AWS Machine Learning Specialty

Cheat Sheet

Fabrice Guillaume - Nov 2020

S3:

- Files can be from 0 bytes to 5Tb
- Unlimited storage
- Universal namespace (names must be unique globally)

DBs:

- **RDS:** Relational DB (MySQL, SQL Server, Oracle, Postgres, Aurora, MariaDB)
- **DynamoDB**: NoSQL
- **RedShift**: BI datawarehouse / Redshift spectrum can query data from S3 and we acn use Quicksight on top for visualization
- **TimeStream**: to handle time series data
- **DocumentDB**: for Mongo

AWS Migration Tools:

- **Data Pipeline**: NEED EC2 instances and/or EMR. Used for creating ETL workflows to automate processing of data at scheduled intervals, then terminate the resources.

  AWS data pipeline can be used to move data from one dynamoDB table in one region to another region. Behind the scenes it launches EMR cluster -> stores in S3 -> starts EMR in the other region, Loads from S3 and sends to dynamoDB.

  - Schedule regular data movement and data processing activities in AWS. Integrates with on-premise and cloud-based storage.
  - A managed ETL service with native integration with S3, DynamoDB, RDS, ENR, EC2 and Redshift
  - set up activities (Copy, EMR, Hadoop, Hive, Redshift, SQL..), specify data source and how to migrate the data. Can run on demand or on schedule.
  - Ex: full copy of RDS MySQL to S3 or Redshit. Export DynamoDB to S3...

- **DMS**: migrate data from your database that is on-premises, on an Amazon Relational Database Service (Amazon RDS) DB instance, or in a database on an Amazon Elastic Compute Cloud (Amazon EC2) instance to a database on an AWS service
  - It can move data from a DB to S3 too
  - DMS does NOT transfer any transformation Other than just column name change

- **Glue**: fully managed ETL that makes it simple and cost effective to categorize your data, clean it, enrich it and move it reliably between various data stores
    - Crawlers help infer the schema and build the catalog. Built-in classifiers can read Apache ORC, Apache ORC, Parquet, JSON, CSV,...
    - Format can easily be changed to avro, csv, parquet, xml..
    - Code is Python or Scala (job creates pyspark code for us)
    - Can create Zeppelin notebooks to do ad-hoc transformations

AWS Helper Tools

- **EMR**: to run big data transforms across multiple EC2 using different frameworks: Spark, TensorFlow, Hive, ...
- **Athena**: serverless platform to run SQL on S3

Streaming Services

- **Kinesis Data Streams:** get streaming data from data producers (IoT device, click stream,...) and leverages **shards**. Once data is in shard, data consumers can process this data such as EC2, Lambda, Kinesis Data Analytics or EMR. Multiple consumers can consume from a shard.
    - Shard made of: unique partition key and a sequence # (+ the data blob, up to 1Mb)
    - Data in shards
        - is held for 24 hours by default, up to 7 days (168 hours)
        - Can be ingested at 1000 records / sec or up to 1 Mb/s
        - Emits up to 2Mb / sec
        - Enhanced fan-out is an optional feature for Kinesis Data Streams consumers that provides logical 2 MB/sec throughput pipes between consumers and shards
    - # of requests will determine the # of shards required
    - Default # shards
    - Interaction with Kinesis stream via:
        - **KPL**: Kinesis Producer Library to write. Java wrapper with automatic and configurable retry mechanism. Can install it onto EC2 instance or in our Java app. Additional processing delay can occur
        - **Kinesis API (SDK)**: low level API calls (PutRecord and PutRecords). No delays in processing. Any AWS SDK (not limited to Java).
        - **KCL**: Kinesis Client Library: to interact with KPL to consume and process the data

- **Kinesis Firehose:** Also helps with streaming data, but it is fully managed: no longer have to deal with shards. Data can also be pre-processed leveraging Lambda. Finally it can be used as a way to store directly into S3 or Redshift. We can use S3 events to then load data into DynamoDB from S3.

