-training-data.html

**Alternatives are Less Efficient/Costly:**

**B:** While using an EC2 instance with a Deep Learning AMI is an option, it's generally less efficient than using SageMaker Training Jobs, which are optimized for machine learning workloads and provide better integration with other AWS services. It also involves more manual configuration.

**C:** AWS Glue is primarily for ETL (Extract, Transform, Load) operations. While Glue can process data, using it solely for model training, even on a subset, is not its primary purpose. Glue also has associated costs which are generally higher than SageMaker.

**D:** Using an EC2 instance directly to train adds complexity to the deployment workflow that SageMaker training jobs can handle more efficiently.

In essence, the Specialist should leverage the power of SageMaker's training jobs and Pipe input mode to efficiently train the model using the full dataset stored in S3. This approach minimizes data transfer bottlenecks, uses SageMaker as it is intended, and avoids the need for large EBS volumes on the notebook instance.

## Question: 10

A Machine Learning Specialist has completed a proof of concept for a company using a small data sample, and now the Specialist is ready to implement an end- to-end solution in AWS using Amazon SageMaker. The historical training data is stored in Amazon RDS.

Which approach should the Specialist use for training a model using that data?

A.Write a direct connection to the SQL database within the notebook and pull data in

B.Push the data from Microsoft SQL Server to Amazon S3 using an AWS Data Pipeline and provide the S3 location within the notebook.

C.Move the data to Amazon DynamoDB and set up a connection to DynamoDB within the notebook to pull data in.

D.Move the data to Amazon ElastiCache using AWS DMS and set up a connection within the notebook to pull

data in for fast access.

**Answer: B**

**Explanation:**

Here's a detailed justification for why option B is the best approach, and why the other options are less suitable for training a model using Amazon SageMaker with data from Amazon RDS:

**Justification for Option B (Push the data from Microsoft SQL Server to Amazon S3 using an AWS Data Pipeline and provide the S3 location within the notebook):**

1. **Scalability and SageMaker Compatibility:** Amazon SageMaker is designed to work seamlessly with data stored in Amazon S3. S3 provides a highly scalable and durable storage solution suitable for large datasets commonly used in machine learning. SageMaker's training jobs can efficiently access and process data directly from S3.

2. **Decoupling:** Extracting data from RDS and storing it in S3 decouples the training process from the database. This is crucial because training can be resource-intensive and might impact the performance of the production RDS database if accessed directly.

3. **AWS Data Pipeline Efficiency:** AWS Data Pipeline automates the data transfer process from RDS to S3. It handles scheduling, error handling, and data transformations if needed, making the data preparation process more robust and manageable.

4. **Data Versioning and Auditability:** Storing data in S3 allows for easier versioning and auditing of the training data. This is important for reproducibility and compliance.

5. **Cost-Effectiveness:** S3 storage is generally more cost-effective than other storage options like DynamoDB or ElastiCache for large datasets used primarily for training.

6. **Best Practice:** Separating the database layer from the ML training workflow aligns with best practices for building scalable and maintainable machine learning pipelines.

**Why other options are less suitable:**

**Option A (Direct Connection to SQL Database):** While technically possible, directly connecting from a SageMaker notebook to the RDS database for training data poses several issues:

**Performance Impact:** Training can put a significant load on the database, potentially impacting other applications relying on the database.
**Security Risks:** Opening direct database access to a notebook increases the security risk.
**Scalability Limitations:** Limited by the database's performance and connection limits.

**Option C (Move to DynamoDB):** DynamoDB is a NoSQL database optimized for fast key-value lookups. While it's fast, it's not ideal for the large-scale data processing typically involved in machine learning training. Also, the cost can be significantly higher than S3 for storing large datasets.

**Option D (Move to ElastiCache):** ElastiCache is an in-memory caching service, designed for very fast data retrieval but is not meant to store large datasets. It's not suitable for storing the entire historical training data. Its primary use case is for caching frequently accessed data.

**Authoritative Links:**

**Amazon S3:**https://aws.amazon.com/s3/
**Amazon SageMaker:**https://aws.amazon.com/sagemaker/
**AWS Data Pipeline:**https://aws.amazon.com/datapipeline/
**Amazon RDS:**https://aws.amazon.com/rds/

In conclusion, moving the data to S3 using AWS Data Pipeline is the most scalable, cost-effective, and secure

approach for training a model with SageMaker, while minimizing the impact on the production RDS database.

## Question: 11

A Machine Learning Specialist receives customer data for an online shopping website. The data includes demographics, past visits, and locality information. The
Specialist must develop a machine learning approach to identify the customer shopping patterns, preferences, and trends to enhance the website for better service and smart recommendations.
Which solution should the Specialist recommend?

A.Latent Dirichlet Allocation (LDA) for the given collection of discrete data to identify patterns in the customer database.

B.A neural network with a minimum of three layers and random initial weights to identify patterns in the customer database.

C.Collaborative filtering based on user interactions and correlations to identify patterns in the customer database.

D.Random Cut Forest (RCF) over random subsamples to identify patterns in the customer database.

**Answer: C**

**Explanation:**

The best solution is collaborative filtering (Option C) because it directly addresses the problem of identifying customer shopping patterns, preferences, and trends based on user interactions. Collaborative filtering leverages the idea that users with similar tastes in the past will likely have similar tastes in the future. It analyzes user interactions (e.g., purchases, ratings, browsing history) and correlations between users or items to make recommendations. This aligns perfectly with the objective of enhancing the website with better service and smart recommendations.

Option A, Latent Dirichlet Allocation (LDA), is a topic modeling technique primarily used for discovering abstract "topics" in a collection of documents. While LDA could be used to find groups of products that are often purchased together, it doesn't directly model user preferences or enable personalized recommendations based on user similarities.

Option B, a neural network, is a powerful technique but requires substantial labeled data for training to accurately identify patterns. The prompt does not indicate the availability of labeled data to train the network.

Developing a deep learning model without explicit training would also be overly complex given the requirements and other alternatives.

Option D, Random Cut Forest (RCF), is an anomaly detection algorithm. While it can identify unusual patterns in the data, it's not the most suitable approach for discovering general customer shopping preferences and providing recommendations. RCF is primarily for identifying outliers or fraudulent activities, not regular patterns.

Collaborative filtering, in contrast, is designed specifically for recommendation systems and pattern identification within user-item interaction data, making it the optimal choice. AWS provides services such as Amazon Personalize, which uses collaborative filtering (and other ML techniques) to create recommendation systems.

For further reading on Collaborative Filtering, refer to:

**Wikipedia:**https://en.wikipedia.org/wiki/Collaborative_filtering
**Amazon Personalize Documentation:**https://docs.aws.amazon.com/personalize/latest/dg/what-is-personalize.html

## Question: 12

A Machine Learning Specialist is working with a large company to leverage machine learning within its products. The company wants to group its customers into categories based on which customers will and will not churn within the next 6 months. The company has labeled the data available to the Specialist.
Which machine learning model type should the Specialist use to accomplish this task?

A.Linear regression
B.Classification
C.Clustering
D.Reinforcement learning

**Answer: B**

**Explanation:**

The problem describes a scenario where the company aims to predict customer churn, a binary outcome (churn or not churn). This prediction task falls under the realm of supervised learning, specifically **classification**. Classification models learn from labeled data (customers labeled as churned or not churned) to assign new data points (customers) to predefined categories or classes.

Linear regression is suitable for predicting continuous numerical values, not for categorical outcomes like churn. Clustering, on the other hand, is an unsupervised learning technique used to group unlabeled data into clusters based on similarity. Since the problem specifies labeled data, clustering is not appropriate. Reinforcement learning involves training an agent to make decisions in an environment to maximize a reward. It's not applicable to predicting churn based on historical data.

Classification algorithms like logistic regression, decision trees, random forests, or gradient boosting machines are well-suited for this problem. These models can learn the patterns and relationships in the customer data to predict which customers are likely to churn. The labeled data provides the necessary target variable (churn or not churn) for training a supervised classification model.

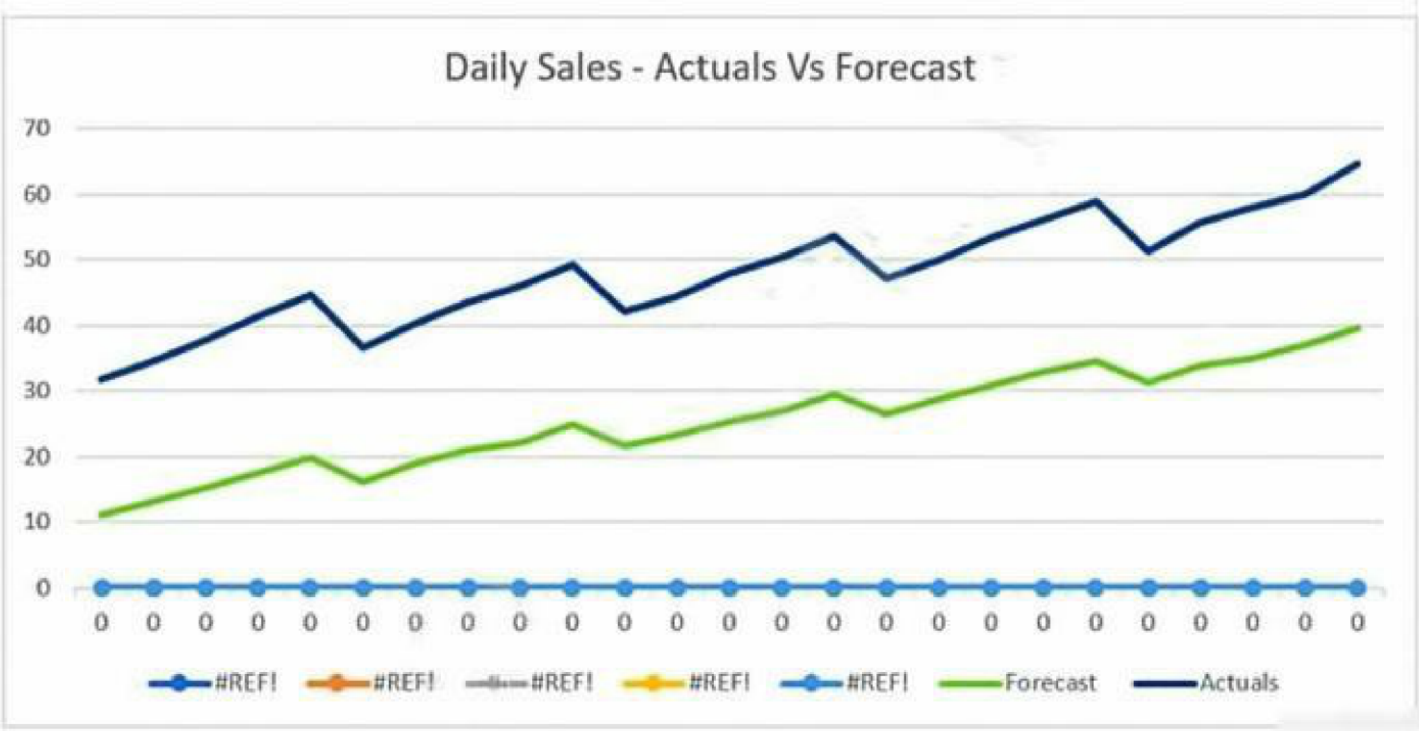
In summary, because the task involves predicting a categorical outcome (churn or not churn) using labeled data, **classification** is the correct machine learning model type.

Here are some authoritative links for further research:

**AWS Documentation on Machine Learning:**https://aws.amazon.com/machine-learning/
**Supervised Learning Concepts:**https://developers.google.com/machine-learning/crash-course/classification/problem-framing
**Classification Algorithms:**https://scikit-learn.org/stable/tutorial/basic/tutorial.html

## Question: 13

The displayed graph is from a forecasting model for testing a time series.




Considering the graph only, which conclusion should a Machine Learning Specialist make about the behavior of the model?

A.The model predicts both the trend and the seasonality well
B.The model predicts the trend well, but not the seasonality.
C.The model predicts the seasonality well, but not the trend.
D.The model does not predict the trend or the seasonality well.

**Answer: A**

**Explanation:**

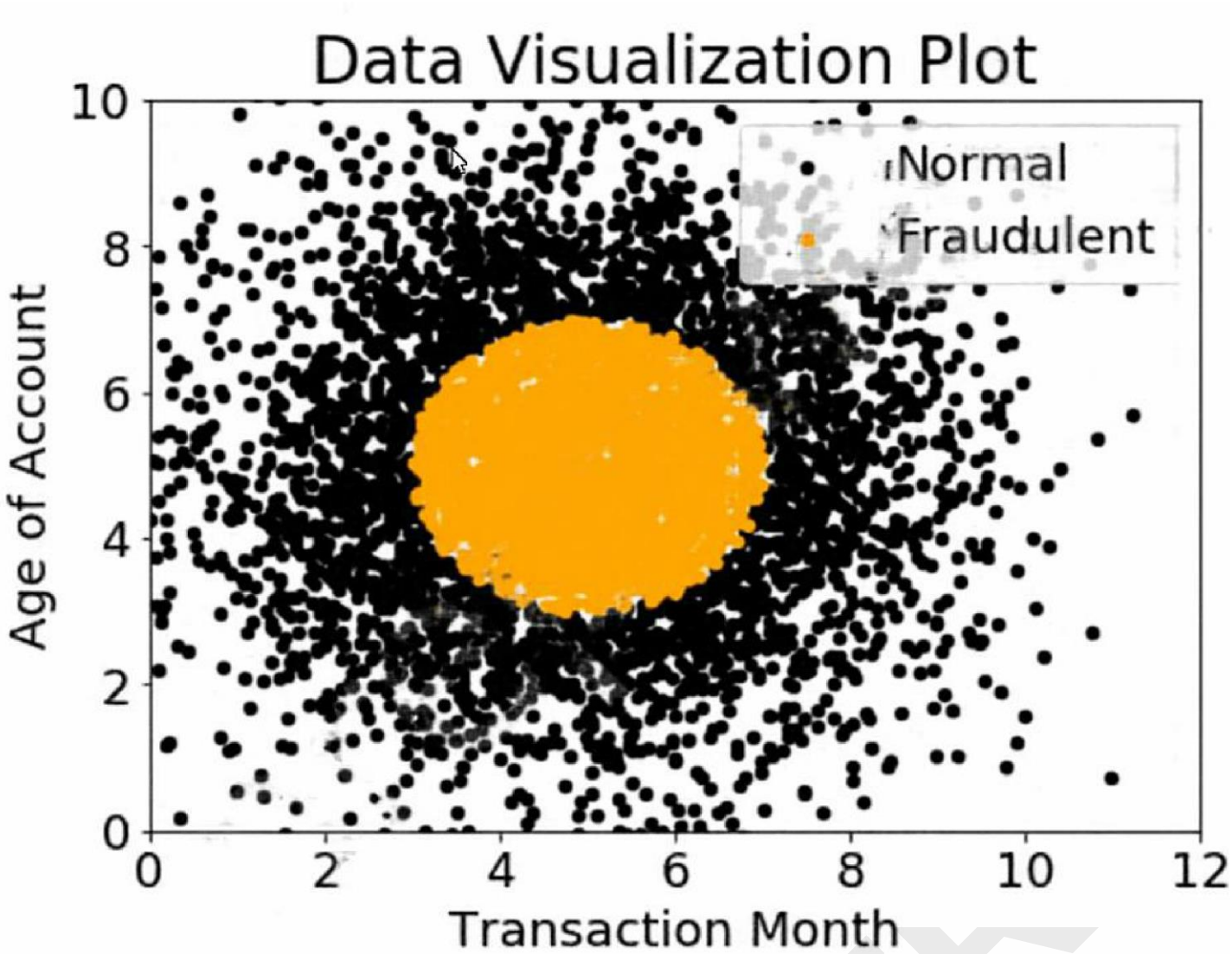
The model predicts both the trend and the seasonality well.

Reference:

https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/

## Question: 14

A company wants to classify user behavior as either fraudulent or normal. Based on internal research, a Machine Learning Specialist would like to build a binary classifier based on two features: age of account and transaction month. The class distribution for these features is illustrated in the figure provided.




Based on this information, which model would have the HIGHEST accuracy?

A.Long short-term memory (LSTM) model with scaled exponential linear unit (SELU)
B.Logistic regression
C.Support vector machine (SVM) with non-linear kernel
D.Single perceptron with tanh activation function

**Answer: C**

**Explanation:**

Answer is C. SVM sample use case is to put the dimensions into a higher hyperplane that can separates it.
Seeing how separable it is, SVM can be used for it.

**Question: 15**

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:**

The correct answer is A: Create a VPC endpoint and apply a bucket access policy that restricts access to the given VPC endpoint and the VPC. This is the most secure and direct approach to fulfilling the requirements.

Here's a detailed justification:

1. **VPC Endpoints for S3:** VPC endpoints for S3 allow resources within your VPC to access S3 without traversing the public internet. This is critical for security-sensitive data, as it prevents PII from potentially being exposed to the public network.
   https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html

2. **Eliminating Public Internet Access:** By using a VPC endpoint, all traffic between the VPC and S3 remains within the Amazon network, avoiding the need for internet gateways or NAT instances.

3. **Bucket Access Policies:** S3 bucket access policies control who can access your S3 buckets and what actions they can perform. These policies are fundamental for securing S3 data.
   https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html

4. **Restricting Access by VPC Endpoint and VPC:** The key to security is applying a bucket policy that specifically restricts access to only traffic originating from the created VPC endpoint and the VPC itself. This ensures that only authorized resources within that VPC can access the S3 bucket containing the PII. The policy would use the aws:sourceVpc and aws:sourceVpcEndpoint condition keys.

5. **Why Other Options are Less Ideal:**

   **B:** While creating a VPC endpoint is correct, allowing access from an EC2 instance in addition to the VPC endpoint doesn't provide the same level of restriction. It's better to enforce access only via the endpoint. Also, implying the endpoint is only for the EC2 instance is incorrect, as the endpoint facilitates connection for the entire VPC.

   **C:** NACLs are primarily for stateless traffic filtering at the subnet level. While NACLs could provide some degree of control, they are not the best tool for controlling access to S3 buckets. Bucket policies provide more granular and specific access control to the object level.

   **D:** Security groups operate at the instance level, and while useful for securing EC2 instances, they are not suitable for directly controlling access to S3 buckets. Bucket Policies are meant to control S3 access.

In summary, creating a VPC endpoint provides a private connection to S3, and a bucket access policy that restricts access based on the VPC endpoint and VPC ID ensures that only authorized resources within that VPC can access the sensitive data, without ever exposing the data to the public internet. This approach offers the best combination of security and functionality for the given requirements.

## Question: 16

During mini-batch training of a neural network for a classification problem, a Data Scientist notices that training

accuracy oscillates.
What is the MOST likely cause of this issue?

A.The class distribution in the dataset is imbalanced.
B.Dataset shuffling is disabled.
C.The batch size is too big.
D.The learning rate is very high.

**Answer: D**

**Explanation:**

The most likely cause of oscillating training accuracy during mini-batch training is a very high learning rate (Option D). Here's why:

A high learning rate causes the optimization algorithm (e.g., gradient descent) to take large steps in the parameter space. Instead of converging smoothly towards the minimum of the loss function, these large steps can overshoot the minimum. This overshoot leads to the accuracy increasing and then decreasing as the model jumps back and forth across the minimum point, resulting in oscillations.

Imbalanced class distribution (Option A) typically leads to biased predictions favoring the majority class, impacting overall accuracy but not necessarily causing rapid oscillations. Disabled dataset shuffling (Option B) can lead to the network learning patterns specific to the order of data, making the training unstable or leading to suboptimal solutions but without the specific symptom of constant, rapid accuracy oscillation. A large batch size (Option C) tends to smooth out the gradient updates and reduces noise, so a batch size that is too big can slow down training or get trapped in local minima, but does not cause accuracy to oscillate.

The relationship between learning rate and convergence is fundamental to neural network training. When the learning rate is properly tuned, the model will converge steadily toward the minimum, improving accuracy with each iteration.A learning rate that is too small can also cause problems, namely slow convergence.When training with mini-batches, the learning rate must be carefully chosen, and it will generally differ depending on the batch size.

For more information on learning rates, consult resources like:

**Machine Learning Mastery on Learning Rate:**https://machinelearningmastery.com/learning-rate-for-deep-learning/
**TensorFlow documentation on Optimizers:**https://www.tensorflow.org/api_docs/python/tf/keras/optimizers

## Question: 17

An employee found a video clip with audio on a company's social media feed. The language used in the video is Spanish. English is the employee's first language, and they do not understand Spanish. The employee wants to do a sentiment analysis.
What combination of services is the MOST efficient to accomplish the task?

A.Amazon Transcribe, Amazon Translate, and Amazon Comprehend
B.Amazon Transcribe, Amazon Comprehend, and Amazon SageMaker seq2seq
C.Amazon Transcribe, Amazon Translate, and Amazon SageMaker Neural Topic Model (NTM)
D.Amazon Transcribe, Amazon Translate and Amazon SageMaker BlazingText

**Answer: A**

**Explanation:**

The most efficient solution to perform sentiment analysis on a Spanish audio clip for an English-speaking employee involves three AWS services working in sequence. Amazon Transcribe is used first to convert the Spanish audio from the video clip into Spanish text. https://aws.amazon.com/transcribe/

Next, Amazon Translate takes the transcribed Spanish text and translates it into English text, making it understandable to the employee. https://aws.amazon.com/translate/

Finally, Amazon Comprehend analyzes the translated English text and performs sentiment analysis to determine the emotional tone or attitude expressed in the content (e.g., positive, negative, neutral). https://aws.amazon.com/comprehend/

Option B is incorrect because while Amazon Transcribe is correct, Amazon Comprehend can directly perform sentiment analysis, rendering SageMaker seq2seq (a sequence-to-sequence model typically used for translation or text summarization, not directly sentiment analysis) unnecessary and less efficient.

Option C includes SageMaker Neural Topic Model (NTM), which is designed to discover topics within a body of text, not for sentiment analysis, making it an unsuitable choice.

Option D is incorrect because BlazingText is optimized for word embedding and text classification, not direct sentiment analysis, and requires considerable training and setup, making it less efficient than Amazon Comprehend for this pre-built task. The other services are correct.

Therefore, using Amazon Transcribe to convert audio to text, Amazon Translate to translate the text to English, and Amazon Comprehend to perform sentiment analysis is the most direct and efficient combination of services for this task as it uses readily available and optimized services.

## Question: 18

A Machine Learning Specialist is packaging a custom ResNet model into a Docker container so the company can leverage Amazon SageMaker for training. The
Specialist is using Amazon EC2 P3 instances to train the model and needs to properly configure the Docker container to leverage the NVIDIA GPUs.
What does the Specialist need to do?

A.Bundle the NVIDIA drivers with the Docker image.

B.Build the Docker container to be NVIDIA-Docker compatible.

C.Organize the Docker container's file structure to execute on GPU instances.

D.Set the GPU flag in the Amazon SageMaker CreateTrainingJob request body.

**Answer: B**

**Explanation:**

The correct answer is B: Build the Docker container to be NVIDIA-Docker compatible. Here's why:

Leveraging GPUs within Docker containers for machine learning, especially on platforms like Amazon SageMaker using EC2 P3 instances, requires specific configurations to enable communication between the container and the host's NVIDIA GPUs. NVIDIA-Docker, now integrated into the NVIDIA Container Toolkit, is designed to solve this problem.

Option A is incorrect because directly bundling NVIDIA drivers into the Docker image isn't the recommended approach. This can lead to compatibility issues between the driver version in the container and the driver version on the host EC2 instance. Moreover, it significantly increases the image size. NVIDIA-Docker dynamically mounts the host's NVIDIA drivers into the container at runtime.

Option C is incorrect because the file structure within the Docker container is generally independent of

whether GPUs are used. The application needs to be written to leverage GPUs (e.g., using CUDA or TensorFlow with GPU support), but the underlying container file structure doesn't inherently change.

Option D is incorrect because while the SageMaker CreateTrainingJob request defines the instance type (e.g., ml.p3.2xlarge) which determines whether GPUs are available, this alone is insufficient for the container to use the GPUs. The Docker container must be NVIDIA-Docker compatible to expose the GPUs to the application running within the container.

NVIDIA-Docker provides the necessary runtime components and tooling to bridge the gap between the container and the host's GPU resources. By ensuring the Docker image is NVIDIA-Docker compatible, the specialist guarantees that the CUDA libraries and drivers from the host EC2 instance are properly exposed to the container, enabling the ResNet model to effectively utilize the GPUs during training. This involves using the nvidia-docker command (or more recently, the NVIDIA Container Toolkit integration with Docker) during the container build or run process.

For authoritative information, refer to the NVIDIA Container Toolkit documentation: https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/index.htmland the AWS documentation on using Docker containers with SageMaker: https://docs.aws.amazon.com/sagemaker/latest/dg/docker-container.html

**Question: 19**

A Machine Learning Specialist is building a logistic regression model that will predict whether or not a person will order a pizza. The Specialist is trying to build the optimal model with an ideal classification threshold.
What model evaluation technique should the Specialist use to understand how different classification thresholds will impact the model's performance?

A.Receiver operating characteristic (ROC) curve
B.Misclassification rate
C.Root Mean Square Error (RMSE)
D.L1 norm

**Answer: A**

**Explanation:**

Here's a detailed justification for why the correct answer is A (Receiver Operating Characteristic (ROC) curve):

The core problem revolves around understanding the impact of different classification thresholds on a logistic regression model's performance in predicting pizza orders. A classification threshold determines the point at which the model classifies a prediction as positive (will order pizza) versus negative (will not order pizza).
Changing this threshold significantly affects the balance between true positives, true negatives, false positives, and false negatives.

An ROC curve is a graphical representation that plots the True Positive Rate (TPR, or sensitivity) against the False Positive Rate (FPR, or 1-specificity) at various threshold settings. By visualizing the ROC curve, the specialist can observe how the model's ability to correctly identify pizza orders (TPR) changes as the tolerance for incorrectly predicting pizza orders (FPR) varies. A curve that is closer to the top left corner of the graph indicates better performance, demonstrating higher TPR and lower FPR across thresholds. The area under the ROC curve (AUC) provides a single scalar value summarizing the overall performance.

Misclassification rate (option B) provides an overall error rate but doesn't reveal the trade-offs between different types of errors. Root Mean Square Error (RMSE) (option C) is used for regression problems and doesn't apply to classification threshold selection. L1 norm (option D) is a regularization technique used during

model training to prevent overfitting, and it is not relevant for evaluating classification threshold performance.

Therefore, the ROC curve is the most appropriate evaluation technique because it helps the specialist visualize and analyze the performance of the logistic regression model across a range of classification thresholds, allowing for informed decisions on selecting the optimal threshold. This allows for the greatest balance between precision and recall based on the specifics of the business requirements.

Supporting resources:

ROC Curves and AUC: Understanding Evaluation Metrics for Imbalanced Classification
Receiver Operating Characteristic (ROC)

**Question: 20**

An interactive online dictionary wants to add a widget that displays words used in similar contexts. A Machine Learning Specialist is asked to provide word features for the downstream nearest neighbor model powering the widget.
What should the Specialist do to meet these requirements?

A.Create one-hot word encoding vectors.
B.Produce a set of synonyms for every word using Amazon Mechanical Turk.
C.Create word embedding vectors that store edit distance with every other word.
D.Download word embeddings pre-trained on a large corpus.

**Answer: D**

**Explanation:**

Here's a detailed justification for why option D is the best approach, along with supporting links:

The goal is to create word features that capture semantic similarity for a nearest neighbor model that recommends words used in similar contexts. Downloading pre-trained word embeddings is the most efficient and effective way to achieve this.

Here's why:

**Semantic Similarity:** Word embeddings (like Word2Vec, GloVe, or fastText) are trained on massive text corpora to capture semantic relationships between words. Words used in similar contexts are placed closer to each other in the embedding space. This directly addresses the problem's requirement of finding words used in similar contexts.

**Computational Efficiency:** Training word embeddings from scratch can be extremely resource-intensive, requiring significant computational power and time. Downloading pre-trained embeddings leverages the work already done by others.

**Generalization:** Pre-trained embeddings have learned from a large corpus and generalize better to unseen data compared to training on a smaller, task-specific dataset.

**Ready Availability:** Many pre-trained word embedding models are readily available for download, making it a quick and easy solution.

Let's look at why the other options are less suitable:

**A. One-hot encoding:** One-hot encoding creates sparse, high-dimensional vectors that do not capture semantic relationships between words. Each word is treated as independent, and the model cannot

understand that "king" and "queen" are more similar than "king" and "apple."

**B. Synonyms using Mechanical Turk:** While synonyms can be helpful, they only capture a narrow aspect of semantic similarity. Words used in similar contexts might not be perfect synonyms but still relevant to the task. Mechanical Turk also introduces potential inconsistencies and requires manual effort.

**C. Edit distance:** Edit distance measures the similarity between strings based on the number of edits required to transform one string into another. This is relevant for spelling correction but doesn't capture semantic meaning. "King" and "ring" would have a small edit distance but are not semantically related in the desired context.

In summary, downloading pre-trained word embeddings provides a quick, efficient, and effective way to capture semantic similarity for recommending words used in similar contexts. It leverages existing work and provides good generalization performance.

**Supporting Links:**

**Word2Vec:**https://papers.nips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f2c492369c905-Paper.pdf
**GloVe:**https://nlp.stanford.edu/projects/glove/
**FastText:**https://fasttext.cc/docs/en/crawl-vectors.html
**Amazon SageMaker BlazingText (for training custom word embeddings, if needed):**
https://docs.aws.amazon.com/sagemaker/latest/dg/blazingtext.html

## Question: 21

A Machine Learning Specialist is configuring Amazon SageMaker so multiple Data Scientists can access notebooks, train models, and deploy endpoints. To ensure the best operational performance, the Specialist needs to be able to track how often the Scientists are deploying models, GPU and CPU utilization on the deployed SageMaker endpoints, and all errors that are generated when an endpoint is invoked.
Which services are integrated with Amazon SageMaker to track this information? (Choose two.)

A.AWS CloudTrail
B.AWS Health
C.AWS Trusted Advisor
D.Amazon CloudWatch
E.AWS Config

**Answer: AD**

**Explanation:**

The question requires identifying AWS services integrated with Amazon SageMaker for tracking model deployments, resource utilization, and endpoint invocation errors.

**A. AWS CloudTrail:** CloudTrail records API calls made to SageMaker. This includes actions like creating, updating, or deleting SageMaker resources (e.g., endpoints, models). Therefore, CloudTrail can track how often Data Scientists are deploying models because these deployment actions are logged as API calls.
[https://docs.aws.amazon.com/sagemaker/latest/dg/cloudtrail-logs.html]

**D. Amazon CloudWatch:** CloudWatch collects metrics and logs related to SageMaker resources. Specifically, CloudWatch can monitor CPU and GPU utilization on deployed SageMaker endpoints. It also captures logs generated during endpoint invocations, including any errors that occur. This allows for monitoring the health and performance of deployed models and troubleshooting issues.
[https://docs.aws.amazon.com/sagemaker/latest/dg/monitoring-cloudwatch.html]

**Why other options are incorrect:**

**B. AWS Health:** AWS Health provides personalized information about the health of AWS services and resources. While it could potentially report on broader SageMaker service availability issues, it doesn't provide the granular endpoint-specific metrics and logs needed for detailed tracking.

**C. AWS Trusted Advisor:** Trusted Advisor provides best practice recommendations related to cost optimization, security, fault tolerance, and performance. It doesn't directly monitor endpoint utilization or track deployment frequency.

**E. AWS Config:** AWS Config tracks resource configurations and changes over time. While it might track the configuration of SageMaker endpoints, it does not directly provide metrics on resource utilization (CPU/GPU) or track error logs generated during endpoint invocations.

Therefore, AWS CloudTrail and Amazon CloudWatch provide the necessary capabilities for tracking model deployments, resource utilization, and endpoint invocation errors.

## Question: 22

A retail chain has been ingesting purchasing records from its network of 20,000 stores to Amazon S3 using Amazon Kinesis Data Firehose. To support training an improved machine learning model, training records will require new but simple transformations, and some attributes will be combined. The model needs to be retrained daily.
Given the large number of stores and the legacy data ingestion, which change will require the LEAST amount of development effort?

A.Require that the stores to switch to capturing their data locally on AWS Storage Gateway for loading into Amazon S3, then use AWS Glue to do the transformation.

B.Deploy an Amazon EMR cluster running Apache Spark with the transformation logic, and have the cluster run each day on the accumulating records in Amazon S3, outputting new/transformed records to Amazon S3.

C.Spin up a fleet of Amazon EC2 instances with the transformation logic, have them transform the data records accumulating on Amazon S3, and output the transformed records to Amazon S3.

D.Insert an Amazon Kinesis Data Analytics stream downstream of the Kinesis Data Firehose stream that transforms raw record attributes into simple transformed values using SQL.

**Answer: D**

**Explanation:**

The best solution is to insert an Amazon Kinesis Data Analytics stream downstream of the Kinesis Data Firehose stream, transforming the data using SQL (Option D). This approach requires the least development effort due to several factors related to ease of integration, scalability, and cost-effectiveness.

Kinesis Data Analytics allows you to process streaming data using standard SQL queries. The transformation logic is simple to implement since SQL is used for data transformation, reducing the need for complex coding.

It easily integrates with existing Kinesis Data Firehose setup. No significant changes are needed for the existing ingestion pipeline.

Kinesis Data Analytics is designed to automatically scale to handle the streaming data volume from 20,000 stores. Also, it has a pay-as-you-go pricing model, making it cost-effective because you only pay for the actual computation.

Alternatives are less attractive. Migrating stores to AWS Storage Gateway (Option A) involves significant changes to data capture infrastructure at each store and requires AWS Glue setup, leading to high development effort. Setting up EMR with Spark (Option B) is more complex and costly for simple transformations and requires managing a cluster. EC2 instances (Option C) require managing infrastructure and developing the transformation logic from scratch.

In summary, Kinesis Data Analytics provides a simple, scalable, and cost-effective solution for transforming data in real-time, minimizing development effort compared to alternatives.

Relevant Links:

**Amazon Kinesis Data Analytics:**https://aws.amazon.com/kinesis/data-analytics/
**Amazon Kinesis Data Firehose:**https://aws.amazon.com/kinesis/data-firehose/

## Question: 23

A Machine Learning Specialist is building a convolutional neural network (CNN) that will classify 10 types of animals. The Specialist has built a series of layers in a neural network that will take an input image of an animal, pass it through a series of convolutional and pooling layers, and then finally pass it through a dense and fully connected layer with 10 nodes. The Specialist would like to get an output from the neural network that is a probability distribution of how likely it is that the input image belongs to each of the 10 classes.
Which function will produce the desired output?

A.Dropout
B.Smooth L1 loss
C.Softmax
D.Rectified linear units (ReLU)

**Answer: C**

**Explanation:**

The question asks for a function that outputs a probability distribution across 10 classes in a CNN.

**Option C, Softmax,** is the correct answer. Softmax is specifically designed to produce a probability distribution over multiple classes. It takes a vector of real numbers as input and transforms it into a probability distribution, where each element represents the probability of the input belonging to a specific class. The sum of these probabilities always equals 1. This aligns perfectly with the requirement of representing the likelihood of an input image belonging to each of the 10 animal classes.

**Option A, Dropout,** is a regularization technique used to prevent overfitting by randomly dropping out neurons during training. It doesn't produce a probability distribution.

**Option B, Smooth L1 loss,** is a loss function used to train regression models. It measures the difference between predicted and actual values and isn't relevant for generating a probability distribution for classification.

**Option D, ReLU (Rectified Linear Unit),** is an activation function that introduces non-linearity in a neural network. It outputs the input directly if it is positive, otherwise, it outputs zero. While ReLU is important in the hidden layers of a CNN, it doesn't produce a probability distribution for the output layer.

Therefore, only Softmax is designed to output a probability distribution, making it the appropriate function for the task.

**Supporting Resources:**

**Softmax:**https://en.wikipedia.org/wiki/Softmax_function
**AWS Machine Learning Documentation:** While AWS specific documentation doesn't usually focus on specific activation functions directly, searching for "classification layers AWS" can point towards solutions using softmax in their examples.

## Question: 24

A Machine Learning Specialist trained a regression model, but the first iteration needs optimizing. The Specialist needs to understand whether the model is more frequently overestimating or underestimating the target. What option can the Specialist use to determine whether it is overestimating or underestimating the target value?

A.Root Mean Square Error (RMSE)
B.Residual plots
C.Area under the curve
D.Confusion matrix

**Answer: B**

**Explanation:**

The correct answer is **B. Residual plots.** Here's why:

A residual plot is a graph that plots the residuals (the difference between the observed and predicted values) on the y-axis against the predicted or independent variable values on the x-axis. It provides a visual way to assess the spread and distribution of errors in a regression model. By examining the plot, a specialist can quickly determine if the model is systematically overestimating or underestimating the target variable.

If the points on the residual plot are randomly scattered around the horizontal axis (residual = 0), it suggests the model is making unbiased predictions. However, if there's a pattern, like a curved shape or more points above or below the line in certain regions, it indicates systematic error. For instance, if the majority of the points are above the x-axis for low predicted values and below for high predicted values, the model tends to underestimate low values and overestimate high values. Conversely, if the majority of the points are below the x-axis for low predicted values and above for high predicted values, the model tends to overestimate low values and underestimate high values.

RMSE (A) provides a single number representing the overall magnitude of errors but doesn't reveal the direction of those errors (overestimation vs. underestimation). AUC (C) is used primarily for classification models and isn't relevant to regression error analysis. A confusion matrix (D) is also a tool for classification problems, summarizing the performance by showing counts of true positives, true negatives, false positives, and false negatives.

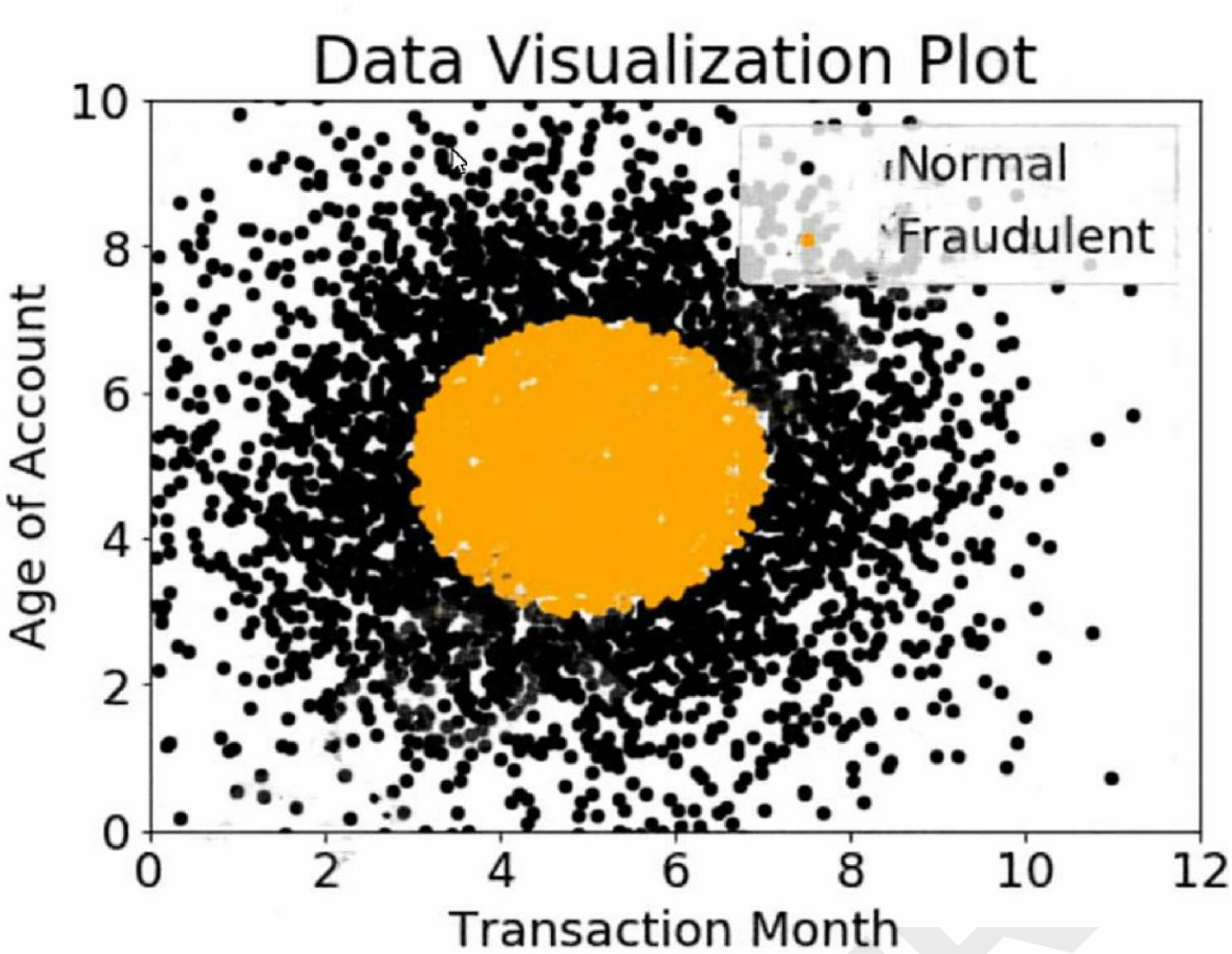
Residual plots directly address the question by visually highlighting the nature of prediction errors. They are a critical diagnostic tool for understanding regression model bias.

Further Reading:

**Residual Plots:**https://www.statology.org/residual-plot/
**Regression Diagnostics:**https://online.stat.psu.edu/stat462/node/132/

## Question: 25

A company wants to classify user behavior as either fraudulent or normal. Based on internal research, a Machine Learning Specialist would like to build a binary classifier based on two features: age of account and transaction month. The class distribution for these features is illustrated in the figure provided.




Based on this information, which model would have the HIGHEST recall with respect to the fraudulent class?

A.Decision tree
B.Linear support vector machine (SVM)
C.Naive Bayesian classifier
D.Single Perceptron with sigmoidal activation function

**Answer: C**

**Explanation:**

C. Naive Bayesian classifier.Naive Bayes classifiers are based on Bayes' theorem and are particularly useful for binary classification problems where the features are categorical or numerical. They are known to perform well on small datasets and are relatively fast to train and predict.In this case, there are two features, the age of the account and the transaction month, which can be used to classify user behavior as fraudulent or normal. Naive Bayes classifiers are suitable for this type of data as they work well with categorical features.

**Question: 26**

A Machine Learning Specialist kicks off a hyperparameter tuning job for a tree-based ensemble model using Amazon SageMaker with Area Under the ROC Curve
(AUC) as the objective metric. This workflow will eventually be deployed in a pipeline that retrains and tunes hyperparameters each night to model click-through on data that goes stale every 24 hours.
With the goal of decreasing the amount of time it takes to train these models, and ultimately to decrease costs, the Specialist wants to reconfigure the input hyperparameter range(s).
Which visualization will accomplish this?

A.A histogram showing whether the most important input feature is Gaussian.

B.A scatter plot with points colored by target variable that uses t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize the large number of input variables in an easier-to-read dimension.

C.A scatter plot showing the performance of the objective metric over each training iteration.

D.A scatter plot showing the correlation between maximum tree depth and the objective metric.

**Answer: D**

**Explanation:**

The goal is to reduce training time and cost for a hyperparameter tuning job in SageMaker, specifically for a tree-based model using AUC as the objective metric, which is retrained daily. To optimize the hyperparameter range, the ML Specialist needs to understand how different hyperparameter values impact the objective metric (AUC).

Option A is incorrect because a histogram of input features, while useful for understanding data distribution, doesn't directly show the relationship between hyperparameters and the objective metric during the tuning process.

Option B is incorrect. While t-SNE can reduce dimensionality for visualization, it's primarily for understanding data structure and clustering, not hyperparameter optimization. It doesn't illustrate the direct impact of specific hyperparameters on the AUC.

Option C is incorrect because a scatter plot of the objective metric (AUC) over training iterations primarily shows the overall training progress and convergence. It does not directly reveal the correlation between specific hyperparameters and the AUC, which is critical for refining the hyperparameter ranges.

Option D, a scatter plot showing the correlation between maximum tree depth and the objective metric (AUC), is the most appropriate choice. Maximum tree depth is a key hyperparameter for tree-based models. By visualizing how different values of maximum tree depth affect the AUC score, the ML Specialist can identify optimal values. If, for example, the AUC plateaus after a certain tree depth, or even decreases due to overfitting, the Specialist can restrict the hyperparameter search range to lower depths. This reduces the search space and allows the tuning job to converge faster, leading to lower training costs and potentially improved model performance. This direct relationship helps in strategically adjusting the hyperparameter search space.