    - Careful: no data retention in Firehose (we can’t replay as opposed to Data Steam)
    - Careful: akso not as “real-time” as Kinesis data stream - delay.
    - Can use Kinesis Agent (java app) that handles retries and everything, or the PutRecord(s) API call

- **Kinesis Video Streams:** to stream video feeds from security cameras, web cams, audio feeds,... Helps streams the data to S3 or to other AWS services (Rekognition)
- **Kinesis Data Analytics:** does NOT help produce stream and needs to feed of Kinesis data stream or Kinesis Firehose.
    - Used to run real time SQL on streaming data as well as some ML (forecasing, RCF anomaly detection).
    - Output data to S3 or Redshift.
    - Can also perform ETL work: clean, enrich, organize, and transform raw data prior to loading your data lake or data warehouse in real-time, reducing or eliminating batch ETL steps. For example, you can build an application that continuously reads IoT sensor data stored in Amazon Managed Streaming for Apache Kafka (Amazon MSK), organize the data by sensor type, remove duplicate data, normalizes data per a specified schema, and then deliver the data to Amazon S3.

NLP

N-GRAM splits text by whitespace with window size of N => for matching phrases in spam email. Can be used for WHOLE phrases, not just words.

TF-IDF helps us determine how important words are within multiple documents => for determining the subject matter of multiple PDFs (find common word combinations repeated throughout PDFs)

OSB splits text by keeping first word and uses delimiter with remaining white spaces between second work with window size N => for determining the subject matter of multiple PDFs (find common word combinations repeated throughout PDFs)

Feature Scaling: Normalization Vs Standardization

Source, source2

In both cases, you're transforming the values of numeric variables so that the transformed data points have specific helpful properties. The difference is that:

- in **scaling**, you're changing the ***range*** of your data. By scaling your variables, you can help compare different variables on equal footing. You want to scale data when you're using methods based on measures of how far apart data points, like support vector machines, or SVM or k-nearest neighbors, or KNN. With these algorithms, a change of "1" in any numeric feature is given the same importance. For example, you might be looking at the prices of some products in both Yen and US Dollars. One US Dollar is worth about 100 Yen, but if you don't scale your prices methods like SVM or KNN will consider a difference in price of 1 Yen as important as a difference of 1 US Dollar! This clearly doesn't fit with our intuitions of the world.

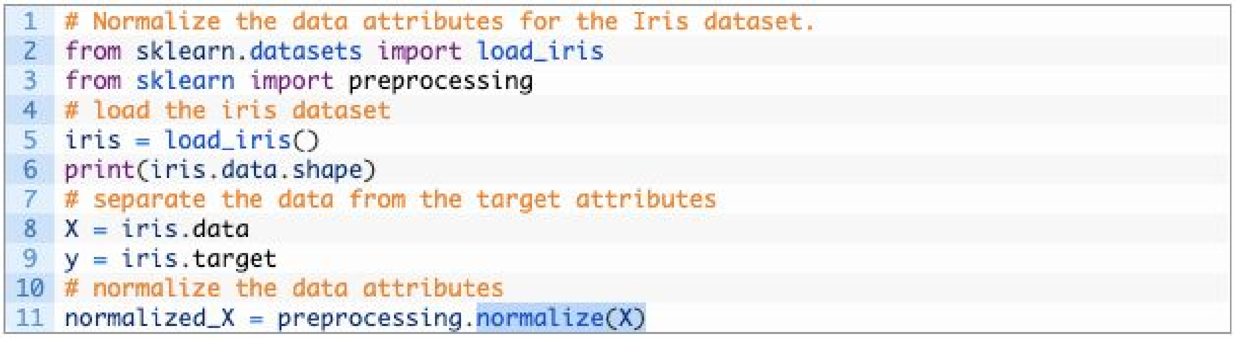
- while in **normalization** you're changing the ***shape*** *of the distribution* of your data. Scaling just changes the range of your data. Normalization is a more radical transformation. The point of normalization is to **change** your observations so that they can be described as a **normal distribution**. In general, you'll only want to normalize your data if you're going to be using a machine learning or statistics technique that assumes your data is normally distributed. Some examples of these include t-tests, ANOVAs, linear regression, linear discriminant analysis (LDA) and Gaussian naive Bayes. (Pro tip: any method with "Gaussian" in the name probably assumes normality.)

Feature Scaling:

- **Normalization** => scale from **0 to 1**, but does NOT handle outliers well
- **Standardization** => uses the avg(mean) as 0 and unit variance (standard deviation of one), other values scaled by their z-score => much less affected by outliers. Some values will be negative.

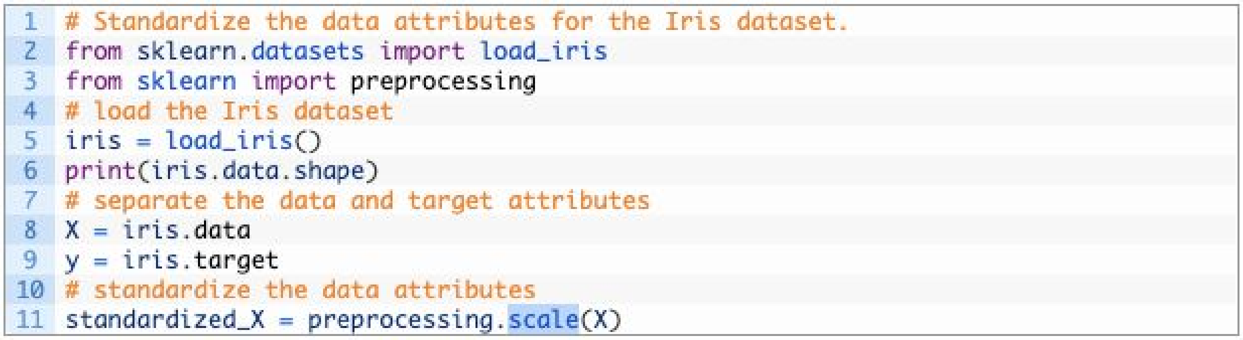
ScitKitLearn:

- Normalization: `sklearn.preprocessing`**`.normalize`**



```
# Normalize the data attributes for the Iris dataset.
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data from the target attributes
X = iris.data
y = iris.target
# normalize the data attributes
normalized_X = preprocessing.normalize(X)
```

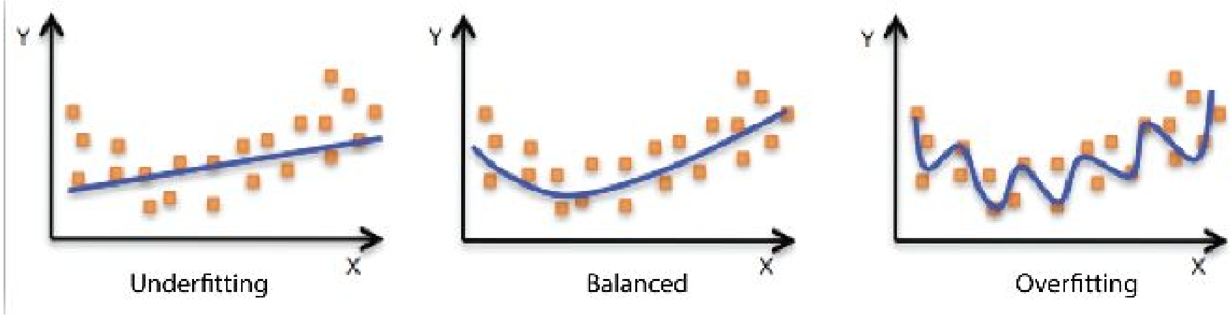
- Standardization: `sklearn.preprocessing`**`.StandardScaler or Scale`**



```
# Standardize the data attributes for the Iris dataset.
from sklearn.datasets import load_iris
from sklearn import preprocessing
# load the Iris dataset
iris = load_iris()
print(iris.data.shape)
# separate the data and target attributes
X = iris.data
y = iris.target
# standardize the data attributes
standardized_X = preprocessing.scale(X)
```

Addressing UnderFitting and OverFitting

https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html



Your model is **underfitting** the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y).