Here are authoritative links for further research:

**Amazon SageMaker Hyperparameter Tuning:**
https://docs.aws.amazon.com/sagemaker/latest/dg/hyperparameter-tuning-how-it-works.html
**Hyperparameter Tuning Strategies:**https://aws.amazon.com/blogs/machine-learning/tuning-machine-learning-models-with-amazon-sagemaker/

## Question: 27

A Machine Learning Specialist is creating a new natural language processing application that processes a dataset comprised of 1 million sentences. The aim is to then run Word2Vec to generate embeddings of the sentences and enable different types of predictions.
Here is an example from the dataset:
"The quck BROWN FOX jumps over the lazy dog.`
Which of the following are the operations the Specialist needs to perform to correctly sanitize and prepare the data in a repeatable manner? (Choose three.)

A.Perform part-of-speech tagging and keep the action verb and the nouns only.

B.Normalize all words by making the sentence lowercase.

C.Remove stop words using an English stopword dictionary.

D.Correct the typography on "quck" to "quick.

E.One-hot encode all words in the sentence.

F.Tokenize the sentence into words.

**Answer: BCF**

**Explanation:**

The correct answer is B, C, and F. Here's why:

**B. Normalize all words by making the sentence lowercase:** Converting all text to lowercase ensures uniformity. The algorithm will treat "The" and "the" as the same word, reducing dimensionality and improving the model's ability to generalize. This is a standard text preprocessing step.

Text preprocessing - Machine Learning Crash Course

**C. Remove stop words using an English stopword dictionary:** Stop words (e.g., "the," "is," "a") are common words that don't carry significant meaning. Removing them reduces noise in the data, decreases the size of the vocabulary, and speeds up training. Libraries like NLTK provide standard stopword lists.

NLTK Stopwords Documentation

**F. Tokenize the sentence into words:** Tokenization is the process of breaking down the sentence into individual words (tokens). This is a fundamental step for Word2Vec and most NLP tasks because the algorithm operates on individual words.

Tokenization - SpaCy Documentation

**Why the other options are incorrect:**

**A. Perform part-of-speech tagging and keep the action verb and the nouns only:** While POS tagging can be useful in some NLP applications, it's not a necessary step for Word2Vec. Word2Vec focuses on the context of words within sentences, not specifically on their grammatical role. Furthermore, filtering only verbs and nouns might remove important contextual information carried by other word types.

**D. Correct the typography on "quck" to "quick":** While correcting spelling errors is generally beneficial, it's a form of data cleaning that can be addressed through custom scripts or rules as needed. It's not a universal prerequisite for Word2Vec and might not be necessary if the misspelled word is rare. The best option would be to train the model and then fix rare errors if the model performs poorly.

**E. One-hot encode all words in the sentence:** One-hot encoding represents words as sparse vectors, with a dimension for each word in the vocabulary. While one-hot encoding is useful in certain scenarios, Word2Vec generates dense word embeddings directly, making one-hot encoding redundant and computationally expensive. Word2Vec maps words to lower-dimensional vector spaces, capturing semantic relationships between words.

## Question: 28

A company is using Amazon Polly to translate plaintext documents to speech for automated company announcements. However, company acronyms are being mispronounced in the current documents. How should a Machine Learning Specialist address this issue for future documents?

A.Convert current documents to SSML with pronunciation tags.

B.Create an appropriate pronunciation lexicon.

C.Output speech marks to guide in pronunciation.

D.Use Amazon Lex to preprocess the text files for pronunciation

**Answer: B**

**Explanation:**

The correct answer is **B. Create an appropriate pronunciation lexicon.**

Here's why:

Amazon Polly offers the capability to use pronunciation lexicons to customize the pronunciation of specific words or phrases. This is particularly useful for acronyms, proper nouns, or domain-specific terms that Polly might mispronounce by default. A lexicon acts as a mapping between the written form of a word and its desired pronunciation, specified using either IPA (International Phonetic Alphabet) or X-SAMPA. When Polly processes text, it consults the lexicon and applies the defined pronunciations.

Option A, "Convert current documents to SSML with pronunciation tags," while also a valid approach, is more suitable for isolated cases or specific documents. SSML (Speech Synthesis Markup Language) allows fine-grained control over speech synthesis, including pronunciation through the <phoneme> tag. However, manually adding pronunciation tags to every instance of an acronym across many documents is cumbersome and not scalable. Creating a lexicon provides a centralized and reusable solution.

Option C, "Output speech marks to guide in pronunciation," is incorrect. Speech marks provide timing information and phonetic data about the synthesized speech, but they don't influence the pronunciation itself. They are more useful for synchronizing speech with other media, such as animations or subtitles.

Option D, "Use Amazon Lex to preprocess the text files for pronunciation," is also incorrect. Amazon Lex is a service for building conversational interfaces (chatbots) using speech and text. While Lex can understand spoken language, it's not designed for general text preprocessing for the purpose of improving pronunciation in Polly. Although Lex can utilize custom vocabulary for better speech recognition, this is not applicable in this scenario where the goal is to correct Polly's speech synthesis.

Therefore, creating a pronunciation lexicon is the most efficient and scalable way to address the mispronunciation of company acronyms in Amazon Polly. It allows for centralized management of pronunciation rules and ensures consistent pronunciation across all documents processed by Polly.

Supporting Documentation:

**Amazon Polly Lexicons:**https://docs.aws.amazon.com/polly/latest/dg/dg-managing-lexicons.html
**Amazon Polly SSML:**https://docs.aws.amazon.com/polly/latest/dg/supportedtags.html

## Question: 29

An insurance company is developing a new device for vehicles that uses a camera to observe drivers' behavior and alert them when they appear distracted. The company created approximately 10,000 training images in a controlled environment that a Machine Learning Specialist will use to train and evaluate machine learning models. During the model evaluation, the Specialist notices that the training error rate diminishes faster as the number of epochs increases and the model is not accurately inferring on the unseen test images.
Which of the following should be used to resolve this issue? (Choose two.)

A.Add vanishing gradient to the model.

B.Perform data augmentation on the training data.

C.Make the neural network architecture complex.

D.Use gradient checking in the model.

E.Add L2 regularization to the model.

**Answer: BE**

**Explanation:**

The problem described indicates overfitting, where the model performs well on training data but poorly on unseen data. Options B and E directly address this.

**Option B: Perform data augmentation on the training data.** Data augmentation artificially increases the size of the training dataset by creating modified versions of existing images (e.g., rotations, flips, zooms). This exposes the model to a wider variety of examples and helps it generalize better, reducing overfitting. By augmenting the original 10,000 images, the model becomes more robust to variations in driver behavior captured by the camera, improving its performance on unseen test images.

**Option E: Add L2 regularization to the model.** L2 regularization adds a penalty term to the loss function that is proportional to the square of the magnitude of the weights. This encourages the model to learn smaller weights, which simplifies the model and reduces its sensitivity to noise in the training data, therefore mitigating overfitting. It penalizes large weights, preventing the model from memorizing the training data.

**Why other options are less suitable:**

**A. Add vanishing gradient to the model.** Vanishing gradients are the opposite problem (model not learning effectively), not overfitting.

**C. Make the neural network architecture complex.** Increasing the complexity of the model would likely exacerbate the overfitting problem. A more complex model has more parameters and is more capable of memorizing the training data.

**D. Use gradient checking in the model.** Gradient checking verifies the correctness of the backpropagation algorithm but does not directly address overfitting.

In summary, data augmentation increases the diversity of the training data, while L2 regularization simplifies the model, both contributing to better generalization and reducing overfitting.

**Supporting Links:**

**Data Augmentation:**https://www.tensorflow.org/tutorials/images/data_augmentation
**Regularization:**https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/l2-regularization

## Question: 30

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: CEF**

**Explanation:**

The correct answer is CEF because these parameters are essential for SageMaker to successfully train a model using built-in algorithms. Let's break down why each choice is crucial.

**C. The IAM role:** SageMaker needs permissions to access resources on your behalf, such as reading training data from S3, writing the trained model back to S3, and potentially accessing other AWS services. The IAM role grants SageMaker these necessary permissions. Without the correct IAM role, the training job will fail due to authorization errors. IAM Role documentation explains how to create and manage IAM roles for SageMaker.

**E. The Amazon EC2 instance class:** This parameter specifies the type of compute instance (CPU or GPU) that SageMaker will use for training. This selection significantly impacts performance and cost. Choosing a suitable instance depends on the algorithm and data size; GPU instances are often preferred for deep learning models due to their parallel processing capabilities. Amazon EC2 Instance Types provides an overview of available instances and their specifications.

**F. The output path:** This parameter tells SageMaker where to save the trained model (the model artifacts) in an S3 bucket after the training job completes. If this isn't specified, the model won't be persisted, and you won't be able to deploy it for inference. It acts as the final destination for your training process.

Now let's look at why the other options are less critical or not always required:

**A. The training channel:** While a training channel (pointing to training data in S3) is fundamental for model training, built-in algorithms require specific channel names like 'train' for training data, which SageMaker expects. This differs from custom training scripts where you explicitly define how channels are named and used.

**B. The validation channel:** A validation channel is optional. Although using a validation dataset is highly recommended for monitoring model performance during training and preventing overfitting, SageMaker can function and train a model even without a validation channel, using the training data alone.

**D. Hyperparameters in a JSON array:** Hyperparameters are often algorithm-specific and influence how the model learns. However, SageMaker built-in algorithms typically offer default values for most hyperparameters. So, while tuning them can improve model performance, explicitly specifying them isn't always mandatory for the training job to run. You can choose to use the algorithm defaults.

## Question: 31

A monitoring service generates 1 TB of scale metrics record data every minute. A Research team performs queries on this data using Amazon Athena. The queries run slowly due to the large volume of data, and the team requires better performance.
How should the records be stored in Amazon S3 to improve query performance?

A.CSV files
B.Parquet files
C.Compressed JSON
D.RecordIO

**Answer: B**

**Explanation:**

Here's a detailed justification for why Parquet files (Option B) are the best choice for improving query performance in this scenario, along with explanations and relevant links:

The core problem is that the Research team's queries are slow due to the sheer volume of data (1 TB/minute) when using Amazon Athena. Athena's performance hinges on efficient data scanning. The goal is to minimize the amount of data Athena needs to read to answer the queries.

Parquet is a columnar storage format. Unlike row-based formats like CSV or JSON, Parquet stores data by columns. This is crucial because analytical queries (like those run by the Research team) often only need to access a subset of columns. With Parquet, Athena can read only the specific columns required for the query, dramatically reducing I/O and processing time. This principle is known as "columnar projection."

CSV files (Option A) and JSON files (Option C) are row-based formats. Athena would have to scan entire rows even if the query only needs one or two columns, making them inefficient for large datasets. Compressing JSON (Option C) helps reduce storage space but doesn't address the fundamental problem of inefficient scanning because it's still a row-based format.

RecordIO (Option D) is a format designed primarily for sequential reading of records, which is common in machine learning training pipelines. While it can be used for storage, it's not optimized for the kind of ad-hoc analytical queries that Athena is intended for. It also doesn't offer the column projection benefits of Parquet.

Furthermore, Parquet supports efficient data compression and encoding schemes, further reducing storage space and improving query performance. Athena can leverage these optimizations natively. Parquet files are also self-describing, meaning they include metadata about the schema, allowing Athena to understand the data structure without requiring external schema definitions in some cases.

In summary, Parquet's columnar storage, efficient compression, and ability to facilitate columnar projection make it the ideal choice for improving Athena query performance on large-scale datasets of metrics. Other formats don't offer the same level of optimization for analytical workloads.

**Authoritative Links:**

**Apache Parquet:**https://parquet.apache.org/
**Amazon Athena Best Practices:**https://docs.aws.amazon.com/athena/latest/ug/best-practices.html (search for "columnar storage" or "Parquet")
**AWS Big Data Blog - Top 10 Performance Tuning Tips for Amazon Athena:**
https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/ (Tip #1 is about using columnar formats)

## Question: 32

Machine Learning Specialist is working with a media company to perform classification on popular articles from the company's website. The company is using random forests to classify how popular an article will be before it is published. A sample of the data being used is below.

| Article_Title | Author | Top_Keywords | Day_Of_Week | URL_of_Article | Page_Views |
|---|---|---|---|---|---|
| Building a Big Data Platform | Jane Doe | Big Data, Spark, Hadoop | Tuesday | http://examplecorp.com/data_platform.html | 1300456 |
| Getting Started with Deep Learning | John Doe | Deep Learning, Machine Learning, Spark | Tuesday | http://examplecorp.com/started_deep_learning.html | 1230661 |
| MXNet ML Guide | Jane Doe | Machine Learning, MXNet, Logistic Regression | Thursday | http://examplecorp.com/mxnet_guide.html | 937291 |
| Intro to NoSQL Databases | Mary Major | NoSQL, Operations, Database | Monday | http://examplecorp.com/nosql_intro_guide.html | 407812 |

Given the dataset, the Specialist wants to convert the Day_Of_Week column to binary values. What technique should be used to convert this column to binary values?

A.Binarization

B.One-hot encoding

C.Tokenization

D.Normalization transformation

**Answer: B**

**Explanation:**

Correct answer is B:One-hot encoding.

## Question: 33

A gaming company has launched an online game where people can start playing for free, but they need to pay if they choose to use certain features. The company needs to build an automated system to predict whether or not a new user will become a paid user within 1 year. The company has gathered a labeled dataset from 1 million users.
The training dataset consists of 1,000 positive samples (from users who ended up paying within 1 year) and 999,000 negative samples (from users who did not use any paid features). Each data sample consists of 200 features including user age, device, location, and play patterns.
Using this dataset for training, the Data Science team trained a random forest model that converged with over 99% accuracy on the training set. However, the prediction results on a test dataset were not satisfactory Which of the following approaches should the Data Science team take to mitigate this issue? (Choose two.)

A.Add more deep trees to the random forest to enable the model to learn more features.

B.Include a copy of the samples in the test dataset in the training dataset.

C.Generate more positive samples by duplicating the positive samples and adding a small amount of noise to the duplicated data.

D.Change the cost function so that false negatives have a higher impact on the cost value than false positives.

E.Change the cost function so that false positives have a higher impact on the cost value than false negatives.

**Answer: CD**

**Explanation:**

The provided scenario describes a classic case of class imbalance, where the number of negative samples (999,000) significantly outweighs the number of positive samples (1,000). The 99% accuracy on the training set is misleading because the model is likely predicting nearly everything as negative, achieving high accuracy simply by correctly classifying the vast majority of negative examples. This is a critical problem, particularly for the gaming company, as it wants to correctly identify users likely to convert to paying customers (positive samples).

Let's justify why options C and D are the correct choices:

**C. Generate more positive samples by duplicating the positive samples and adding a small amount of noise to the duplicated data.** This is an oversampling technique, specifically a form of data augmentation. By creating synthetic positive samples, the model is exposed to a wider range of variations within the positive class, helping it to learn the characteristics of positive users more effectively. Adding noise helps to prevent overfitting to the original positive samples. This addresses the core issue of class imbalance by increasing the representation of the minority class. It helps the model to better distinguish between positive and negative samples and improve the generalization to unseen data. More information on oversampling techniques can be found here: https://developers.google.com/machine-learning/data-prep/transform/resample

**D. Change the cost function so that false negatives have a higher impact on the cost value than false positives.** Because the business goal focuses on identifying potential paying customers, misclassifying a

paying customer as a non-paying one (false negative) is more costly than misclassifying a non-paying user as a paying one (false positive). By assigning a higher cost to false negatives in the cost function (e.g., using a weighted loss function), the model will be penalized more severely for failing to identify potential paying customers. This adjustment encourages the model to prioritize recall (identifying all positive cases) over precision (avoiding false positives), which is appropriate given the business context. This technique is explained in detail in https://www.tensorflow.org/tutorials/structured_data/imbalanced_data.

Now let's analyze why the other options are incorrect:

**A. Add more deep trees to the random forest to enable the model to learn more features.** While increasing the complexity of the model might help in some cases, it is more likely to exacerbate overfitting to the majority class, especially in the presence of severe class imbalance. The fundamental issue is not the lack of learning capacity but the skewed distribution of classes.

**B. Include a copy of the samples in the test dataset in the training dataset.** This is a severe mistake that leads to data leakage. The model will be unfairly trained on data it will later be tested on, resulting in an artificially inflated and unrealistic performance evaluation.

**E. Change the cost function so that false positives have a higher impact on the cost value than false negatives.** This would worsen the situation. False positives are less detrimental to the gaming company, so increasing their cost is counterproductive.

## Question: 34

A Data Scientist is developing a machine learning model to predict future patient outcomes based on information collected about each patient and their treatment plans. The model should output a continuous value as its prediction. The data available includes labeled outcomes for a set of 4,000 patients. The study was conducted on a group of individuals over the age of 65 who have a particular disease that is known to worsen with age.
Initial models have performed poorly. While reviewing the underlying data, the Data Scientist notices that, out of 4,000 patient observations, there are 450 where the patient age has been input as 0. The other features for these observations appear normal compared to the rest of the sample population
How should the Data Scientist correct this issue?

A.Drop all records from the dataset where age has been set to 0.
B.Replace the age field value for records with a value of 0 with the mean or median value from the dataset
C.Drop the age feature from the dataset and train the model using the rest of the features.
D.Use k-means clustering to handle missing features

**Answer: D**

**Explanation:**

The correct approach to handle the 450 instances of age being 0 is to use a more sophisticated imputation technique, specifically k-means clustering, as option D suggests. While options A and C involve data loss which should generally be avoided unless absolutely necessary, and option B uses a simplistic imputation which can lead to biases.

Here's why k-means clustering is a better choice in this scenario:

1. **Preserves Data:** Unlike dropping the records (A) or the entire age feature (C), k-means clustering maintains all data points for training. This is crucial, especially given the relatively small dataset size of 4,000.
2. **Contextual Imputation:**k-means clustering leverages the relationships between all features to impute the missing age values. It groups similar patients based on their other features, and then uses the average age within the cluster to estimate the missing values. This is more informed than simply using the overall mean or median (B).

3. **Reduces Bias:** Replacing missing values with the mean or median introduces bias. This is because it does not consider the specific characteristics of the patient whose age is missing. k-means clustering helps to mitigate this by imputing values based on similar patient profiles.

4. **Captures Non-Linear Relationships:**k-means, being a clustering algorithm, can implicitly capture non-linear relationships between features, potentially leading to better imputation accuracy.

5. **Domain Relevance:** Given that the disease worsens with age, age is a crucial feature for the model. Therefore, it is preferable to impute missing values accurately rather than discard the feature entirely.

**Why the other options are less suitable:**

**A (Dropping Records):** Removing 450 records (over 10% of the data) is significant data loss, particularly problematic with a dataset of only 4,000. This can severely impact the model's accuracy and generalizability.

**B (Mean/Median Imputation):** Replacing missing age values with the mean or median age ignores the relationships between age and other features. This can introduce significant bias and lead to inaccurate predictions.

**C (Dropping the Age Feature):** Given the domain knowledge that the disease worsens with age, dropping the age feature would be a detrimental loss of valuable information that could improve model performance.

In summary, using k-means clustering for imputation is the most appropriate choice because it preserves data, considers feature relationships, reduces bias, and retains a critical feature for the model. This results in a more accurate and reliable model for predicting patient outcomes.

**Authoritative Links:**

**Handling Missing Data:**https://developers.google.com/machine-learning/data-prep/transform/missing-values
**Imputation Techniques:**https://scikit-learn.org/stable/modules/impute.html
**k-Means Clustering:**https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

## Question: 35

A Data Science team is designing a dataset repository where it will store a large amount of training data commonly used in its machine learning models. As Data
Scientists may create an arbitrary number of new datasets every day, the solution has to scale automatically and be cost-effective. Also, it must be possible to explore the data using SQL.
Which storage scheme is MOST adapted to this scenario?

A.Store datasets as files in Amazon S3.

B.Store datasets as files in an Amazon EBS volume attached to an Amazon EC2 instance.

C.Store datasets as tables in a multi-node Amazon Redshift cluster.

D.Store datasets as global tables in Amazon DynamoDB.

**Answer: A**

**Explanation:**

The best storage scheme for the Data Science team's dataset repository is **A. Store datasets as files in Amazon S3.** Here's why:

**Scalability:** Amazon S3 is designed for virtually unlimited storage capacity. It can automatically scale to accommodate the daily creation of new datasets without manual intervention. https://aws.amazon.com/s3/ **Cost-Effectiveness:** S3 offers various storage classes (Standard, Intelligent-Tiering, Standard-IA, Glacier) optimized for different access patterns and cost requirements. This allows the team to store less frequently

accessed datasets in cheaper tiers, optimizing overall costs.

**SQL Exploration:** While S3 itself isn't a SQL database, services like Amazon Athena and Amazon Redshift Spectrum enable querying data directly in S3 using SQL. This allows Data Scientists to explore the data without moving it to a separate database. https://aws.amazon.com/athena/ , https://aws.amazon.com/redshift/spectrum/

**Data Lake Foundation:** S3 is commonly used as the foundation for data lakes due to its scalability, cost-effectiveness, and integration with other AWS services. It supports storing various data formats suitable for machine learning models.

**Alternative B (EBS):** EBS volumes are attached to specific EC2 instances and do not offer the same level of automatic scalability or cost-effectiveness as S3. Managing storage across multiple EBS volumes can become complex.

**Alternative C (Redshift):** Redshift is primarily a data warehouse for structured data. While it can store datasets, it's less suitable for storing arbitrary files and might be more expensive for this purpose compared to S3, especially given the potentially high volume of new datasets. Redshift is optimized for analytical queries, but for simple exploration using SQL, Athena over S3 is more appropriate.

**Alternative D (DynamoDB):** DynamoDB is a NoSQL database designed for fast key-value lookups. It's not suitable for storing large datasets or performing complex SQL queries. Also, storing entire datasets in DynamoDB would be very expensive.

Therefore, S3 offers the best combination of scalability, cost-effectiveness, and SQL exploration capabilities for the Data Science team's needs.

## Question: 36

A Machine Learning Specialist deployed a model that provides product recommendations on a company's website. Initially, the model was performing very well and resulted in customers buying more products on average. However, within the past few months, the Specialist has noticed that the effect of product recommendations has diminished and customers are starting to return to their original habits of spending less. The Specialist is unsure of what happened, as the model has not changed from its initial deployment over a year ago.
Which method should the Specialist try to improve model performance?

A.The model needs to be completely re-engineered because it is unable to handle product inventory changes.

B.The model's hyperparameters should be periodically updated to prevent drift.

C.The model should be periodically retrained from scratch using the original data while adding a regularization term to handle product inventory changes

D.The model should be periodically retrained using the original training data plus new data as product inventory changes.

**Answer: D**

**Explanation:**

The correct answer is D because the scenario describes a classic case of model drift. Model drift occurs when the statistical properties of the target variable, or the input features, change over time, leading to a decline in the model's predictive performance. In this case, customer behavior related to product recommendations has changed since the model was initially trained. This could be due to new trends, new products, competitor strategies, or a change in the customer demographic.

Option A is incorrect because re-engineering the entire model is often overkill for addressing model drift. Retraining is usually the first step.

Option B suggests updating hyperparameters, but this only addresses the model's internal settings and not the fundamental changes in the underlying data distribution. Hyperparameter tuning addresses model optimization with the same input data distribution but will not solve the drift problem.

Option C suggests retraining with original data and adding a regularization term. While regularization helps prevent overfitting, it won't address the core issue of the model not being exposed to the new data patterns reflecting recent customer behavior and product inventory changes. The original data is stale and no longer represents the current reality.

Retraining the model periodically with new data (Option D) allows the model to learn the current patterns and trends in customer behavior. This ensures that the model's predictions remain relevant and accurate over time, combating the effects of model drift. This is a common and effective strategy in machine learning model maintenance. By incorporating new data that reflects current product inventory changes and customer preferences, the model is better equipped to adapt to evolving market conditions and maintain its
performance.

For further information, consider researching the following topics:

**Model Drift:**https://www.evidentlyai.com/blog/model-drift-detection
**Retraining strategies:** AWS documentation on model retraining.
**Concept Drift:**https://towardsdatascience.com/handling-concept-drift-in-machine-learning-65ff54a15b16

## Question: 37

A Machine Learning Specialist working for an online fashion company wants to build a data ingestion solution for the company's Amazon S3-based data lake.
The Specialist wants to create a set of ingestion mechanisms that will enable future capabilities comprised of:☞ Real-time analytics
☞ Interactive analytics of historical data
☞ Clickstream analytics
☞ Product recommendations
Which services should the Specialist use?

A.AWS Glue as the data catalog; Amazon Kinesis Data Streams and Amazon Kinesis Data Analytics for real-time data insights; Amazon Kinesis Data Firehose for delivery to Amazon ES for clickstream analytics; Amazon EMR to generate personalized product recommendations

B.Amazon Athena as the data catalog: Amazon Kinesis Data Streams and Amazon Kinesis Data Analytics for near-real-time data insights; Amazon Kinesis Data Firehose for clickstream analytics; AWS Glue to generate personalized product recommendations

C.AWS Glue as the data catalog; Amazon Kinesis Data Streams and Amazon Kinesis Data Analytics for historical data insights; Amazon Kinesis Data Firehose for delivery to Amazon ES for clickstream analytics; Amazon EMR to generate personalized product recommendations

D.Amazon Athena as the data catalog; Amazon Kinesis Data Streams and Amazon Kinesis Data Analytics for historical data insights; Amazon DynamoDB streams for clickstream analytics; AWS Glue to generate personalized product recommendations

**Answer: A**

**Explanation:**

The optimal solution leverages a combination of AWS services, each tailored for specific data ingestion and analytics needs. AWS Glue serves as the central data catalog, providing a metadata repository that allows various services to discover and understand the data stored in S3. Kinesis Data Streams is ideal for ingesting real-time data streams, and Kinesis Data Analytics provides the capability to process and analyze these streams in real-time, supporting immediate insights. Kinesis Data Firehose is well-suited for capturing and loading clickstream data into Amazon Elasticsearch Service (Amazon ES), enabling clickstream analytics. Amazon EMR offers a scalable platform for running big data processing frameworks like Spark and Hadoop, crucial for generating personalized product recommendations, a computationally intensive task.

Option B incorrectly uses Athena as the data catalog; while Athena allows querying data in S3, Glue is the preferred service for managing metadata and enabling data discovery across various AWS services.

Furthermore, Glue is not typically used to directly generate personalized product recommendations. Option C incorrectly associates Kinesis Data Streams and Analytics with historical data insights. These services are designed for real-time processing, not historical analysis. Option D makes incorrect use of DynamoDB Streams for clickstream analytics. DynamoDB Streams are typically for tracking changes to data in DynamoDB tables, not for ingesting high-velocity clickstream data. Also, Athena as catalog is not ideal.

Therefore, option A offers the most appropriate combination of services to address all the requirements.

Supporting Links:

AWS Glue: https://aws.amazon.com/glue/
Amazon Kinesis Data Streams: https://aws.amazon.com/kinesis/data-streams/
Amazon Kinesis Data Analytics: https://aws.amazon.com/kinesis/data-analytics/
Amazon Kinesis Data Firehose: https://aws.amazon.com/kinesis/data-firehose/
Amazon EMR: https://aws.amazon.com/emr/

**Question: 38**

A company is observing low accuracy while training on the default built-in image classification algorithm in Amazon SageMaker. The Data Science team wants to use an Inception neural network architecture instead of a ResNet architecture.
Which of the following will accomplish this? (Choose two.)

A.Customize the built-in image classification algorithm to use Inception and use this for model training.

B.Create a support case with the SageMaker team to change the default image classification algorithm to Inception.

C.Bundle a Docker container with TensorFlow Estimator loaded with an Inception network and use this for model training.

D.Use custom code in Amazon SageMaker with TensorFlow Estimator to load the model with an Inception network, and use this for model training.

E.Download and apt-get install the inception network code into an Amazon EC2 instance and use this instance as a Jupyter notebook in Amazon SageMaker.

**Answer: CD**

**Explanation:**

The question addresses adapting the Amazon SageMaker built-in image classification algorithm when it yields low accuracy and the Data Science team prefers an Inception neural network architecture over the default ResNet.

Option C is correct because bundling a Docker container offers a high degree of customization and control over the training environment. By including a TensorFlow Estimator loaded with an Inception network, the team can directly define and train the model using the desired architecture. This approach ensures compatibility and avoids limitations of the built-in algorithm. [Docker & SageMaker integration: https://docs.aws.amazon.com/sagemaker/latest/dg/docker-container.html]

Option D is also correct. Using custom code within Amazon SageMaker with TensorFlow Estimator gives a level of flexibility similar to using a Docker container but might be easier to implement initially. The team can define the Inception network architecture using TensorFlow within a SageMaker script, leverage SageMaker's infrastructure for training, and manage dependencies easily. This aligns well with the company's requirement to move away from the built-in algorithm and train the model using Inception. [SageMaker with TensorFlow: https://aws.amazon.com/blogs/machine-learning/bring-your-own-tensorflow-model-to-amazon-sagemaker/]

Option A is incorrect because customizing the built-in image classification algorithm beyond its intended parameters can be difficult or impossible. The built-in algorithms are designed with specific architectures in

mind, and attempting to fundamentally alter them might not be supported.

Option B is incorrect. AWS support doesn't alter the default behaviors of pre-built algorithms based on user preferences. Their support focuses on assisting with usage and troubleshooting existing features.

Option E is incorrect because directly installing the Inception network code into an EC2 instance and using it as a Jupyter notebook within SageMaker doesn't directly integrate the Inception model with SageMaker's training and deployment functionalities. While it enables experimentation, it doesn't leverage SageMaker's scalability and managed environment for training.

**Question: 39**

A Machine Learning Specialist built an image classification deep learning model. However, the Specialist ran into an overfitting problem in which the training and testing accuracies were 99% and 75%, respectively.
How should the Specialist address this issue and what is the reason behind it?

A.The learning rate should be increased because the optimization process was trapped at a local minimum.

B.The dropout rate at the flatten layer should be increased because the model is not generalized enough.

C.The dimensionality of dense layer next to the flatten layer should be increased because the model is not complex enough.

D.The epoch number should be increased because the optimization process was terminated before it reached the global minimum.

**Answer: B**

**Explanation:**

The correct answer is B: Increase the dropout rate at the flatten layer.

Overfitting, indicated by high training accuracy (99%) and significantly lower testing accuracy (75%), signifies that the model is memorizing the training data rather than learning generalizable patterns. It's fitting the noise in the training data.

**Why B is correct:**

**Dropout:** Dropout is a regularization technique used in neural networks. It randomly disables a fraction of neurons during each training iteration. This forces the network to learn more robust and independent features, preventing over-reliance on specific neurons. It essentially creates multiple "thinned" versions of the network.

**Flatten Layer:** The flatten layer in a convolutional neural network (CNN) transforms the multi-dimensional feature maps from the convolutional layers into a one-dimensional vector before feeding them into the fully connected (dense) layers. This layer is a key transition point where complex features extracted by the convolutional layers are used for classification. A high dropout rate at the flatten layer helps prevent overfitting by forcing the subsequent dense layers to learn from a more diverse set of features.

**Generalization:** By increasing the dropout rate, the model is forced to become more generalized. It becomes less sensitive to specific features in the training data and more capable of performing well on unseen data. This is exactly what's needed to address the overfitting issue.

**Why other options are incorrect:**

**A (Increasing Learning Rate):** Increasing the learning rate can sometimes help escape local minima, but it more often exacerbates overfitting. A high learning rate can cause the model to bounce around and miss the optimal solution, making it less stable.

**C (Increasing Dimensionality of Dense Layer):** Increasing the dimensionality of a dense layer makes the model more complex. A more complex model is more likely to overfit, not less.

**D (Increasing Epoch Number):** Increasing the epoch number will train the model for longer. If the model is already overfitting, training for longer will only worsen the problem. It will overfit even more to the training data.

**Supporting Cloud Concepts:**

Deep learning models, particularly for image classification, require substantial computational resources. AWS provides various services like EC2 instances with GPUs or specialized machine learning services like SageMaker to train these models. SageMaker also supports various regularization techniques and
hyperparameter tuning, which are crucial in addressing overfitting issues like this.

**Authoritative Links:**

**Dropout Regularization:** Dropout: A Simple Way to Prevent Neural Networks from Overfitting
https://jmlr.org/papers/v15/srivastava14a.html
**Overfitting:**https://www.ibm.com/docs/en/cloud-private/3.1.2?topic=terms-overfitting **AWS SageMaker:**https://aws.amazon.com/sagemaker/

## Question: 40

A Machine Learning team uses Amazon SageMaker to train an Apache MXNet handwritten digit classifier model using a research dataset. The team wants to receive a notification when the model is overfitting. Auditors want to view the Amazon SageMaker log activity report to ensure there are no unauthorized API calls.
What should the Machine Learning team do to address the requirements with the least amount of code and fewest steps?

A.Implement an AWS Lambda function to log Amazon SageMaker API calls to Amazon S3. Add code to push a custom metric to Amazon CloudWatch. Create an alarm in CloudWatch with Amazon SNS to receive a notification when the model is overfitting.

B.Use AWS CloudTrail to log Amazon SageMaker API calls to Amazon S3. Add code to push a custom metric to Amazon CloudWatch. Create an alarm in CloudWatch with Amazon SNS to receive a notification when the model is overfitting.

C.Implement an AWS Lambda function to log Amazon SageMaker API calls to AWS CloudTrail. Add code to push a custom metric to Amazon CloudWatch. Create an alarm in CloudWatch with Amazon SNS to receive a notification when the model is overfitting.

D.Use AWS CloudTrail to log Amazon SageMaker API calls to Amazon S3. Set up Amazon SNS to receive a notification when the model is overfitting

**Answer: B**

**Explanation:**

The correct answer is B. Here's why:

**Auditing API Calls (Unauthorized API Calls):** AWS CloudTrail is the best service for logging API calls made to AWS services, including SageMaker. It provides an audit trail of who did what, when, and from where. Implementing a Lambda function (options A and C) for this purpose would be redundant and less efficient than using CloudTrail, which is specifically designed for this.

**Overfitting Notification:** The question requires notification upon detecting overfitting. The best way to detect overfitting is to monitor model performance metrics (e.g., validation loss) during training.

**Custom Metric in CloudWatch:** The proposed solution involves pushing custom metrics (specifically, metrics indicating overfitting) to Amazon CloudWatch. This is essential for tracking model performance.

**CloudWatch Alarm & SNS:** A CloudWatch alarm can be configured to trigger when a specified metric crosses a threshold, indicating overfitting. By connecting this alarm to an Amazon SNS topic, the team can receive

notifications when the overfitting condition is met.

Let's analyze why the other options are less suitable:

**Option A & C (Using Lambda to log API Calls):** Both options A and C suggest using Lambda for logging SageMaker API calls. This introduces unnecessary complexity, code, and operational overhead when AWS CloudTrail is the native and recommended solution for this purpose. CloudTrail is a fully managed service specifically designed for logging API activity, making it the simplest and most efficient approach. Option C also proposes logging to CloudTrail using Lambda, which is logically incorrect as CloudTrail already captures the logs directly.

**Option D (Missing Custom Metric Push):** Option D lacks the critical step of pushing a custom metric that indicates overfitting to CloudWatch. Without this metric, a CloudWatch alarm cannot be created to trigger notifications related to overfitting. Simply logging API calls to S3 doesn't address the requirement of being notified about overfitting during training.

In conclusion, Option B effectively addresses both the auditing requirements and the overfitting notification requirement with the least amount of code and fewest steps. It leverages native AWS services like CloudTrail and CloudWatch, which are designed for these specific purposes.

**Supporting Links:**

**AWS CloudTrail:**https://aws.amazon.com/cloudtrail/
**Amazon CloudWatch:**https://aws.amazon.com/cloudwatch/
**Amazon SNS:**https://aws.amazon.com/sns/

## Question: 41

A Machine Learning Specialist is building a prediction model for a large number of features using linear models, such as linear regression and logistic regression.
During exploratory data analysis, the Specialist observes that many features are highly correlated with each other. This may make the model unstable.
What should be done to reduce the impact of having such a large number of features?

A.Perform one-hot encoding on highly correlated features.
B.Use matrix multiplication on highly correlated features.
C.Create a new feature space using principal component analysis (PCA)
D.Apply the Pearson correlation coefficient.

**Answer: C**

**Explanation:**

The correct answer is C: Create a new feature space using principal component analysis (PCA). Here's why:

The problem describes multicollinearity, where features are highly correlated. Multicollinearity in linear models like linear and logistic regression leads to unstable coefficient estimates. This means the model's parameters become sensitive to small changes in the data, making it difficult to interpret feature importance and potentially harming the model's generalization performance.

PCA addresses this by transforming the original features into a new set of uncorrelated features called principal components. These components are linear combinations of the original features, ordered by the amount of variance they explain in the data. By selecting the top principal components, the specialist can reduce the dimensionality of the feature space while retaining most of the important information. This reduces multicollinearity, stabilizes the model, and can improve its performance.

Option A, one-hot encoding, expands categorical features into multiple binary features. While useful for handling categorical data, it would actually increase the number of features, exacerbating the multicollinearity problem, not solving it. It's not relevant to correlated numerical features, which is the core issue.

Option B, using matrix multiplication on highly correlated features, does not address multicollinearity. Matrix multiplication is a fundamental linear algebra operation but offers no inherent solution to feature correlation. It might even amplify the problem depending on how it's applied.

Option D, applying the Pearson correlation coefficient, is a diagnostic tool for detecting multicollinearity, not a solution. It helps identify which features are correlated, but it doesn't reduce the number of features or address the model instability caused by the correlation.

PCA effectively mitigates multicollinearity by creating uncorrelated features, reducing the dimensionality of the feature space, and stabilizing the model.

Further reading:

**Principal Component Analysis (PCA):**https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
**Multicollinearity:**https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/multicollinearity/

## Question: 42

A Machine Learning Specialist is implementing a full Bayesian network on a dataset that describes public transit in New York City. One of the random variables is discrete, and represents the number of minutes New Yorkers wait for a bus given that the buses cycle every 10 minutes, with a mean of 3 minutes.
Which prior probability distribution should the ML Specialist use for this variable?

A.Poisson distribution
B.Uniform distribution
C.Normal distribution
D.Binomial distribution

**Answer: A**

**Explanation:**

The Poisson distribution is the most suitable prior probability distribution for the given scenario due to its properties relating to count data and waiting times in contexts like public transportation.

The random variable represents the number of minutes New Yorkers wait for a bus. This is essentially counting the occurrences of a specific event (the bus arriving) within a given interval (10 minutes cycle). The Poisson distribution models the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known average rate and independently of the time since the last event. The mean of 3 minutes aligns directly with the expected value (lambda) of the Poisson distribution.

While the uniform distribution could represent a random waiting time within the 10-minute cycle, it doesn't incorporate the provided information about the mean waiting time. The normal distribution, while common, is best suited for continuous variables and is not ideal for modeling discrete counts of waiting time. The binomial distribution models the number of successes in a fixed number of trials. While related to counts, it does not directly model waiting times like the Poisson distribution, which captures the frequency of events within an interval. The Poisson distribution, characterized by its single parameter (lambda) representing the average rate, can be easily updated with new data through Bayesian inference, making it an appropriate choice for a

prior.

Therefore, the Poisson distribution is most appropriate for modeling the number of minutes New Yorkers wait for a bus, given a mean waiting time, as it models count data representing events within a specified interval, such as a bus cycle.https://en.wikipedia.org/wiki/Poisson_distributionhttps://www.statisticshowto.com/probability-and-statistics/distributions/poisson-distribution/

**Question: 43**

A Data Science team within a large company uses Amazon SageMaker notebooks to access data stored in Amazon S3 buckets. The IT Security team is concerned that internet-enabled notebook instances create a security vulnerability where malicious code running on the instances could compromise data privacy.
The company mandates that all instances stay within a secured VPC with no internet access, and data communication traffic must stay within the AWS network.
How should the Data Science team configure the notebook instance placement to meet these requirements?

A.Associate the Amazon SageMaker notebook with a private subnet in a VPC. Place the Amazon SageMaker endpoint and S3 buckets within the same VPC.

B.Associate the Amazon SageMaker notebook with a private subnet in a VPC. Use IAM policies to grant access to Amazon S3 and Amazon SageMaker.

C.Associate the Amazon SageMaker notebook with a private subnet in a VPC. Ensure the VPC has S3 VPC endpoints and Amazon SageMaker VPC endpoints attached to it.

D.Associate the Amazon SageMaker notebook with a private subnet in a VPC. Ensure the VPC has a NAT gateway and an associated security group allowing only outbound connections to Amazon S3 and Amazon SageMaker.

**Answer: C**

**Explanation:**

The correct answer is C. To meet the security requirements of a company mandating no internet access for its Amazon SageMaker notebook instances and ensuring all data communication stays within the AWS network, you need to place the notebook instance in a private subnet of a VPC and configure VPC endpoints for both Amazon S3 and SageMaker.

Option C fulfills all the requirements. Associating the notebook with a private subnet ensures it doesn't have a direct public IP address, thus isolating it from the internet. Using S3 VPC endpoints allows the notebook instance, through the AWS network, to access data stored in S3 buckets without traversing the public internet. Similarly, SageMaker VPC endpoints ensure that calls to SageMaker APIs (like training jobs or model deployments) also remain within the AWS network, eliminating any external internet dependencies.

Option A lacks the essential VPC endpoints. While placing the notebook, endpoint, and S3 within the same VPC is good practice, it doesn't guarantee the traffic will remain within the AWS network without VPC endpoints. The notebook instance would still try to use public internet to reach S3 without an S3 VPC endpoint.

Option B is incomplete. IAM policies grant access, but they don't force the traffic to stay within the AWS network. IAM policies authorize access, but VPC endpoints control network routing. While IAM policies are still needed for proper authorization, they are not sufficient by themselves.

Option D introduces a NAT gateway which defeats the purpose of blocking internet access. A NAT gateway allows instances in a private subnet to initiate outbound internet connections. This opens up the same security vulnerability the company is trying to avoid. Even with security group restrictions, a NAT gateway inherently permits external connections, violating the core requirement.

Therefore, using VPC endpoints for S3 and SageMaker combined with a private subnet creates a fully isolated environment where the notebook can securely access data and services while adhering to the company's security policy.

Relevant documentation:

**VPC Endpoints:**https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html
**SageMaker Notebook Instances in a VPC:**https://docs.aws.amazon.com/sagemaker/latest/dg/notebook-vpc.html

**Question: 44**

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: BCF**

**Explanation:**

The scenario describes overfitting, where the model memorizes the training data instead of learning generalizable patterns. The goal is to improve the model's performance on unseen data (the test set).

**B. Increase regularization:** Regularization techniques (L1, L2) add a penalty to the model's complexity, discouraging it from learning the noise in the training data and promoting simpler, more generalizable models.
This combats overfitting. (Source: https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/l1-regularization)

**C. Increase dropout:** Dropout is a regularization technique specific to neural networks. During training, it randomly deactivates some neurons. This prevents neurons from becoming overly reliant on specific features or other neurons, forcing the network to learn more robust and independent representations, thereby reducing overfitting. (Source: https://jmlr.org/v15/srivastava14a.html)

**F. Decrease feature combinations:** Creating too many complex feature combinations can lead to the model fitting the training data very closely, including its noise. This results in overfitting. Reducing the number of feature combinations simplifies the model and makes it more likely to generalize well to unseen data. Overly complex feature engineering, while helpful in some cases, can be a source of overfitting if not carefully managed. By reducing feature combinations, you are essentially reducing the complexity of the model.

Why the other options are incorrect:

**A. Decrease regularization:** Decreasing regularization would likely exacerbate overfitting, allowing the model to become even more complex and memorize the training data better.
**D. Decrease dropout:** Decreasing dropout reduces its regularization effect, which can further promote overfitting in neural networks.
**E. Increase feature combinations:** Increasing feature combinations leads to a more complex model, potentially overfitting the training data.

## Question: 45

A Data Scientist needs to create a serverless ingestion and analytics solution for high-velocity, real-time streaming data. The ingestion process must buffer and convert incoming records from JSON to a query-optimized, columnar format without data loss. The output datastore must be highly available, and Analysts must be able to run SQL queries against the data and connect to existing business intelligence dashboards.
Which solution should the Data Scientist build to satisfy the requirements?

A.Create a schema in the AWS Glue Data Catalog of the incoming data format. Use an Amazon Kinesis Data Firehose delivery stream to stream the data and transform the data to Apache Parquet or ORC format using the AWS Glue Data Catalog before delivering to Amazon S3. Have the Analysts query the data directly from Amazon S3 using Amazon Athena, and connect to BI tools using the Athena Java Database Connectivity (JDBC) connector.

B.Write each JSON record to a staging location in Amazon S3. Use the S3 Put event to trigger an AWS Lambda function that transforms the data into Apache Parquet or ORC format and writes the data to a processed data location in Amazon S3. Have the Analysts query the data directly from Amazon S3 using Amazon Athena, and connect to BI tools using the Athena Java Database Connectivity (JDBC) connector.

C.Write each JSON record to a staging location in Amazon S3. Use the S3 Put event to trigger an AWS Lambda function that transforms the data into Apache Parquet or ORC format and inserts it into an Amazon RDS PostgreSQL database. Have the Analysts query and run dashboards from the RDS database.