Your model is **overfitting** your training data when you see that the model performs well on the training data but does not perform well on the evaluation data. This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.

If your model is **underfitting => poor performance on the training** data could be because the model is too simple (the input features are not expressive enough) to describe the target well. Performance can be improved by **increasing model flexibility**. To increase model flexibility, try the following:

- Add **new domain-specific features** and **more feature** Cartesian products, and change the types of **feature processing** used (e.g., increasing n-grams size)
- **Decrease** the amount of **regularization** used

If your model is **overfitting the training** data, it makes sense to take actions that r**educe model flexibility**. To reduce model flexibility, try the following:

- **Feature selection**: consider using fewer feature combinations, decrease n-grams size, and decrease the number of numeric attribute bins.
- **Increase** the amount of **regularization** used.

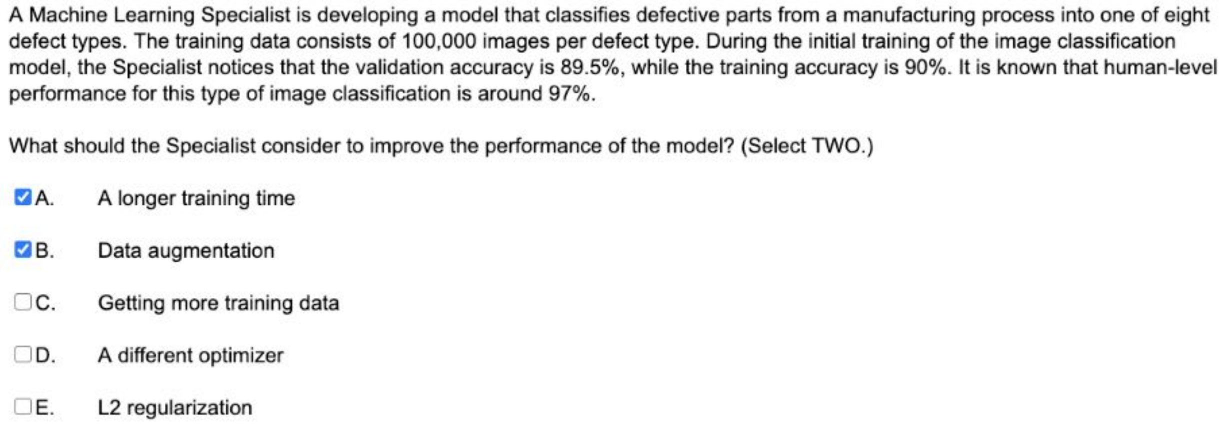
Accuracy on training and test data could be poor because the learning algorithm did **not have enough** data to learn from. You could improve performance by doing the following:

- **Increase the amount of training data** examples.
- **Increase the number of passes** on the existing training data.

Example for AWS - below model has both training and test accuracy ~90% below the human-level accuracy => Algorithm did not have enough data to learn from

- **Increase the amount of training data** examples.
- **Increase the number of passes** on the existing training data.

Correct answers are: A and C for below problem



A Machine Learning Specialist is developing a model that classifies defective parts from a manufacturing process into one of eight defect types. The training data consists of 100,000 images per defect type. During the initial training of the image classification model, the Specialist notices that the validation accuracy is 89.5%, while the training accuracy is 90%. It is known that human-level performance for this type of image classification is around 97%.

What should the Specialist consider to improve the performance of the model? (Select TWO.)

☑A. A longer training time

☑B. Data augmentation

□C. Getting more training data

□D. A different optimizer

□E. L2 regularization

Data Augmentation could potentially work too but given the # of images 100,000

CSV -> Parquet/ORC

While you can use Firehose, **Firehose** is better suited at transforming **JSON to Parquet**.
To convert **CSV to Parquet**, best is to stream that data to S3 and then use AWS **Glue**
https://medium.com/searce/convert-csv-json-files-to-apache-parquet-using-aws-glue-a760d177b45f

Data Visualization

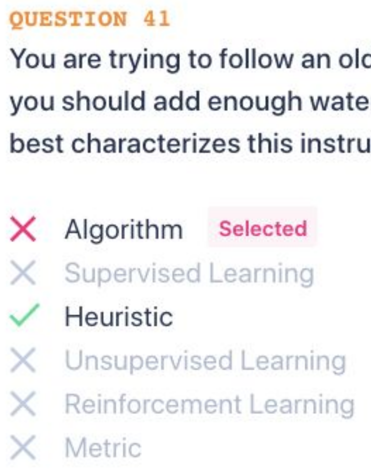
- Relationships
    - Scatter plots (2 values) - can see potential positive/negative correlation or none
    - Bubble plots (3 values)
- Comparisons
    - Bar chart (single value)
    - Line chart (one or more values)
- Distributions
    - Histograms (put values into bins)
    - Box plot (median value, 25/75 quartile, min and max, outliers, 50% majority values…)
    - Scatter Plot
- Compositions
    - Pie chart
    - Stacked Area chart
    - Stacked column chart

Heuristic vs model

- **Algorithm**: set of steps to follow to solve a specific problem intended to be **repeatable** with the same outcome
- **Heuristic**: a mental short-cut, an educated guess or “rule of thumb” that provides guidance on doing a task but **does NOT guarantee a consistent outcome**
    - well-known formula like predicting rate of deceleration of a car when brakes are applied

An Algorithm is a clearly defined set of instructions to solve a problem, Heuristics involve utilising an approach of learning and discovery to reach a solution.
So, if you know how to solve a problem then use an algorithm. If you need to develop a solution then it's heuristics.

A heuristic technique, often called simply a heuristic, is any approach to problem solving or self-discovery that employs a practical method, not guaranteed to be optimal, perfect, or rational, but instead sufficient for reaching an immediate goal. Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a satisfactory solution. A heuristic is an educated guess or intuition that usually does not ensure an optimal outcome. By asking the chef to evaluate the quantity of water needed based on individual perception best embodies a heuristic



QUESTION 41

You are trying to follow an old family recipe for making sourdough bread. The recipe states that you should add enough water to the flour such that it is slightly sticky. Which if the following best characterizes this instruction?

✗ Algorithm Selected
✗ Supervised Learning
✓ Heuristic
✗ Unsupervised Learning
✗ Reinforcement Learning
✗ Metric

Kinesis Data Streams and Kinesis Data Analytics cannot write data directly to S3. Kinesis Data **Firehose** is used as the main delivery mechanism for outputting data into S3. You can also use Lambda to write data into S3.

Kinesis Analytics destinations:

- Kinesis Data Stream
- Firehose
- Lambda.