D.Use Amazon Kinesis Data Analytics to ingest the streaming data and perform real-time SQL queries to convert the records to Apache Parquet before delivering to Amazon S3. Have the Analysts query the data directly from Amazon S3 using Amazon Athena and connect to BI tools using the Athena Java Database Connectivity (JDBC) connector.

**Answer: A**

**Explanation:**

The correct answer is A. Here's why:

**A is the most suitable solution because:**

**Kinesis Data Firehose for Real-time Ingestion:** Kinesis Data Firehose is designed specifically for real-time streaming data ingestion into AWS data stores. It handles high velocity data effectively.

**Buffering:** Firehose buffers incoming records, fulfilling the requirement of buffering.

**JSON to Columnar Format Conversion:** Firehose can transform JSON data to a query-optimized columnar format like Parquet or ORC using AWS Glue Data Catalog for schema discovery and transformation. **Serverless:** Kinesis Data Firehose is a managed service, which means serverless.

**AWS Glue Data Catalog:** Glue Data Catalog centrally stores metadata (schema definitions) making data discovery and querying easier.

**Amazon S3 for Storage:** S3 provides highly available and durable storage for the transformed data. **Amazon Athena for SQL Querying:** Athena allows analysts to run SQL queries directly against the data in S3 without the need for a database. This fulfills the SQL query requirement.

**BI Tool Connectivity:** Athena's JDBC connector enables seamless integration with existing business intelligence (BI) dashboards.

**Why the other options are less suitable:**

**B: Lambda-based Transformation:** While Lambda can transform data, using Firehose is more efficient and scalable for high-velocity streaming data. S3 Put event triggering Lambda adds unnecessary complexity. Lambda can be used for complex transformations that Kinesis Data Firehose cannot handle.

**C: RDS PostgreSQL:** Using RDS PostgreSQL as the final datastore is not ideal for large-scale analytical workloads. Columnar formats in S3 queried by Athena are better suited for this. Also, inserting individual records into RDS is not as efficient.

**D: Kinesis Data Analytics for Format Conversion:** While Kinesis Data Analytics can perform real-time SQL, it

is primarily for real-time data processing and analytics, not optimal for format conversion at this scale and for storage in a columnar format. It is less cost-effective for simple format conversion.

**Supporting Links:**

**Kinesis Data Firehose:**https://aws.amazon.com/kinesis/data-firehose/
**AWS Glue Data Catalog:**https://aws.amazon.com/glue/
**Amazon Athena:**https://aws.amazon.com/athena/

**Question: 46**

An online reseller has a large, multi-column dataset with one column missing 30% of its data. A Machine Learning Specialist believes that certain columns in the dataset could be used to reconstruct the missing data.
Which reconstruction approach should the Specialist use to preserve the integrity of the dataset?

A.Listwise deletion
B.Last observation carried forward
C.Multiple imputation
D.Mean substitution

**Answer: C**

**Explanation:**

The question concerns handling missing data in a machine learning context to preserve dataset integrity, specifically when some features might predict the missing values. Option C, multiple imputation, is the most suitable approach in this scenario.

Here's why: Multiple imputation involves creating multiple plausible datasets, each with different imputed values for the missing data. The imputation is based on the relationships between the missing variable and other observed variables in the dataset. This effectively leverages the "certain columns" the Specialist suspects can reconstruct missing data. By generating multiple imputed datasets, we acknowledge the uncertainty surrounding the missing values and avoid introducing bias by selecting a single "best guess." This approach leverages the predictive power of other columns to create values for the missing ones.

Option A, listwise deletion, removes entire rows containing missing values. This leads to significant data loss (30% in this case) and can introduce bias if the missingness is not completely random. It ignores the potential to reconstruct the missing data using other features.

Option B, last observation carried forward (LOCF), is primarily used in time-series data where the last known value is assumed to be a reasonable estimate for the missing value. It's inappropriate for general datasets without a temporal component and wouldn't utilize relationships with other columns for reconstruction.

Option D, mean substitution, replaces missing values with the mean of the available data for that column. While simple, it reduces variance and can distort the distribution of the variable, leading to biased statistical analyses and inaccurate machine learning models. It doesn't leverage information from other columns to predict the missing values.

Multiple imputation, on the other hand, maintains the relationships between variables and provides a more accurate representation of the data's distribution, preventing bias. It considers all observed data, therefore adhering to the principle of using other available columns to impute and fill in data. By generating several imputed datasets, it enables better estimation of uncertainty.

Therefore, multiple imputation is the superior choice for reconstructing missing data while preserving the integrity and statistical properties of the dataset in a machine learning context. It's crucial when relationships

between variables hold vital information for accurate data imputation.

Relevant Links:

Multiple Imputation: https://en.wikipedia.org/wiki/Multiple_imputation
Handling Missing Data: https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1c6929c79431

**Question: 47**

A company is setting up an Amazon SageMaker environment. The corporate data security policy does not allow communication over the internet.
How can the company enable the Amazon SageMaker service without enabling direct internet access to Amazon SageMaker notebook instances?

A.Create a NAT gateway within the corporate VPC.
B.Route Amazon SageMaker traffic through an on-premises network.
C.Create Amazon SageMaker VPC interface endpoints within the corporate VPC.
D.Create VPC peering with Amazon VPC hosting Amazon SageMaker.

**Answer: C**

**Explanation:**

The correct answer is C: Create Amazon SageMaker VPC interface endpoints within the corporate VPC. This approach allows SageMaker notebook instances to communicate with the SageMaker service through AWS's internal network, completely avoiding the public internet.

VPC interface endpoints, powered by AWS PrivateLink, provide private connectivity to AWS services and supported VPC endpoint services from within your VPC, without exposing your traffic to the public internet. By creating SageMaker VPC interface endpoints (specifically for SageMaker API, SageMaker Runtime, and SageMaker Notebook Instances), you establish a secure, direct connection between your notebook instances and the SageMaker service, all within the boundaries of your VPC.

Option A, creating a NAT Gateway, enables outbound internet access for instances in a private subnet. This would violate the corporate data security policy prohibiting internet communication.

Option B, routing traffic through an on-premises network, would require exposing SageMaker traffic to the internet or establishing a complex VPN/Direct Connect setup which also requires internet access as an intermediate hop, complicating security and adding latency and overhead.

Option D, VPC Peering, connects two VPCs, but it doesn't inherently solve the problem of internet access. While VPC Peering allows instances in the VPCs to communicate, it doesn't automatically create a private pathway to the SageMaker service. It still depends on routing and could potentially require internet access.

Therefore, using VPC interface endpoints is the most secure and efficient method to enable SageMaker access without internet access, aligning with the company's strict data security policy.

Here are some authoritative links for further research:

**AWS PrivateLink:**https://aws.amazon.com/privatelink/
**Using Amazon SageMaker with VPC:**https://docs.aws.amazon.com/sagemaker/latest/dg/vpc.html
**Create VPC Endpoints:**https://docs.aws.amazon.com/vpc/latest/privatelink/create-endpoint.html

**Question: 48**

A Machine Learning Specialist is training a model to identify the make and model of vehicles in images. The Specialist wants to use transfer learning and an existing model trained on images of general objects. The Specialist collated a large custom dataset of pictures containing different vehicle makes and models. What should the Specialist do to initialize the model to re-train it with the custom data?

A.Initialize the model with random weights in all layers including the last fully connected layer.

B.Initialize the model with pre-trained weights in all layers and replace the last fully connected layer.

C.Initialize the model with random weights in all layers and replace the last fully connected layer.

D.Initialize the model with pre-trained weights in all layers including the last fully connected layer.

**Answer: B**

**Explanation:**

The correct approach to transfer learning in this scenario, where a pre-trained model on general object images is adapted for identifying vehicle makes and models, is option B: "Initialize the model with pre-trained weights in all layers and replace the last fully connected layer."

Here's why:

1. **Leveraging Pre-trained Knowledge:** Transfer learning exploits knowledge gained from training on a large dataset (in this case, general object images) to improve performance on a related, but different task (vehicle identification). The early layers of a convolutional neural network (CNN) learn general features like edges, textures, and shapes. These features are useful regardless of whether the image contains a cat, a dog, or a car. Therefore, initializing with pre-trained weights allows the model to start with a strong foundation, reducing the need to learn these basic features from scratch on the new dataset.

2. **The Role of the Last Fully Connected Layer:** The last fully connected layer of the pre-trained model is specifically designed to classify images into the categories it was originally trained on (e.g., cats, dogs, birds, etc.). These categories are irrelevant to the vehicle identification task. Therefore, this layer needs to be replaced with a new fully connected layer whose output size matches the number of vehicle makes and models in the custom dataset.

3. **Why not Random Initialization for all Layers?** Initializing all layers with random weights (options A and C) defeats the purpose of transfer learning. It effectively means training a new model from scratch, which is computationally expensive and requires a large amount of labeled data to achieve good performance. The whole point of transfer learning is to avoid this.

4. **Why not using pre-trained weights in all layers including the last one (option D)?** This will cause a classification based on the initial set of labels, which is completely unrelated to the target vehicle identification task. You can't directly use the pre-trained last layer for new classes as the features space of pre-trained classes is different from the one required for the target task.

**In summary:** By initializing with pre-trained weights in most layers and replacing only the last fully connected layer, the model can quickly adapt to the vehicle identification task by fine-tuning the pre-trained features and learning new, task-specific features in the final layer.

**Authoritative Links:**

**Transfer Learning:**https://papers.nips.cc/paper/2009/file/f93cd75372f03442ab92f679fbb2e6cd-Paper.pdf **How transferable are features in deep neural networks?**https://arxiv.org/abs/1411.1792

**Question: 49**

An office security agency conducted a successful pilot using 100 cameras installed at key locations within the main office. Images from the cameras were uploaded to Amazon S3 and tagged using Amazon Rekognition, and the results were stored in Amazon ES. The agency is now looking to expand the pilot into a full production system using thousands of video cameras in its office locations globally. The goal is to identify activities performed by non-employees in real time
Which solution should the agency consider?

A.Use a proxy server at each local office and for each camera, and stream the RTSP feed to a unique Amazon Kinesis Video Streams video stream. On each stream, use Amazon Rekognition Video and create a stream processor to detect faces from a collection of known employees, and alert when non-employees are detected.

B.Use a proxy server at each local office and for each camera, and stream the RTSP feed to a unique Amazon Kinesis Video Streams video stream. On each stream, use Amazon Rekognition Image to detect faces from a collection of known employees and alert when non-employees are detected.

C.Install AWS DeepLens cameras and use the DeepLens_Kinesis_Video module to stream video to Amazon Kinesis Video Streams for each camera. On each stream, use Amazon Rekognition Video and create a stream processor to detect faces from a collection on each stream, and alert when non-employees are detected.

D.Install AWS DeepLens cameras and use the DeepLens_Kinesis_Video module to stream video to Amazon Kinesis Video Streams for each camera. On each stream, run an AWS Lambda function to capture image fragments and then call Amazon Rekognition Image to detect faces from a collection of known employees, and alert when non-employees are detected.

**Answer: A**

**Explanation:**

Here's a detailed justification for why option A is the most suitable solution for the office security agency's requirements, along with supporting concepts and links:

The primary goal is real-time identification of activities performed by non-employees using thousands of video cameras globally. This demands a solution capable of handling high-volume, continuous video streams and performing real-time analysis.

**Kinesis Video Streams:** This service is specifically designed to ingest, store, and process video streams at scale. It efficiently handles the continuous stream of video data from numerous cameras.
https://aws.amazon.com/kinesis/video-streams/

**RTSP and Proxy Servers:** Real-Time Streaming Protocol (RTSP) is a common protocol for streaming video from IP cameras. Using a proxy server at each local office is beneficial for several reasons: it can reduce bandwidth consumption by caching frequently accessed content, improve security by hiding the internal network structure, and distribute the load of streaming video. Each camera streaming to a unique Kinesis Video Stream gives granular control and parallelism.

**Amazon Rekognition Video:** This service provides real-time facial analysis capabilities within video streams. It allows you to detect faces, compare them against a collection of known faces (employees), and identify non-employees. It provides real-time analysis. https://aws.amazon.com/rekognition/video/

**Stream Processor:** A Rekognition Video stream processor enables continuous analysis of the video stream for specific events (in this case, non-employee detection). It automates the process of facial recognition and alerting.

**Why other options are less suitable:**

**Option B (Amazon Rekognition Image):** Rekognition Image analyzes still images, not video streams in real time. Analyzing individual frames extracted from a video stream would be less efficient and miss crucial temporal information (like movement patterns).

**Option C and D (AWS DeepLens):** While DeepLens is useful for edge computing and machine learning at the

device level, it might not be the most cost-effective solution for thousands of cameras. It introduces complexity by adding device management. The question emphasizes expansion of an existing system that uploads to S3; DeepLens is a different approach that might not integrate easily.

**Option D (AWS Lambda):** Using a Lambda function to capture image fragments and then call Rekognition Image would introduce latency and complexity, making real-time analysis challenging to achieve. Lambda is also not suited for continuous stream processing in this manner; Rekognition Video is a dedicated service for this specific purpose.

Therefore, Option A provides the most efficient, scalable, and real-time solution for identifying non-employees in a global environment with thousands of video cameras by utilizing Kinesis Video Streams for ingestion and Rekognition Video for the facial recognition.

## Question: 50

A Marketing Manager at a pet insurance company plans to launch a targeted marketing campaign on social media to acquire new customers. Currently, the company has the following data in Amazon Aurora:

- Profiles for all past and existing customers
- Profiles for all past and existing insured pets
- Policy-level information
- Premiums received
- Claims paid

What steps should be taken to implement a machine learning model to identify potential new customers on social media?

A.Use regression on customer profile data to understand key characteristics of consumer segments. Find similar profiles on social media

B.Use clustering on customer profile data to understand key characteristics of consumer segments. Find similar profiles on social media

C.Use a recommendation engine on customer profile data to understand key characteristics of consumer segments. Find similar profiles on social media.

D.Use a decision tree classifier engine on customer profile data to understand key characteristics of consumer segments. Find similar profiles on social media.

**Answer: B**

**Explanation:**

The most appropriate approach to identify potential new customers on social media given the provided data and business objective is option B: "Use clustering on customer profile data to understand key characteristics of consumer segments. Find similar profiles on social media."

Here's why:

**Understanding Consumer Segments:** The core goal is to identify groups of customers with similar characteristics. Clustering algorithms are specifically designed for this purpose. They can automatically group customers based on shared attributes derived from their profiles (demographics, pet information, policy details, etc.). This allows the marketing manager to define distinct consumer segments. K-Means clustering is a popular algorithm for this.

**Targeted Marketing:** By understanding the common traits of profitable customer segments, the marketing manager can then search for similar profiles on social media platforms. This ensures the marketing campaign is targeted at individuals who are most likely to be interested in pet insurance.

**Alternatives:**

**Regression (A):** Regression is primarily used for predicting continuous values (e.g., customer lifetime value). While it could be used to predict customer behavior, it doesn't directly help in identifying distinct customer

segments.

**Recommendation Engine (C):** Recommendation engines are designed to suggest items to users based on their past behavior. This is not the best fit for finding new customers on social media.

**Decision Tree Classifier (D):** Decision trees are used for classification problems, where you want to predict a categorical outcome (e.g., whether someone will buy a policy or not). While potentially useful after customer segments are identified, it's not the initial step in identifying segments.

Clustering is thus superior as it provides a natural way to group existing customers into segments, enabling targeted marketing efforts on social media to find new customers with similar profiles. This approach allows the pet insurance company to focus its resources on individuals most likely to convert.

Further Research:

**Clustering:**https://scikit-learn.org/stable/modules/clustering.html
**K-Means Clustering:**https://aws.amazon.com/blogs/machine-learning/k-means-clustering-with-amazon-sagemaker/
**Targeted Marketing:**https://www.oracle.com/uk/cx/marketing/what-is-targeted-marketing/

## Question: 51

A manufacturing company has a large set of labeled historical sales data. The manufacturer would like to predict how many units of a particular part should be produced each quarter.
Which machine learning approach should be used to solve this problem?

A.Logistic regression
B.Random Cut Forest (RCF)
C.Principal component analysis (PCA)
D.Linear regression

**Answer: D**

**Explanation:**

Here's a detailed justification for choosing linear regression as the appropriate machine learning approach:

The problem describes a scenario where a manufacturing company wants to predict the quantity of units to produce. Since we are predicting a continuous numerical value (number of units), this is fundamentally a regression problem.

Let's evaluate each option:

**A. Logistic Regression:** Logistic regression is used for classification problems, where the goal is to predict a categorical outcome (e.g., spam or not spam, success or failure). It predicts the probability of a binary outcome. This is not relevant to predicting a continuous number of units.

**B. Random Cut Forest (RCF):** RCF is an unsupervised anomaly detection algorithm. It's used to identify data points that are significantly different from the rest of the dataset. This isn't about predicting future sales quantity. Its use case is specifically detecting anamolies, such as fraud detection.

**C. Principal Component Analysis (PCA):** PCA is a dimensionality reduction technique. It's used to reduce the number of variables in a dataset while preserving its essential information. While PCA can be used as a preprocessing step, it doesn't directly predict sales quantity. It is more related to data optimization.

**D. Linear Regression:** Linear regression models the relationship between a dependent variable (sales quantity) and one or more independent variables (historical sales data, time period). It aims to find the best-fitting linear equation to predict the dependent variable based on the independent variables. Given the

historical data, it can forecast future sales quantity. It fits the definition of a regression problem which is predicting future sales quantity.

Therefore, **Linear Regression** is the most appropriate machine learning approach. It directly addresses the problem of predicting a continuous numerical value (the number of units to produce) based on labeled historical data.**Authoritative Links for further research:**

AWS Machine Learning Documentation: https://aws.amazon.com/machine-learning/
Linear Regression on AWS: https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html Random Cut Forest on AWS: https://docs.aws.amazon.com/sagemaker/latest/dg/randomcutforest.html PCA: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
Logistic Regression: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

**Question: 52**

A financial services company is building a robust serverless data lake on Amazon S3. The data lake should be flexible and meet the following requirements:

- Support querying old and new data on Amazon S3 through Amazon Athena and Amazon Redshift Spectrum.
- Support event-driven ETL pipelines
- Provide a quick and easy way to understand metadata

Which approach meets these requirements?

A.Use an AWS Glue crawler to crawl S3 data, an AWS Lambda function to trigger an AWS Glue ETL job, and an AWS Glue Data catalog to search and discover metadata.

B.Use an AWS Glue crawler to crawl S3 data, an AWS Lambda function to trigger an AWS Batch job, and an external Apache Hive metastore to search and discover metadata.

C.Use an AWS Glue crawler to crawl S3 data, an Amazon CloudWatch alarm to trigger an AWS Batch job, and an AWS Glue Data Catalog to search and discover metadata.

D.Use an AWS Glue crawler to crawl S3 data, an Amazon CloudWatch alarm to trigger an AWS Glue ETL job, and an external Apache Hive metastore to search and discover metadata.

**Answer: A**

**Explanation:**

The correct answer is A. Here's a detailed justification:

**AWS Glue Crawler:** An AWS Glue crawler is crucial for automatically discovering the schema of the data residing in the S3 data lake. It scans the data and infers the structure, creating metadata tables in the AWS Glue Data Catalog. This addresses the requirement of supporting querying data through Athena and Redshift Spectrum, which rely on metadata.

**AWS Lambda Function:** AWS Lambda is perfectly suited for event-driven ETL pipelines. It can be triggered by S3 events (e.g., new data arriving) and initiate an AWS Glue ETL job. This satisfies the requirement of an event-driven pipeline.

**AWS Glue ETL Job:** AWS Glue ETL jobs provide the core functionality for transforming and loading data. This ensures that the data can be processed and prepared for analysis and querying.

**AWS Glue Data Catalog:** The AWS Glue Data Catalog is a managed metadata repository. It acts as a central place to store and discover metadata about the data in the S3 data lake. This directly addresses the requirement for a quick and easy way to understand metadata. Athena and Redshift Spectrum both integrate with the Glue Data Catalog.

**Why other options are not as suitable:**

**B:** While AWS Glue crawler and Lambda function can be used to crawl S3 data and trigger Batch job. Using an external Apache Hive metastore adds unnecessary complexity and management overhead compared to using the AWS Glue Data Catalog, which is fully managed and integrated with other AWS services.

**C & D:** Using CloudWatch alarms to trigger ETL jobs is not event-driven in the truest sense. It relies on a schedule or metric threshold rather than directly reacting to data arriving in S3. AWS Lambda offers a more flexible and responsive solution.

**Authoritative Links:**

AWS Glue: https://aws.amazon.com/glue/
AWS Lambda: https://aws.amazon.com/lambda/
Amazon Athena: https://aws.amazon.com/athena/
Amazon Redshift Spectrum: https://aws.amazon.com/redshift/spectrum/

## Question: 53

A company's Machine Learning Specialist needs to improve the training speed of a time-series forecasting model using TensorFlow. The training is currently implemented on a single-GPU machine and takes approximately 23 hours to complete. The training needs to be run daily.
The model accuracy is acceptable, but the company anticipates a continuous increase in the size of the training data and a need to update the model on an hourly, rather than a daily, basis. The company also wants to minimize coding effort and infrastructure changes.
What should the Machine Learning Specialist do to the training solution to allow it to scale for future demand?

A.Do not change the TensorFlow code. Change the machine to one with a more powerful GPU to speed up the training.

B.Change the TensorFlow code to implement a Horovod distributed framework supported by Amazon SageMaker. Parallelize the training to as many machines as needed to achieve the business goals.

C.Switch to using a built-in AWS SageMaker DeepAR model. Parallelize the training to as many machines as needed to achieve the business goals.

D.Move the training to Amazon EMR and distribute the workload to as many machines as needed to achieve the business goals.

**Answer: B**

**Explanation:**

The correct answer is B. Here's a detailed justification:

The problem requires reducing the training time of a TensorFlow time-series model, anticipating increasing data size, and a shift to hourly training updates while minimizing coding effort and infrastructure changes.

Option B, utilizing Horovod within Amazon SageMaker, directly addresses these needs. Horovod is a distributed deep learning training framework that simplifies the parallelization of TensorFlow models. SageMaker provides managed infrastructure and tooling that significantly reduces the effort required to set up and manage distributed training environments. https://horovod.readthedocs.io/en/stable/ &
https://aws.amazon.com/sagemaker/

Horovod's ease of integration with TensorFlow minimizes code changes, satisfying the "minimize coding effort" requirement. Scaling the training is achieved by parallelizing the workload across multiple machines provisioned by SageMaker, addressing the need for future scalability and faster training.

Option A, simply upgrading the GPU, might provide some initial improvement, but it's a short-term solution. It doesn't scale well as data increases and eventually hits hardware limitations. It also does not address the need to update the model hourly.

Option C, switching to SageMaker DeepAR, could be considered. However, it necessitates a complete model rewrite which increases the coding effort substantially, violating the "minimize coding effort" constraint. DeepAR is a viable solution when starting a time-series forecasting project from scratch, but not optimal when refactoring existing models.

Option D, moving the training to Amazon EMR, introduces a larger infrastructure change. EMR typically involves more complex cluster management and requires significant configuration compared to SageMaker's managed distributed training capabilities. This increases the operational overhead and defeats the need to minimize infrastructural changes. Additionally, while EMR supports distributed training, it doesn't offer the same seamless integration with TensorFlow and Horovod as SageMaker.

**Question: 54**

Which of the following metrics should a Machine Learning Specialist generally use to compare/evaluate machine learning classification models against each other?

A.Recall
B.Misclassification rate
C.Mean absolute percentage error (MAPE)
D.Area Under the ROC Curve (AUC)

**Answer: D**

**Explanation:**

The correct answer is **D. Area Under the ROC Curve (AUC)**. Here's why:

AUC is a comprehensive metric for evaluating classification models because it considers the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) across various classification thresholds. A ROC (Receiver Operating Characteristic) curve plots these rates, and AUC quantifies the area under this curve. A higher AUC (closer to 1) indicates a better-performing model, suggesting it effectively distinguishes between positive and negative classes. This makes AUC a valuable metric for model comparison, as it provides an aggregated measure of performance independent of a single chosen threshold.

A. **Recall** (also known as sensitivity or true positive rate) focuses only on the ability of the model to identify positive instances. While important, it doesn't consider false positives. A model can achieve perfect recall by simply predicting everything as positive, which is often undesirable.

B. **Misclassification rate** (or error rate) represents the proportion of incorrect predictions. While simple to understand, it can be misleading, especially with imbalanced datasets where one class significantly outnumbers the other. A model might have a low misclassification rate simply by predicting the majority class most of the time.

C. **Mean Absolute Percentage Error (MAPE)** is a common metric for regression problems, not classification. It measures the average percentage difference between predicted and actual continuous values, which is not relevant when evaluating the performance of classification models that predict categorical outcomes.

Therefore, AUC offers a more balanced and informative measure of classification model performance than recall or misclassification rate, and it is appropriate for classification tasks unlike MAPE. It considers both sensitivity and specificity, making it suitable for comparing models across different classification tasks, particularly where the class distribution might be uneven. Selecting a good threshold can be chosen based on a cost/benefit analysis.

Further reading:

**Understanding AUC:**https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 **ROC and AUC:**https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

## Question: 55

A company is running a machine learning prediction service that generates 100 TB of predictions every day. A Machine Learning Specialist must generate a visualization of the daily precision-recall curve from the predictions, and forward a read-only version to the Business team.
Which solution requires the LEAST coding effort?

A.Run a daily Amazon EMR workflow to generate precision-recall data, and save the results in Amazon S3. Give the Business team read-only access to S3.

B.Generate daily precision-recall data in Amazon QuickSight, and publish the results in a dashboard shared with the Business team.

C.Run a daily Amazon EMR workflow to generate precision-recall data, and save the results in Amazon S3. Visualize the arrays in Amazon QuickSight, and publish them in a dashboard shared with the Business team.

D.Generate daily precision-recall data in Amazon ES, and publish the results in a dashboard shared with the Business team.

**Answer: C**

**Explanation:**

The question asks for the solution that requires the least coding effort to visualize daily precision-recall curves from 100 TB of daily predictions and share a read-only version with the business team.

Option C is the best answer because it leverages the strengths of both Amazon EMR and QuickSight with minimal coding. EMR is suitable for processing large datasets like 100TB to generate the precision-recall data. Then, QuickSight can visualize this data from S3 and publish an interactive dashboard for the business team with read-only access. QuickSight's ability to directly visualize data stored in S3 simplifies the process. This avoids complex data ingestion pipelines directly into visualization tools.

Option A requires only EMR and S3. It generates the precision-recall data and saves it to S3, granting the Business team read-only access. However, it doesn't natively provide a visualization, leaving the Business team to handle visualization themselves (which demands more coding or using other tools by the business team).

Option B suggests generating precision-recall data in QuickSight directly. While QuickSight excels at visualization, generating insights directly within QuickSight for 100TB of daily data would be computationally expensive and challenging. QuickSight is not designed to handle that data volume for real-time calculations without pre-aggregation. This would likely involve writing custom code or utilizing more complex QuickSight functionalities.

Option D suggests using Amazon ES. ES is a good choice for indexing and searching logs and operational data. Generating and visualizing 100TB of data in ES is not its core strength. ES would need to ingest the data, potentially index it, and then create visualizations. This would likely require more coding and is not optimized for numerical computation required for precision-recall curve generation.

Therefore, Option C provides the best balance between data processing and visualization with the least coding effort by utilizing EMR for data processing and QuickSight for visualization.

**Supporting Links:**

**Amazon EMR:**https://aws.amazon.com/emr/
**Amazon QuickSight:**https://aws.amazon.com/quicksight/
**Amazon S3:**https://aws.amazon.com/s3/

## Question: 56

A Machine Learning Specialist is preparing data for training on Amazon SageMaker. The Specialist is using one of the SageMaker built-in algorithms for the training. The dataset is stored in .CSV format and is transformed into a numpy.array, which appears to be negatively affecting the speed of the training.
What should the Specialist do to optimize the data for training on SageMaker?

A.Use the SageMaker batch transform feature to transform the training data into a DataFrame.

B.Use AWS Glue to compress the data into the Apache Parquet format.

C.Transform the dataset into the RecordIO protobuf format.

D.Use the SageMaker hyperparameter optimization feature to automatically optimize the data.

**Answer: C**

**Explanation:**

The correct answer is **C. Transform the dataset into the RecordIO protobuf format.** Here's why:

The problem states that converting the CSV data to a NumPy array is slowing down the training process. SageMaker built-in algorithms are optimized to work with specific data formats, often more efficient than generic NumPy arrays for large datasets.

**RecordIO/protobuf** is a highly efficient binary format that SageMaker algorithms are designed to read quickly. This format reduces parsing overhead compared to CSV and optimizes data transfer to the training instances. Using RecordIO also allows efficient sharding of data for distributed training.

Let's analyze why the other options are less optimal:

**A. Use the SageMaker batch transform feature to transform the training data into a DataFrame.** Batch transform is primarily for inference (generating predictions on a large dataset), not optimizing the training data format. While DataFrames can be efficient, they are not the most efficient format for SageMaker built-in algorithms compared to RecordIO.

**B. Use AWS Glue to compress the data into the Apache Parquet format.** Parquet is a columnar storage format that's excellent for analytics and querying in services like Athena. While compression is beneficial, Parquet is not the most natively supported and optimized format for SageMaker training specifically.

Furthermore, Glue is generally used for data transformation at rest, not specifically to accelerate the data pipeline to the training job.

**D. Use the SageMaker hyperparameter optimization feature to automatically optimize the data.**

Hyperparameter optimization optimizes the model's parameters, not the data format. While hyperparameter optimization is a crucial part of machine learning, it doesn't address the data format inefficiency highlighted in the scenario.

In summary, RecordIO/protobuf format is the most suitable option because it is designed to improve the speed of the training process when using built-in SageMaker algorithms due to its efficient parsing and data transfer characteristics.

Here are authoritative links for further research:

SageMaker Data Formats: Outlines supported data formats for SageMaker.
SageMaker Built-in Algorithms: Provides details about individual algorithms and their expected data format.
Amazon SageMaker Examples: Contains many examples utilizing the RecordIO format for training.

## Question: 57

A Machine Learning Specialist is required to build a supervised image-recognition model to identify a cat. The ML Specialist performs some tests and records the following results for a neural network-based image classifier: Total number of images available = 1,000
Test set images = 100 (constant test set)
The ML Specialist notices that, in over 75% of the misclassified images, the cats were held upside down by their owners.
Which techniques can be used by the ML Specialist to improve this specific test error?

A.Increase the training data by adding variation in rotation for training images.

B.Increase the number of epochs for model training

C.Increase the number of layers for the neural network.

D.Increase the dropout rate for the second-to-last layer.

**Answer: A**

**Explanation:**

The correct answer is A: Increase the training data by adding variation in rotation for training images. Here's why:

The problem lies in the model's inability to generalize to images of cats in an unusual orientation (upside down). This indicates a lack of representation of rotated images in the training data, leading to poor performance on such examples in the test set.

Option A directly addresses this issue. By augmenting the training dataset with rotated images of cats, the model is exposed to the variations it's currently failing to recognize. This increased diversity in the training data enables the model to learn features that are invariant to rotation, improving its ability to correctly classify cats regardless of their orientation. Data augmentation is a common technique in machine learning to improve model generalization and robustness.

Option B, increasing the number of epochs, might help to a limited extent, but if the data lacks the necessary information (rotated images), the model will primarily overfit the existing data, leading to minimal improvements in the specific error case.

Option C, increasing the number of layers, increases the model's complexity and risk of overfitting if the training data isn't representative. It doesn't specifically address the lack of rotated images.

Option D, increasing the dropout rate, is a regularization technique that aims to prevent overfitting. While it might improve generalization slightly, it doesn't directly tackle the issue of the model's unfamiliarity with rotated images.

In conclusion, the most effective solution is to augment the training dataset with rotated cat images. This directly addresses the identified problem and enhances the model's ability to handle variations in cat orientation, thereby improving performance on the test set.

Further research:

Data Augmentation: https://www.tensorflow.org/tutorials/images/data_augmentation Overfitting and Underfitting: https://www.tensorflow.org/tutorials/keras/overfit_and_underfit

## Question: 58

A Machine Learning Specialist needs to be able to ingest streaming data and store it in Apache Parquet files for exploration and analysis.
Which of the following services would both ingest and store this data in the correct format?

A.AWS DMS

B.Amazon Kinesis Data Streams

C.Amazon Kinesis Data Firehose

D.Amazon Kinesis Data Analytics

**Answer: C**

**Explanation:**

The correct answer is **C. Amazon Kinesis Data Firehose**. Here's a detailed justification:

Kinesis Data Firehose is designed specifically for loading streaming data into data lakes and data warehouses. It can automatically convert streaming data into formats like Parquet before storing it in destinations such as Amazon S3. This fulfills the requirement of storing the ingested data in Parquet format. It can also handle data transformation, compression, and encryption while streaming.

Option A, AWS DMS (Database Migration Service), is used for migrating databases, not ingesting streaming data. It is geared toward transferring data from one database to another and does not directly handle streaming ingestion in Parquet.

Option B, Amazon Kinesis Data Streams, is for collecting and processing large streams of data records in real-time. While it can ingest data, it doesn't natively support converting and storing the data directly in Parquet format. You'd need additional processing, possibly using Kinesis Data Analytics, to transform and store the data in Parquet.

Option D, Amazon Kinesis Data Analytics, is for processing and analyzing streaming data in real time using SQL or Apache Flink. It doesn't directly ingest data or store it; instead, it performs computations on the data ingested by Kinesis Data Streams or Data Firehose. While it can be part of a solution, it's not the single service that both ingests and stores data in Parquet format.

Therefore, Kinesis Data Firehose stands out as the service that meets both ingestion and storage criteria, directly outputting Parquet format to destinations such as S3.

Authoritative Links:

**Amazon Kinesis Data Firehose:**https://aws.amazon.com/kinesis/data-firehose/
**Parquet Format:**https://parquet.apache.org/

## Question: 59

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:**

Here's a detailed justification for why option B is the correct approach, along with explanations of why the other options are less suitable:

The goal is to handle features with varying statistical dispersion and ensure good prediction performance in production. This requires splitting the data into training, validation, and test sets before rescaling to prevent data leakage. Data leakage occurs when information from the validation or test sets inadvertently influences the training process, leading to overly optimistic performance estimates during development that do not generalize well to unseen production data.

**Option B: Correct Approach**

1. **Split the data:** Divide the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters and evaluate model performance during training, and the test set is used for a final, unbiased evaluation of the model's generalization ability.

2. **Rescale the training set:** Apply a scaling technique (e.g., standardization or normalization) to the training set. Standardization transforms the data to have zero mean and unit variance, while normalization scales the data to a range between 0 and 1. This addresses the issue of features having widely varying scales, which can bias some models that are sensitive to feature magnitudes (e.g., gradient descent-based algorithms).

3. **Apply the same scaling to validation and test sets:** Crucially, use the scaling parameters (mean and standard deviation for standardization, min and max for normalization) derived from the training set to rescale the validation and test sets. This ensures that the validation and test sets are transformed in the same way as the training set, preventing data leakage. The model should only ever "see" the statistical properties of the training data.

**Why other options are incorrect:**

**Option A:** Random sampling after rescaling is irrelevant to this problem. Random sampling before splitting will not prevent data leakage. Random sampling without fixing the random seed will cause training/validation/test splits to be different on each run, leading to different results.

**Option C:** Rescaling the entire dataset before splitting introduces data leakage. The scaling parameters (mean, standard deviation, min, max) are influenced by all data points, including those in the validation and test sets. This gives the model an unfair advantage and leads to overly optimistic performance estimates during development, which do not generalize well to unseen production data.

**Option D:** Rescaling training, validation, and test sets independently introduces data leakage. The validation and test sets will be transformed using their own scaling parameters, which are different from those used for the training set. This leads to a mismatch between the data the model is trained on and the data it is evaluated on, making the validation and test performance unreliable.

**In summary:** Splitting the data before scaling and then applying the same scaling to the validation and test set using parameters only derived from the training set is the best practice to prevent data leakage, address the issue of varying feature scales, and ensure reliable performance in production.

**Authoritative Links:**

**Data Leakage:**https://www.kaggle.com/dansbecker/data-leakage
**Feature Scaling:**https://scikit-learn.org/stable/modules/preprocessing.html
**Train/Validation/Test Split:**https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

**Question: 60**

A Machine Learning Specialist is assigned a TensorFlow project using Amazon SageMaker for training, and needs to continue working for an extended period with no Wi-Fi access.
Which approach should the Specialist use to continue working?

A.Install Python 3 and boto3 on their laptop and continue the code development using that environment.

B.Download the TensorFlow Docker container used in Amazon SageMaker from GitHub to their local environment, and use the Amazon SageMaker Python SDK to test the code.

C.Download TensorFlow from tensorflow.org to emulate the TensorFlow kernel in the SageMaker environment.

D.Download the SageMaker notebook to their local environment, then install Jupyter Notebooks on their laptop and continue the development in a local notebook.

**Answer: B**

**Explanation:**

The correct answer is **B. Download the TensorFlow Docker container used in Amazon SageMaker from GitHub to their local environment, and use the Amazon SageMaker Python SDK to test the code.**

Here's why:

**SageMaker Environment Replication:** The goal is to replicate the SageMaker environment as closely as possible to avoid discrepancies when the code is eventually deployed back to SageMaker. Downloading the TensorFlow Docker container used by SageMaker ensures you're using the same libraries, versions, and dependencies.

**Docker for Consistency:** Docker containers provide a consistent and isolated environment, encapsulating all the necessary software and configurations. This avoids issues caused by differing operating systems or package versions on the local machine.

**SageMaker Python SDK:** The SageMaker Python SDK is designed to interact with SageMaker services. Even locally, it allows you to structure your code in a way that's compatible with SageMaker's training and deployment pipelines. You can test your code locally using the SDK without actually interacting with the SageMaker service until Wi-Fi access is restored.

**Why other options are less ideal:**

**A:** Installing Python and boto3 is a good starting point but doesn't guarantee the same TensorFlow version and dependencies as the SageMaker environment. Boto3 helps you interact with AWS services, but doesn't replicate the SageMaker environment.

**C:** Downloading TensorFlow from tensorflow.org provides a TensorFlow installation, but might not match the specific version and configurations used in the SageMaker environment. It also doesn't address the other dependencies and environment settings.

**D:** Downloading the SageMaker notebook and installing Jupyter Notebooks is a good idea for editing the notebook but will require you to install the correct TensorFlow environment and all other required packages to run without access to the SageMaker Kernel or environment. It also doesn't capture the complete environment as well as the Docker container.

**Authoritative Links for further research:**

**Amazon SageMaker:**https://aws.amazon.com/sagemaker/
**SageMaker Python SDK:**https://sagemaker.readthedocs.io/en/stable/
**Docker:**https://www.docker.com/
**AWS Deep Learning Containers:**https://aws.amazon.com/reInvent/news/aws-deep-learning-containers-now-available/

In summary, option B is the best approach because it allows the specialist to create a local environment that closely mimics the Amazon SageMaker environment, ensuring consistency and reducing potential issues when deploying the code back to SageMaker.

## Question: 61

A Machine Learning Specialist is working with a large cybersecurity company that manages security events in real time for companies around the world. The cybersecurity company wants to design a solution that will allow it to use machine learning to score malicious events as anomalies on the data as it is being ingested. The company also wants be able to save the results in its data lake for later processing and analysis.
What is the MOST efficient way to accomplish these tasks?

A.Ingest the data using Amazon Kinesis Data Firehose, and use Amazon Kinesis Data Analytics Random Cut Forest (RCF) for anomaly detection. Then use Kinesis Data Firehose to stream the results to Amazon S3.

B.Ingest the data into Apache Spark Streaming using Amazon EMR, and use Spark MLlib with k-means to perform anomaly detection. Then store the results in an Apache Hadoop Distributed File System (HDFS) using Amazon EMR with a replication factor of three as the data lake.

C.Ingest the data and store it in Amazon S3. Use AWS Batch along with the AWS Deep Learning AMIs to train a k-means model using TensorFlow on the data in Amazon S3.

D.Ingest the data and store it in Amazon S3. Have an AWS Glue job that is triggered on demand transform the new data. Then use the built-in Random Cut Forest (RCF) model within Amazon SageMaker to detect anomalies in the data.

**Answer: A**

**Explanation:**

The most efficient solution is option A, leveraging Amazon Kinesis Data Firehose and Kinesis Data Analytics with Random Cut Forest (RCF).

**Real-time Anomaly Detection:** The company requires real-time anomaly detection as data is ingested. Kinesis Data Analytics enables processing streaming data in real time. RCF, a built-in algorithm in Kinesis Data Analytics, is specifically designed for anomaly detection in streaming data.
https://docs.aws.amazon.com/kinesisanalytics/latest/java/how-it-works-anomaly.html

**Data Ingestion and Delivery:** Kinesis Data Firehose efficiently ingests streaming data and delivers it to various destinations, including S3 for data lake storage. https://aws.amazon.com/kinesis/data-firehose/

**Efficiency and Scalability:** This solution is highly efficient because it uses managed services optimized for streaming data processing. Kinesis Data Analytics automatically scales to handle the data stream's volume.

**Cost-Effectiveness:** Kinesis Data Analytics and Firehose are pay-as-you-go services, optimizing cost based on actual usage.

Options B, C, and D are less efficient:

**Option B:** Using Apache Spark Streaming on EMR involves more operational overhead for cluster management. While Spark MLlib can perform anomaly detection, it's not as readily integrated for real-time anomaly detection on streaming data as Kinesis Data Analytics RCF. Storing in HDFS within EMR is less efficient and scalable for a data lake compared to S3.

**Option C:** Training a k-means model with AWS Batch and Deep Learning AMIs is suitable for batch processing and model retraining, not real-time anomaly detection. It's less efficient for the company's need to score malicious events as they're ingested.

**Option D:** Using AWS Glue for data transformation and SageMaker for anomaly detection introduces latency.

Triggering a Glue job on demand isn't as real-time as Kinesis Data Analytics. While SageMaker's RCF is effective, it's better suited for batch anomaly detection or online learning scenarios than for direct integration with a streaming ingestion pipeline.

## Question: 62

A Data Scientist wants to gain real-time insights into a data stream of GZIP files. Which solution would allow the use of SQL to query the stream with the LEAST latency?

A.Amazon Kinesis Data Analytics with an AWS Lambda function to transform the data.
B.AWS Glue with a custom ETL script to transform the data.
C.An Amazon Kinesis Client Library to transform the data and save it to an Amazon ES cluster.
D.Amazon Kinesis Data Firehose to transform the data and put it into an Amazon S3 bucket.

**Answer: A**

**Explanation:**

The correct answer is A, Amazon Kinesis Data Analytics with an AWS Lambda function to transform the data. Here's why:

Kinesis Data Analytics is specifically designed for real-time processing and analysis of streaming data using SQL. This inherently offers low latency querying because it operates directly on the stream without the need for batch processing or data movement to other storage services for querying.

Option A uses a Lambda function to transform the GZIP files before they are ingested into Kinesis Data Analytics. Lambda functions are serverless and execute code quickly on demand, minimizing any added latency from the transformation step. This architecture facilitates the immediate availability of transformed data for SQL queries within Kinesis Data Analytics.

Option B, AWS Glue, is primarily a batch-oriented ETL (Extract, Transform, Load) service. While Glue can process data streams, it is better suited for scheduled batch jobs and incurs higher latency compared to Kinesis Data Analytics' continuous processing.

Option C, Kinesis Client Library (KCL) with Elasticsearch (ES), involves more manual coding and infrastructure management. While KCL can read from the Kinesis stream and perform transformations, the subsequent indexing and querying within Elasticsearch introduce additional latency compared to directly querying the stream with SQL using Kinesis Data Analytics.

Option D, Kinesis Data Firehose to S3, focuses on data delivery to S3 for storage and subsequent analysis. It is not designed for real-time SQL querying of the stream itself. Querying the data in S3 would require services like Athena or Redshift Spectrum, which involve loading data and therefore incur significantly higher latency.

Therefore, only Option A provides a real-time SQL query capability on a data stream with the lowest latency. It leverages the continuous processing nature of Kinesis Data Analytics along with the fast transformation speed of Lambda functions.

**Supporting Links:**

**Amazon Kinesis Data Analytics:**https://aws.amazon.com/kinesis/data-analytics/
**AWS Lambda:**https://aws.amazon.com/lambda/

## Question: 63

A retail company intends to use machine learning to categorize new products. A labeled dataset of current products was provided to the Data Science team. The dataset includes 1,200 products. The labeled dataset has 15 features for each product such as title dimensions, weight, and price. Each product is labeled as belonging to one of six categories such as books, games, electronics, and movies.
Which model should be used for categorizing new products using the provided dataset for training?

A.AnXGBoost model where the objective parameter is set to multi:softmax
B.A deep convolutional neural network (CNN) with a softmax activation function for the last layer
C.A regression forest where the number of trees is set equal to the number of product categories
D.A DeepAR forecasting model based on a recurrent neural network (RNN)

**Answer: A**

**Explanation:**

Here's a detailed justification for why option A is the most appropriate choice, along with supporting concepts and links for further research:

The core task is a multi-class classification problem: assigning each product to one of six distinct categories. XGBoost with the multi:softmax objective function is specifically designed for such scenarios. XGBoost is a powerful and efficient gradient boosting algorithm known for its accuracy, speed, and ability to handle complex datasets with numerous features. The multi:softmax objective directly optimizes for multi-class classification by predicting the probability of each class and selecting the class with the highest probability.
Given the dataset size (1,200 products) and the number of features (15), XGBoost can likely achieve good performance without excessive computational resources.

Option B, a CNN, is generally better suited for image or sequence data where spatial or temporal relationships between features are important. While CNNs can be applied to tabular data, it typically requires significant feature engineering to represent the data in a suitable format. Furthermore, CNNs often require larger datasets to train effectively, which could be a limitation with only 1,200 samples.

Option C, a Regression Forest, is primarily designed for regression tasks where the goal is to predict a continuous value, not discrete categories. While Regression Forests can be adapted for classification, it's typically done through ensemble methods or by treating each category as a separate regression problem, which might not be as efficient or accurate as a dedicated multi-class classifier like XGBoost with multi:softmax. Setting the number of trees equal to the number of categories is arbitrary and lacks theoretical justification.

Option D, DeepAR, is a specialized forecasting model based on recurrent neural networks (RNNs), primarily used for time series forecasting problems. It is not applicable to the product categorization problem, which involves classifying independent instances based on their features.

Therefore, XGBoost with the multi:softmax objective is the most appropriate model because it's a well-established, efficient, and accurate algorithm specifically designed for multi-class classification tasks, making it suitable for the given dataset and problem.

**Authoritative Links for Further Research:**

**XGBoost Documentation:**https://xgboost.readthedocs.io/en/stable/ - Provides comprehensive information on XGBoost, including the multi:softmax objective function.
**Multi-Class Classification:**https://developers.google.com/machine-learning/glossary/multi-class-classification - A glossary entry explaining multi-class classification from Google's Machine Learning Crash Course.
**scikit-learn's documentation on ensemble methods:**https://scikit-learn.org/stable/modules/ensemble.html - Includes details on Random Forests (which are related to Regression Forests) and their use in classification.

## Question: 64

A Data Scientist is working on an application that performs sentiment analysis. The validation accuracy is poor, and the Data Scientist thinks that the cause may be a rich vocabulary and a low average frequency of words in the dataset.
Which tool should be used to improve the validation accuracy?

A.Amazon Comprehend syntax analysis and entity detection

B.Amazon SageMaker BlazingText cbow mode

C.Natural Language Toolkit (NLTK) stemming and stop word removal

D.Scikit-leam term frequency-inverse document frequency (TF-IDF) vectorizer

**Answer: D**

**Explanation:**

The problem describes a scenario where a sentiment analysis model suffers from poor validation accuracy due to a rich vocabulary and low average word frequency. This indicates that rare words might be unduly influencing the model, while common words might not be contributing enough to the sentiment classification.

Option A (Amazon Comprehend syntax analysis and entity detection) focuses on understanding sentence structure and identifying key entities. While useful for more complex NLP tasks, it doesn't directly address the issue of vocabulary size and word frequency in improving sentiment analysis accuracy.

Option B (Amazon SageMaker BlazingText cbow mode) utilizes the Continuous Bag-of-Words (CBOW) algorithm for word embeddings. While helpful for capturing semantic relationships between words, it doesn't explicitly tackle the problem of weighting words based on their frequency within the document set, which is critical when rare words are negatively impacting model performance.

Option C (Natural Language Toolkit (NLTK) stemming and stop word removal) is a good preprocessing step. Stemming reduces words to their root form, and removing stop words (like "the", "a", "is") eliminates common, non-informative words. While helpful, it doesn't address the nuances of differing importance among the remaining words after these steps. It can potentially help reduce noise and vocabulary size, but it's not as effective as TF-IDF in assigning weights based on term importance in the context of the entire dataset.

Option D (Scikit-learn TF-IDF vectorizer) is the most appropriate solution. TF-IDF (Term Frequency-Inverse Document Frequency) assigns weights to words based on how frequently they appear in a specific document (TF) and how rarely they appear across the entire collection of documents (IDF). Rare words receive a higher IDF score, thus increasing their weight, while common words receive a lower IDF score. This helps the model focus on the more discriminative terms and reduces the impact of frequently occurring but non-informative words that are not removed by stop word lists. This directly addresses the problem of low-frequency words unduly influencing the model and boosts the contribution of words that are more specific and useful for sentiment classification. Therefore, TF-IDF will improve the validation accuracy by highlighting important words.

For further research, see:

**Scikit-learn TF-IDF:**https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
**Understanding TF-IDF:**https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/

## Question: 65

Machine Learning Specialist is building a model to predict future employment rates based on a wide range of

economic factors. While exploring the data, the
Specialist notices that the magnitude of the input features vary greatly. The Specialist does not want variables with a larger magnitude to dominate the model.
What should the Specialist do to prepare the data for model training?

A.Apply quantile binning to group the data into categorical bins to keep any relationships in the data by replacing the magnitude with distribution.

B.Apply the Cartesian product transformation to create new combinations of fields that are independent of the magnitude.

C.Apply normalization to ensure each field will have a mean of 0 and a variance of 1 to remove any significant magnitude.

D.Apply the orthogonal sparse bigram (OSB) transformation to apply a fixed-size sliding window to generate new features of a similar magnitude.

**Answer: C**

**Explanation:**

The correct answer is C: Apply normalization to ensure each field will have a mean of 0 and a variance of 1 to remove any significant magnitude. This addresses the problem of features with widely varying magnitudes dominating the model. This technique, often called standardization or Z-score normalization, transforms the data so that each feature has a mean of 0 and a standard deviation of 1. By doing this, each feature contributes more equitably to the model, preventing features with larger scales from unduly influencing the learning process.

Option A, quantile binning, converts numerical features into categorical features. While binning can be useful in some scenarios, it doesn't directly address the issue of magnitude differences and might lose information present in the original numerical values.

Option B, the Cartesian product transformation, creates new features by combining existing features. This would drastically increase the dimensionality of the data and does not address the magnitude issue. Furthermore, it is more relevant for feature engineering related to interactions between different features rather than scaling.

Option D, the orthogonal sparse bigram (OSB) transformation, is predominantly used in natural language processing to create features from text data by looking at pairs of words. It's not applicable to handling numerical feature magnitude differences in general machine learning problems.

Normalization is a crucial preprocessing step for many machine learning algorithms, especially those sensitive to feature scaling, such as linear regression, logistic regression, support vector machines (SVMs), and neural networks. These models are designed to work best when the input features are on similar scales.

Normalization ensures that the optimization process converges more efficiently and effectively. It brings all features onto a comparable scale, leading to more stable and better performing models.

https://scikit-learn.org/stable/modules/preprocessing.htmlhttps://developers.google.com/machine-learning/data-preparation/transform/normalization

## Question: 66

A Machine Learning Specialist must build out a process to query a dataset on Amazon S3 using Amazon Athena. The dataset contains more than 800,000 records stored as plaintext CSV files. Each record contains 200 columns and is approximately 1.5 MB in size. Most queries will span 5 to 10 columns only.
How should the Machine Learning Specialist transform the dataset to minimize query runtime?

A.Convert the records to Apache Parquet format.

B.Convert the records to JSON format.

C.Convert the records to GZIP CSV format.

D.Convert the records to XML format.

**Answer: A**

**Explanation:**

The correct answer is A, converting the records to Apache Parquet format. Here's why:

Parquet is a columnar storage format. Unlike row-oriented formats like CSV, JSON, and XML, Parquet stores data by columns. This is a significant advantage for analytical queries that typically access only a subset of columns, as the question specifies ("Most queries will span 5 to 10 columns only"). Athena only reads the columns needed for the query, dramatically reducing I/O and query processing time.

In this scenario, converting to Parquet directly addresses the core requirement of minimizing query runtime. Columnar storage allows Athena to efficiently skip irrelevant data, resulting in faster queries when accessing only a few columns out of the total 200. The large number of records (800,000+) and the record size (1.5 MB) highlight the benefits of columnar storage, as the volume of unnecessary data read with row-oriented formats would be substantial.

Options B, C, and D are less optimal. While GZIP CSV (C) compresses the data and reduces storage costs, it still reads the entire row for each record, mitigating the performance gains of column selection. JSON (B) and XML (D) are both row-oriented and verbose, leading to larger file sizes and slower parsing speeds than Parquet. They also don't offer the columnar selection advantages.

Therefore, utilizing a columnar format like Parquet combined with the selective querying capabilities of Athena significantly optimizes query runtime, making it the best choice. Parquet also supports schema evolution and efficient encoding techniques, further boosting performance.

Further research:

**Apache Parquet:**https://parquet.apache.org/
**Amazon Athena Performance Tuning:**https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html
**Columnar Storage:**https://en.wikipedia.org/wiki/Column-oriented_DBMS

## Question: 67

A Machine Learning Specialist is developing a daily ETL workflow containing multiple ETL jobs. The workflow consists of the following processes:
* Start the workflow as soon as data is uploaded to Amazon S3.
* When all the datasets are available in Amazon S3, start an ETL job to join the uploaded datasets with multiple terabyte-sized datasets already stored in Amazon
S3.
* Store the results of joining datasets in Amazon S3.
* If one of the jobs fails, send a notification to the Administrator.
Which configuration will meet these requirements?

A.Use AWS Lambda to trigger an AWS Step Functions workflow to wait for dataset uploads to complete in Amazon S3. Use AWS Glue to join the datasets. Use an Amazon CloudWatch alarm to send an SNS notification to the Administrator in the case of a failure.

B.Develop the ETL workflow using AWS Lambda to start an Amazon SageMaker notebook instance. Use a lifecycle configuration script to join the datasets and persist the results in Amazon S3. Use an Amazon CloudWatch alarm to send an SNS notification to the Administrator in the case of a failure.

C.Develop the ETL workflow using AWS Batch to trigger the start of ETL jobs when data is uploaded to Amazon S3. Use AWS Glue to join the datasets in Amazon S3. Use an Amazon CloudWatch alarm to send an SNS notification to the Administrator in the case of a failure.

D.Use AWS Lambda to chain other Lambda functions to read and join the datasets in Amazon S3 as soon as the data is uploaded to Amazon S3. Use an Amazon CloudWatch alarm to send an SNS notification to the Administrator in the case of a failure.

**Answer: A**

**Explanation:**

Here's a detailed justification for why option A is the best choice, along with supporting concepts and links:

Option A provides the most robust, scalable, and manageable solution for the described ETL workflow. It leverages purpose-built AWS services designed for each stage of the process.

**AWS Lambda and Step Functions:** Lambda can be triggered by S3 events (object creation) to initiate the workflow. Step Functions orchestrates the ETL process, managing dependencies and state. The "wait" state in Step Functions is ideal for pausing execution until all datasets are available in S3. This provides a reliable mechanism to ensure all required data is present before the next job begins.

https://docs.aws.amazon.com/step-functions/latest/dg/concepts-wait-state.html and
https://docs.aws.amazon.com/lambda/latest/dg/services-s3.html
**AWS Glue:** Glue is designed for data integration and ETL operations. It can efficiently join terabyte-sized datasets stored in S3, leveraging its serverless, Apache Spark-based engine. It also provides data cataloging and schema discovery. https://aws.amazon.com/glue/
**Amazon CloudWatch and SNS:** CloudWatch monitors the ETL jobs (especially Glue) and can be configured with alarms that trigger SNS notifications upon failure. This allows administrators to be promptly alerted to issues.
https://docs.aws.amazon.com/cloudwatch/latest/monitoring/AlarmThatSendsEmail.html

The other options have significant drawbacks:

**Option B:** Using SageMaker notebook instances for ETL is less efficient and more expensive than Glue, especially at the terabyte scale. Notebook instances are more suited for interactive data exploration and model building, not automated ETL. Lifecycle configurations are not ideal for managing complex workflow dependencies.

**Option C:** AWS Batch is primarily designed for batch computing and not the best fit for triggering jobs directly upon S3 object uploads. While it can be incorporated into a workflow, Lambda + Step Functions provide a more event-driven and flexible approach.

**Option D:** Chaining Lambda functions can become complex and difficult to manage, especially for a multi-step ETL process involving terabyte-sized datasets. Lambda functions have execution time limits and are not ideal for long-running ETL tasks or direct manipulation of very large datasets. Lambda's memory limitations also hinder it's ability to work with very large datasets.

In summary, Option A is the most appropriate choice due to its use of serverless components, robust orchestration, scalability for large datasets, and efficient error handling.

## Question: 68

An agency collects census information within a country to determine healthcare and social program needs by province and city. The census form collects responses for approximately 500 questions from each citizen.
Which combination of algorithms would provide the appropriate insights? (Choose two.)

A.The factorization machines (FM) algorithm
B.The Latent Dirichlet Allocation (LDA) algorithm
C.The principal component analysis (PCA) algorithm
D.The k-means algorithm
E.The Random Cut Forest (RCF) algorithm

**Answer: CD**

**Explanation:**

The task requires identifying algorithms to gain insights from a census dataset containing 500 features per citizen, aiming to determine healthcare and social program needs by province and city.

**C. The principal component analysis (PCA) algorithm:** PCA is a dimensionality reduction technique. With 500 features, the dataset likely suffers from high dimensionality. PCA identifies the principal components, which are linear combinations of the original features that capture the most variance in the data. By reducing the number of features, PCA simplifies the data while preserving essential information, making further analysis more efficient and interpretable. It can help identify the most important underlying factors influencing healthcare and social program needs. [https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html]

**D. The k-means algorithm:** K-means is a clustering algorithm. Once PCA has reduced the dimensionality, k-means can group citizens into clusters based on their remaining features (principal components). Each cluster will represent a segment of the population with similar characteristics and needs. Analyzing the characteristics of each cluster (e.g., demographic information, health conditions, social program participation) allows the agency to tailor healthcare and social programs to specific groups within each province and city.

This is more efficient than trying to analyze individual citizen responses for 500 questions across millions of citizens. [https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html]

**Why other options are less suitable:**

**A. The factorization machines (FM) algorithm:** FM is primarily used for recommendation systems and predicting user preferences based on feature interactions. While it could identify feature interactions within the census data, it's not the most direct approach for initial data exploration and segmentation for healthcare and social program needs.

**B. The Latent Dirichlet Allocation (LDA) algorithm:** LDA is primarily used for topic modeling in text data. Since census data is typically structured and numerical, LDA isn't directly applicable.

**E. The Random Cut Forest (RCF) algorithm:** RCF is used for anomaly detection. While useful for identifying unusual responses, it's not the primary technique for understanding the general population's healthcare and social program needs by province and city.

Therefore, PCA followed by k-means provides a structured approach to reduce dimensionality and then segment the population, leading to actionable insights for targeted program development and resource allocation.

## Question: 69

A large consumer goods manufacturer has the following products on sale:
* 34 different toothpaste variants
* 48 different toothbrush variants
* 43 different mouthwash variants

The entire sales history of all these products is available in Amazon S3. Currently, the company is using custom-built autoregressive integrated moving average (ARIMA) models to forecast demand for these products. The company wants to predict the demand for a new product that will soon be launched.
Which solution should a Machine Learning Specialist apply?

A.Train a custom ARIMA model to forecast demand for the new product.

B.Train an Amazon SageMaker DeepAR algorithm to forecast demand for the new product.

C.Train an Amazon SageMaker k-means clustering algorithm to forecast demand for the new product.

D.Train a custom XGBoost model to forecast demand for the new product.

**Answer: B**

**Explanation:**

The most suitable solution is to train an Amazon SageMaker DeepAR algorithm. Here's why:

**DeepAR for Forecasting:** DeepAR is specifically designed for probabilistic forecasting of time series data. It excels at predicting future values based on historical patterns, making it ideal for demand forecasting.

**Handling Multiple Time Series:** DeepAR can efficiently model multiple related time series simultaneously. This is crucial because the company has sales data for various toothpaste, toothbrush, and mouthwash variants.

**Cold Start Problem:** The company is launching a new product with no historical sales data. DeepAR addresses this "cold start" problem by leveraging the historical data of similar existing products to make initial predictions.

**Custom ARIMA Limitations:** While ARIMA models are suitable for individual time series, they don't readily handle multiple related time series or cold start scenarios. Building and maintaining 125 custom ARIMA models (34+48+43) becomes complex and less scalable than a DeepAR approach. Furthermore, ARIMA does not efficiently use information across all time series to improve forecasts for new products.

**XGBoost Inappropriateness:** XGBoost is excellent for classification and regression tasks, but it's not inherently designed for time series forecasting. Applying XGBoost would require significant feature engineering to convert the time series data into a format suitable for XGBoost, making DeepAR a more straightforward and efficient solution.

**K-means Clustering Inapplicability:** k-means clustering is used for grouping data points, not for forecasting future values. It would not be suitable for the demand prediction problem.

**SageMaker Benefits:** Using SageMaker DeepAR simplifies model training, deployment, and management. SageMaker provides the infrastructure and tools needed to efficiently train and deploy the DeepAR model.

Therefore, DeepAR offers the best combination of accuracy, scalability, and ease of implementation for the given demand forecasting scenario, especially with the challenge of a new product launch.

Further Research:

**Amazon SageMaker DeepAR:**https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html
**Probabilistic Time Series Forecasting with Deep Learning:**https://arxiv.org/abs/1704.04110

## Question: 70

A Machine Learning Specialist uploads a dataset to an Amazon S3 bucket protected with server-side encryption using AWS KMS.
How should the ML Specialist define the Amazon SageMaker notebook instance so it can read the same dataset from Amazon S3?

A.Define security group(s) to allow all HTTP inbound/outbound traffic and assign those security group(s) to the Amazon SageMaker notebook instance.

B.|¡onfigure
the Amazon SageMaker notebook instance to have access to the VPC. Grant permission in the KMS key policy to the notebook's KMS
role.

C.Assign an IAM role to the Amazon SageMaker notebook with S3 read access to the dataset. Grant permission in the KMS key policy to that role.

D.Assign the same KMS key used to encrypt data in Amazon S3 to the Amazon SageMaker notebook instance.

**Answer: C**

**Explanation:**

The correct answer is C. Here's why:

Amazon S3 objects encrypted with KMS keys require proper authorization for access. SageMaker notebook instances, like other AWS services, need appropriate IAM roles to interact with resources like S3. Directly assigning the KMS key to the notebook instance (option D) doesn't grant permissions; KMS keys aren't directly attached to EC2 instances or notebook instances. They're used for encryption/decryption operations controlled by IAM roles.

Option A is incorrect because opening all HTTP traffic in security groups poses significant security risks. Security groups should only allow necessary traffic. Also, security groups alone don't handle KMS decryption permissions.

Option B is partially correct. VPC configuration might be needed in some scenarios, especially if S3 has VPC endpoints. However, granting permission in the KMS key policy to the notebook's KMS role is not quite accurate. SageMaker notebooks don't inherently have a KMS role; instead, the IAM role assigned to the notebook should be granted KMS decryption permissions.

Option C addresses the core requirement: providing the notebook instance with the necessary permissions. By assigning an IAM role to the SageMaker notebook instance with s3:GetObject (read) access to the specific S3 bucket and objects, you allow it to access the encrypted data. Crucially, granting permission within the KMS key policy to that role (the notebook's IAM role) authorizes the role to use the KMS key to decrypt the S3 objects. This aligns with the principle of least privilege – granting only the permissions necessary to perform the task. The KMS key policy must explicitly allow the notebook instance's IAM role to perform kms:Decrypt on the key.

In summary, accessing KMS-encrypted S3 data from a SageMaker notebook instance requires:

1. An IAM role attached to the SageMaker notebook instance.
2. The IAM role having s3:GetObject permissions on the S3 bucket/objects.
3. The KMS key policy granting the IAM role kms:Decrypt permission.

This approach securely grants the notebook instance the necessary permissions to access the encrypted data without compromising security.

Relevant Documentation:

**IAM roles:**https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html
**KMS key policies:**https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html **SageMaker IAM roles:**https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-roles.html **S3 Bucket Permissions:**https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-iam.html

## Question: 71

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 to 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 in Amazon S3 that is accessible for 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 PySpark 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 real-time 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:**

Here's a detailed justification for why option B is the best solution, along with supporting concepts and links:

Option B leverages AWS Glue, a fully managed ETL (Extract, Transform, Load) service, which directly addresses the requirement of minimizing server management. Glue allows Data Scientists to author ETL jobs using PySpark, enabling reuse of existing logic with minimal code modification. A Glue trigger can be scheduled to execute the ETL job at regular intervals, meeting the scheduling requirement. Glue natively integrates with Amazon S3, allowing it to read raw data from S3 and write processed data back to S3. Glue's serverless nature eliminates the need to provision and manage EC2 instances or EMR clusters, simplifying the operational overhead.

Option A, while utilizing EMR, involves managing a persistent cluster, contradicting the minimization of server management. Lambda functions, while serverless, have execution time limits and may not be suitable for large data processing tasks inherent in ETL. Furthermore, orchestrating Spark jobs from Lambda adds complexity.

Option C attempts to use Lambda for the entire ETL process. While Lambda is serverless, it's not designed for large-scale data transformation. PySpark is designed to run on a distributed cluster. Trying to reimplement the PySpark logic inside a single Lambda function defeats the purpose and it exceeds the execution time limitations for Lambda.

Option D utilizes Kinesis Data Analytics, which is optimized for real-time data streaming and analysis. This is not the requirement. The data is processed at specific intervals so real time analytics is not applicable.

In summary, AWS Glue provides a serverless, fully managed environment perfectly suited for running PySpark-based ETL jobs on a schedule, fulfilling all the given requirements efficiently.

Relevant links:

**AWS Glue Documentation:**https://aws.amazon.com/glue/
**AWS Glue ETL Jobs:**https://docs.aws.amazon.com/glue/latest/dg/add-job.html
**AWS Glue Triggers:**https://docs.aws.amazon.com/glue/latest/dg/glue-triggers.html
**Amazon EMR Documentation:**https://aws.amazon.com/emr/
**AWS Lambda Documentation:**https://aws.amazon.com/lambda/
**Amazon Kinesis Data Analytics Documentation:**https://aws.amazon.com/kinesis/data-analytics/

**Question: 72**

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: AC**

**Explanation:**

The Data Scientist faces two main challenges: high variance (overfitting) in the logistic regression model indicated by the gap between training and validation accuracy, and the need for model interpretability to satisfy the Marketing team. L1 regularization (Option A) addresses both concerns. L1 regularization adds a penalty to the model's loss function proportional to the absolute value of the coefficients. This encourages sparsity in the model, effectively shrinking the coefficients of less important features to zero. This reduces model complexity, mitigating overfitting and improving generalization to the validation set. Because L1 regularization zeroes out unimportant features, it also performs feature selection, making the model easier to interpret as the Marketing team can focus only on the non-zero coefficients, which directly show the impact of each feature on the churn prediction.

Recursive feature elimination (RFE) (Option C) is another valid approach. RFE works by iteratively training a model, ranking features based on their importance, and removing the least important feature(s) until a desired number of features is reached. This process helps identify the most relevant features for prediction, addressing the Marketing team's need for interpretable features that have a direct impact on model outcome, while also reducing dimensionality.

Option B (Adding features) is unlikely to improve the model's performance in this case. The model is already overfitting, and adding more features would likely exacerbate the problem. Furthermore, it would make the model less interpretable. Option D (t-SNE) is a dimensionality reduction technique primarily used for visualizing high-dimensional data in lower dimensions (e.g., 2D or 3D). It's not directly helpful for improving the model's predictive performance or model interpretability. Option E (Linear Discriminant Analysis) is a dimensionality reduction technique and classification algorithm but may not be as effective as L1
regularization or RFE for feature selection and regularization, especially when dealing with a large number of features. It primarily focuses on maximizing class separability rather than sparsity.

Therefore, L1 regularization tackles both the overfitting and interpretability issues by reducing model complexity through feature selection and revealing the direct impact of the remaining features on the prediction. RFE helps identify features with impact.

Relevant Links:

**L1 Regularization:**https://scikit-learn.org/stable/modules/linear_model.html#lasso
**Recursive Feature Elimination:**https://scikit-learn.org/stable/modules/feature_selection.html#recursive-feature-elimination

## Question: 73

An aircraft engine manufacturing company is measuring 200 performance metrics in a time-series. Engineers want to detect critical manufacturing defects in near- real time during testing. All of the data needs to be stored for offline analysis. What approach would be the MOST effective to perform near-real time defect detection?

A.Use AWS IoT Analytics for ingestion, storage, and further analysis. Use Jupyter notebooks from within AWS IoT Analytics to carry out analysis for anomalies.

B.Use Amazon S3 for ingestion, storage, and further analysis. Use an Amazon EMR cluster to carry out Apache Spark ML k-means clustering to determine anomalies.

C.Use Amazon S3 for ingestion, storage, and further analysis. Use the Amazon SageMaker Random Cut Forest (RCF) algorithm to determine anomalies.

D.Use Amazon Kinesis Data Firehose for ingestion and Amazon Kinesis Data Analytics Random Cut Forest (RCF) to perform anomaly detection. Use Kinesis Data Firehose to store data in Amazon S3 for further analysis.

**Answer: D**

**Explanation:**

The correct answer is D because it outlines the most effective approach for near real-time defect detection in time-series data with subsequent storage for offline analysis. Kinesis Data Firehose excels at ingesting streaming data and delivering it to destinations like S3. Kinesis Data Analytics, specifically the Random Cut Forest (RCF) algorithm, is designed for real-time anomaly detection in streaming data. RCF is well-suited for identifying deviations in high-dimensional data, which aligns with the 200 performance metrics described in the scenario. This allows for immediate identification of critical manufacturing defects during testing. The use of Kinesis Data Firehose to simultaneously store the data in S3 enables offline analysis, fulfilling all requirements of the problem statement.

Option A is less suitable because AWS IoT Analytics, while capable of ingestion and storage, is not optimized for the near real-time anomaly detection required here. Jupyter notebooks, while versatile, are not ideal for continuous real-time analysis compared to Kinesis Data Analytics. Option B, using S3 for ingestion, lacks the necessary real-time processing capabilities. While EMR with Spark ML can perform anomaly detection, it is less suited for low-latency, continuous analysis than Kinesis Data Analytics. Option C, storing in S3 directly for ingestion is inappropriate for streaming data. SageMaker RCF is suitable for anomaly detection, but not in the direct context of real-time streaming data ingestion. It would require building custom ingestion pipeline to process the data from S3.

Key considerations include the need for real-time anomaly detection (Kinesis Data Analytics), handling of streaming data (Kinesis Data Firehose), suitability of RCF for anomaly detection (Kinesis Data Analytics), and the requirement for data storage for offline analysis (Kinesis Data Firehose -> S3).

Further Research:

**Amazon Kinesis Data Firehose:**https://aws.amazon.com/kinesis/data-firehose/
**Amazon Kinesis Data Analytics:**https://aws.amazon.com/kinesis/data-analytics/
**Random Cut Forest (RCF) Algorithm:**https://docs.aws.amazon.com/sagemaker/latest/dg/anomaly-detection.html

## Question: 74

A Machine Learning team runs its own training algorithm on Amazon SageMaker. The training algorithm requires external assets. The team needs to submit both its own algorithm code and algorithm-specific parameters to Amazon SageMaker. What combination of services should the team use to build a custom algorithm in Amazon SageMaker? (Choose two.)

A.AWS Secrets Manager

B.AWS CodeStar
C.Amazon ECR
D.Amazon ECS
E.Amazon S3

**Answer: CE**

**Explanation:**

The correct answer is C and E: Amazon ECR and Amazon S3. Let's break down why.

**Amazon ECR (Elastic Container Registry):**

SageMaker custom algorithms often involve packaging your training code and its dependencies into Docker containers. ECR is a fully managed Docker container registry that allows you to easily store, manage, and deploy Docker container images.

The team can containerize their training algorithm code, including any custom libraries or dependencies, and then push the resulting Docker image to ECR. SageMaker can then pull this image and use it to run training jobs. This makes the training environment consistent and portable.

Using containers makes deploying custom algorithms and defining its environment repeatable and simplified.

https://aws.amazon.com/ecr/

**Amazon S3 (Simple Storage Service):**

S3 is an object storage service used for storing data such as training datasets, model artifacts, and in this case, algorithm-specific parameters.

The Machine Learning team can store algorithm-specific parameters, configuration files, or external assets needed by the training algorithm in S3.

SageMaker can access these parameters during the training process, enabling the algorithm to be configured as needed. S3 is a central place to store data for training, including both the training data and the metadata for the training job.

https://aws.amazon.com/s3/

**Why the other options are incorrect:**

**A. AWS Secrets Manager:** Secrets Manager is for managing secrets (e.g., database credentials, API keys), not for storing algorithm parameters or container images. While credentials could be passed to the training job via secrets manager, it's not a primary component for building a custom algorithm itself.

**B. AWS CodeStar:** CodeStar is a service for quickly developing, building, and deploying applications on AWS. It helps set up CI/CD pipelines. While useful for software development in general, it's not a direct requirement for creating a custom SageMaker algorithm. The focus of the problem is storage and execution environment of the algorithm.

**D. Amazon ECS (Elastic Container Service):** ECS is a container orchestration service, but for the specific task of creating and using a custom SageMaker training algorithm, ECR for image storage and S3 for parameters are more directly relevant. ECS is focused on running containerized applications, while the team here is focused on packaging a training algorithm for SageMaker.

In summary, ECR is used to store and manage the containerized algorithm code, and S3 is used to store algorithm-specific parameters. These two services, when combined, allow the Machine Learning team to build and execute their custom algorithm within SageMaker.

**Question: 75**

A Machine Learning Specialist wants to determine the appropriate SageMakerVariantInvocationsPerInstance setting for an endpoint automatic scaling configuration. The Specialist has performed a load test on a single instance and determined that peak requests per second (RPS) without service degradation is about 20 RPS. As this is the first deployment, the Specialist intends to set the invocation safety factor to 0.5.
Based on the stated parameters and given that the invocations per instance setting is measured on a per-minute basis, what should the Specialist set as the
SageMakerVariantInvocationsPerInstance setting?

A.10
B.30
C.600
D.2,400

**Answer: C**

**Explanation:**

Here's a detailed justification for why the correct answer is C (600):

The question requires calculating the appropriate SageMakerVariantInvocationsPerInstance setting for SageMaker endpoint autoscaling, considering the safe RPS a single instance can handle. The provided information includes: peak RPS per instance (20), invocation safety factor (0.5), and the fact that the setting is measured on a per-minute basis.

First, calculate the safe RPS per instance by applying the safety factor: Safe RPS = Peak RPS Safety Factor = 20 RPS 0.5 = 10 RPS. This step accounts for variance and ensures endpoint stability during scaling.

Next, convert the safe RPS to invocations per minute. Since there are 60 seconds in a minute, multiply the safe RPS by 60: Invocations per Minute = Safe RPS 60 seconds/minute = 10 RPS 60 seconds/minute = 600 invocations/minute.

Therefore, the SageMakerVariantInvocationsPerInstance setting should be set to 600. This means that SageMaker Auto Scaling will attempt to keep the number of invocations per instance at or below 600 per minute. If the actual invocations per instance exceed this threshold, SageMaker Auto Scaling will initiate a scale-out event (adding more instances) to distribute the load. The safety factor is crucial in preventing overload and maintaining performance. Setting it lower would result in more scaling operations and higher costs. A higher value may lead to reduced performance.

Options A, B, and D are incorrect as they do not properly account for the safety factor and/or the conversion from requests per second to requests per minute. A and B disregard converting RPS to RPM and fail to use the safety factor effectively. D incorrectly calculates the Invocations per minute.

Relevant Documentation:

**SageMaker Automatic Scaling:**https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling.html
**Target Tracking Scaling Policies:**https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html

## Question: 76

A company uses a long short-term memory (LSTM) model to evaluate the risk factors of a particular energy sector. The model reviews multi-page text documents to analyze each sentence of the text and categorize it as either a potential risk or no risk. The model is not performing well, even though the Data Scientist has experimented with many different network structures and tuned the corresponding hyperparameters.
Which approach will provide the MAXIMUM performance boost?

A.Initialize the words by term frequency-inverse document frequency (TF-IDF) vectors pretrained on a large collection of news articles related to the energy sector.

B.Use gated recurrent units (GRUs) instead of LSTM and run the training process until the validation loss stops decreasing.

C.Reduce the learning rate and run the training process until the training loss stops decreasing.

D.Initialize the words by word2vec embeddings pretrained on a large collection of news articles related to the energy sector.

**Answer: D**

**Explanation:**

The question addresses improving the performance of an LSTM model for risk factor evaluation in the energy sector, specifically analyzing text documents to classify sentences as potential risks or no risks. While the data scientist has tuned the model architecture and hyperparameters, the model's performance is still suboptimal.

The most impactful approach is initializing word embeddings with pre-trained word2vec vectors (Option D). Here's why:

1. **Pre-trained Word Embeddings:** Word2vec is a technique for learning distributed representations of words, capturing semantic relationships. By pre-training word2vec on a large corpus of energy sector news articles, the model gains a contextual understanding of relevant terminology and their interrelations before training on the specific risk analysis task. This provides a solid foundation.

2. **Knowledge Transfer:** The model benefits from knowledge learned from a vast dataset, transferring this knowledge to the target task. This is particularly useful when the target dataset is relatively small, as it reduces the reliance on learning word representations from scratch.

3. **Improved Generalization:** Pre-trained embeddings enhance generalization, as the model can better handle unseen words or phrases that are semantically similar to those in the pre-training corpus.

4. **TF-IDF Limitations (Option A):** TF-IDF (term frequency-inverse document frequency) measures the importance of a term in a document relative to a collection of documents. While useful for information retrieval, it doesn't capture semantic relationships between words like word2vec does. It's a bag-of-words model, not a contextual embedding.

5. **GRU vs. LSTM (Option B):** GRUs are a simplified version of LSTMs and might offer slight performance improvements in some cases, but the core issue here isn't model architecture complexity. The lack of rich word representations is a more significant bottleneck.

6. **Learning Rate Adjustment (Option C):** Reducing the learning rate can help fine-tune the model and prevent overfitting, but it addresses optimization issues rather than the fundamental issue of poor word representations. It is a step in the right direction, but a smaller gain than Option D.

Therefore, initializing with word2vec embeddings allows the model to leverage existing knowledge and understand the nuances of the energy sector's language, resulting in a significant performance boost by providing a superior starting point for the LSTM network.

**Authoritative Links:**

**Word2Vec:**https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf **Using pre-trained word embeddings:**https://towardsdatascience.com/using-pre-trained-word-embeddings-in-pytorch-c253eb6ff100

## Question: 77

A Machine Learning Specialist needs to move and transform data in preparation for training. Some of the data needs to be processed in near-real time, and other data can be moved hourly. There are existing Amazon EMR MapReduce jobs to clean and feature engineering to perform on the data.
Which of the following services can feed data to the MapReduce jobs? (Choose two.)

A.AWS DMS
B.Amazon Kinesis
C.AWS Data Pipeline
D.Amazon Athena
E.Amazon ES

**Answer: BC**

**Explanation:**

The correct answers are B and C: Amazon Kinesis and AWS Data Pipeline. Here's why:

**Amazon Kinesis:** Kinesis is designed for real-time data streaming. It can ingest, buffer, and process high-velocity data streams, making it suitable for feeding near-real-time data to MapReduce jobs in EMR. Kinesis Data Streams, specifically, is well-suited for handling continuous data flow.
https://aws.amazon.com/kinesis/data-streams/

**AWS Data Pipeline:** Data Pipeline is a managed orchestration service for data movement and transformation. It allows you to create complex data workflows with dependencies, scheduling, and error handling. It can schedule hourly or other batch-oriented data movements to EMR for processing. It is ideal for regularly moving data, running scripts, or initiating EMR jobs. https://aws.amazon.com/datapipeline/

Here's why the other options are not the best fit:

**AWS DMS (Database Migration Service):** DMS is primarily for migrating databases to AWS or between databases. While it can move data, it's not the most efficient or appropriate choice for constantly feeding data to EMR jobs, particularly in near-real-time.

**Amazon Athena:** Athena is a query service that allows you to analyze data directly from S3 using SQL. It is more suitable for querying existing data lakes, not for moving data to EMR.

**Amazon ES (Elasticsearch Service):** Elasticsearch Service is a search and analytics engine. It's designed for indexing and searching data, not for feeding data to MapReduce jobs. It would be more relevant if the requirement was to search the processed data after it has been refined in EMR.

## Question: 78

A Machine Learning Specialist previously trained a logistic regression model using scikit-learn on a local machine, and the Specialist now wants to deploy it to production for inference only.
What steps should be taken to ensure Amazon SageMaker can host a model that was trained locally?

A.Build the Docker image with the inference code. Tag the Docker image with the registry hostname and upload it to Amazon ECR.

B.Serialize the trained model so the format is compressed for deployment. Tag the Docker image with the registry hostname and upload it to Amazon S3.

C.Serialize the trained model so the format is compressed for deployment. Build the image and upload it to Docker Hub.

D.Build the Docker image with the inference code. Configure Docker Hub and upload the image to Amazon ECR.

**Answer: A**

**Explanation:**

The correct answer is A. Here's a detailed justification:

To deploy a locally trained scikit-learn model on SageMaker for inference, the model needs to be packaged in a way that SageMaker can understand and execute. This involves containerizing the model and its inference code within a Docker image.

**Why Docker?** SageMaker uses Docker containers to provide a consistent and reproducible environment for hosting models. Docker allows you to package the model, its dependencies (like scikit-learn), and the inference script into a single, self-contained unit. This ensures the model runs the same way in SageMaker as it did locally.

**Steps involved:**

1. **Build the Docker Image:** The Docker image should contain:

   The serialized (saved) model file (e.g., using joblib or pickle).
   An inference script (e.g., inference.py) that loads the model and defines how to perform inference. This script will receive prediction requests and return predictions.
   Any necessary dependencies specified in a requirements file (e.g., requirements.txt).
   A web server like Flask or Gunicorn that serves prediction requests.

2. **Tag the Docker Image:** Before pushing the image, you must tag it using your AWS account ID and the region's Elastic Container Registry (ECR) hostname. This tells Docker where to push the image.

3. **Upload to Amazon ECR:** Amazon ECR is a fully managed Docker container registry that allows you to store, manage, and deploy Docker container images. By pushing the image to ECR, you make it accessible to SageMaker. SageMaker can then pull the image from ECR and use it to deploy the model endpoint.

**Why other options are incorrect:**

**B:** While serializing the model is important, uploading the image directly to S3 is not how SageMaker deploys Docker container-based models. ECR is the dedicated container registry service for AWS.

**C:** Docker Hub is a public container registry. While you could potentially use it, it's generally recommended to use ECR for private model deployments within AWS for security and access control reasons. Also, building an image without inference code is incorrect.

**D:** Configuring Docker Hub is irrelevant since the image needs to be in Amazon ECR, and you must have inference code in the docker image.

**Supporting Links:**

**SageMaker Inference with Custom Docker Containers:**
https://docs.aws.amazon.com/sagemaker/latest/dg/CreateModel.html
**Amazon ECR:**https://aws.amazon.com/ecr/
**SageMaker Scikit-learn example:**https://sagemaker-examples.readthedocs.io/en/latest/sagemaker-python-sdk/scikit_learn_inference_pipeline/scikit_learn_inference_pipeline.html

## Question: 79

A trucking company is collecting live image data from its fleet of trucks across the globe. The data is growing rapidly and approximately 100 GB of new data is generated every day. The company wants to explore machine learning uses cases while ensuring the data is only accessible to specific IAM users.

Which storage option provides the most processing flexibility and will allow access control with IAM?

A.Use a database, such as Amazon DynamoDB, to store the images, and set the IAM policies to restrict access to only the desired IAM users.

B.Use an Amazon S3-backed data lake to store the raw images, and set up the permissions using bucket policies.

C.Setup up Amazon EMR with Hadoop Distributed File System (HDFS) to store the files, and restrict access to the EMR instances using IAM policies.

D.Configure Amazon EFS with IAM policies to make the data available to Amazon EC2 instances owned by the IAM users.

**Answer: B**

**Explanation:**

The correct answer is B: Use an Amazon S3-backed data lake to store the raw images, and set up the permissions using bucket policies. Here's a detailed justification:

**Scalability and Cost-Effectiveness:** Amazon S3 is designed for massive scalability and cost-effective storage of large amounts of unstructured data like images. It's a highly suitable choice for the 100 GB daily data ingestion. Data lakes are commonly built on S3 due to its scalability and cost efficiency for large data sets.

**IAM Integration:** S3 offers robust integration with IAM, enabling granular access control. You can define bucket policies and IAM policies that specify which IAM users or roles have permissions to access, read, write, or delete objects within the S3 bucket. This directly addresses the requirement of restricting data access to specific IAM users.

**Processing Flexibility:** Storing the data in its raw format within S3 provides maximum flexibility for future machine learning use cases. You can readily process the data using various AWS services like AWS Glue, Amazon SageMaker, Amazon EMR, or AWS Lambda.

**Data Lake Architecture:** Building a data lake on S3 allows you to store data in its raw format without requiring upfront transformation, which aligns with the company's need to explore machine learning use cases before fully defining the processing pipeline.

**DynamoDB (Option A):** While DynamoDB supports IAM policies, storing images directly in DynamoDB can be less cost-effective and may require more complex data modeling compared to storing them as objects in S3. DynamoDB is better suited for structured data.

**Amazon EMR with HDFS (Option C):** Setting up Amazon EMR solely for storage using HDFS is overkill for this scenario. EMR is designed for large-scale data processing, not primarily for storage. HDFS can be more complex to manage than S3.

**Amazon EFS (Option D):** Amazon EFS is a file system designed for shared access by multiple EC2 instances.

While EFS supports IAM policies, it is more appropriate when you need file-system semantics and shared access from multiple compute instances. Using EFS solely for storage can be less cost-effective than S3, especially for data that is accessed less frequently.

Therefore, using an S3-backed data lake offers the best balance of scalability, cost-effectiveness, IAM integration, and processing flexibility for storing and accessing the trucking company's image data.

Authoritative links:

Amazon S3 Documentation
IAM Policies for S3
AWS Data Lake

## Question: 80

A credit card company wants to build a credit scoring model to help predict whether a new credit card applicant will default on a credit card payment. The company has collected data from a large number of sources with

thousands of raw attributes. Early experiments to train a classification model revealed that many attributes are highly correlated, the large number of features slows down the training speed significantly, and that there are some overfitting issues.
The Data Scientist on this project would like to speed up the model training time without losing a lot of information from the original dataset.
Which feature engineering technique should the Data Scientist use to meet the objectives?

A.Run self-correlation on all features and remove highly correlated features

B.Normalize all numerical values to be between 0 and 1

C.Use an autoencoder or principal component analysis (PCA) to replace original features with new features

D.Cluster raw data using k-means and use sample data from each cluster to build a new dataset

**Answer: C**

**Explanation:**

The correct answer is **C. Use an autoencoder or principal component analysis (PCA) to replace original features with new features.**

Here's why:

The problem describes a dataset with high dimensionality, high correlation between features, slow training times, and overfitting. These issues are classic scenarios where dimensionality reduction techniques are beneficial.

**PCA and Autoencoders for Dimensionality Reduction:** PCA and autoencoders are feature extraction methods specifically designed to reduce the number of features while retaining as much variance (information) as possible. PCA achieves this through linear transformations, creating uncorrelated principal components. Autoencoders, particularly non-linear ones, can capture more complex relationships and potentially represent data more compactly.

**Addressing the Problems:** These techniques directly address the stated issues:

Reduced Training Time: Fewer features means faster model training.

Correlation Handling: Both PCA and autoencoders aim to create new, less correlated features. PCA guarantees uncorrelated features, while autoencoders learn compressed representations that ideally remove redundancies.

Overfitting Mitigation: By reducing the dimensionality, the model has fewer parameters to fit, decreasing the risk of overfitting, especially when dealing with a limited dataset size relative to the number of features.

Let's analyze the other options:

**A. Run self-correlation on all features and remove highly correlated features:** This is a form of feature selection. While removing highly correlated features is helpful, it doesn't necessarily address the high dimensionality as effectively as PCA or autoencoders. It can also lead to information loss if you're just removing features without capturing their underlying variance into new features.

**B. Normalize all numerical values to be between 0 and 1:** Normalization is a good preprocessing step for many machine learning algorithms, especially gradient descent-based ones, as it helps with faster convergence. However, it doesn't reduce the number of features or directly address the correlation issue. It primarily helps improve the performance of certain algorithms on the existing feature set, but doesn't speed up training due to feature reduction or prevent overfitting due to fewer features.

**Why C is superior:** PCA and autoencoders perform feature extraction, creating a new, lower-dimensional feature space that captures the essence of the original data. This is more effective at reducing dimensionality and mitigating the issues described than simply selecting a subset of the original features.

In summary, PCA and autoencoders are the most appropriate feature engineering techniques to address high

dimensionality, high correlation, slow training, and overfitting issues in the context of this credit card scoring model.

**Authoritative Links:**

**PCA:**https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
**Autoencoders:**https://www.tensorflow.org/tutorials/generative/autoencoder