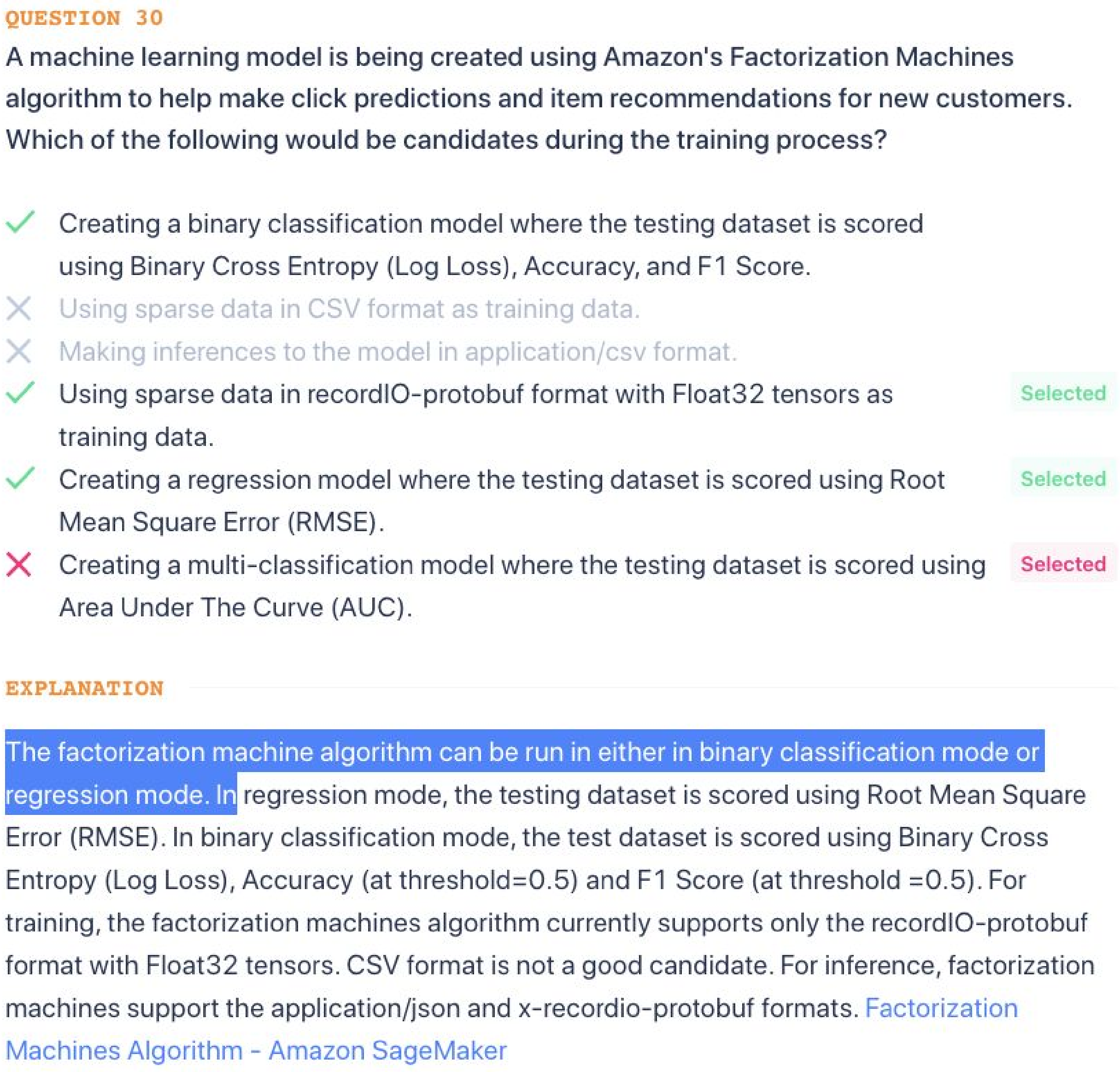
Amazon Machine Learning: no longer available for new customer. Use SageMake instead.

QUESTION 14

You work for a manufacturing company who has hundreds of conveyor belts with built-in IoT sensors. These sensors stream data into AWS using Kinesis Data Streams. The features associated with the data is belt_id, building_number, belt_temp, outside_temp, and power_consumption. During the processing of the data, you need to transform the data and store it in a data store. Which combination of services can you use to achieve this?

- ✗ Use Kinesis Data Streams to immediately write the data into S3. Next, setup a Lambda function that fires anytime an object is PUT onto S3. Transform the data from the Lambda function, then write the transformed data into S3 — Selected
- ✓ Immediately send the data to Lambda from Kinesis Data Streams. Transform the data in Lambda and write the transformed data into S3
- ✓ Setup Kinesis Data Analytics to ingest the data from Kinesis Data Stream, then run real-time SQL queries on the data to transform it. After the data is transformed, ingest the data with Kinesis Data Firehose and write the data into S3 — Selected
- ✗ Use Kinesis Data Analytics to run real-time SQL queries to transform the data and immediately write the transformed data into S3.
- ✓ Setup Kinesis Firehose to ingest data from Kinesis Data Streams, then send data to Lambda. Transform the data in Lambda and write the transformed data into S3 — Selected

Factorization machine can be run as binary classifier or regression - not multi-class



QUESTION 30

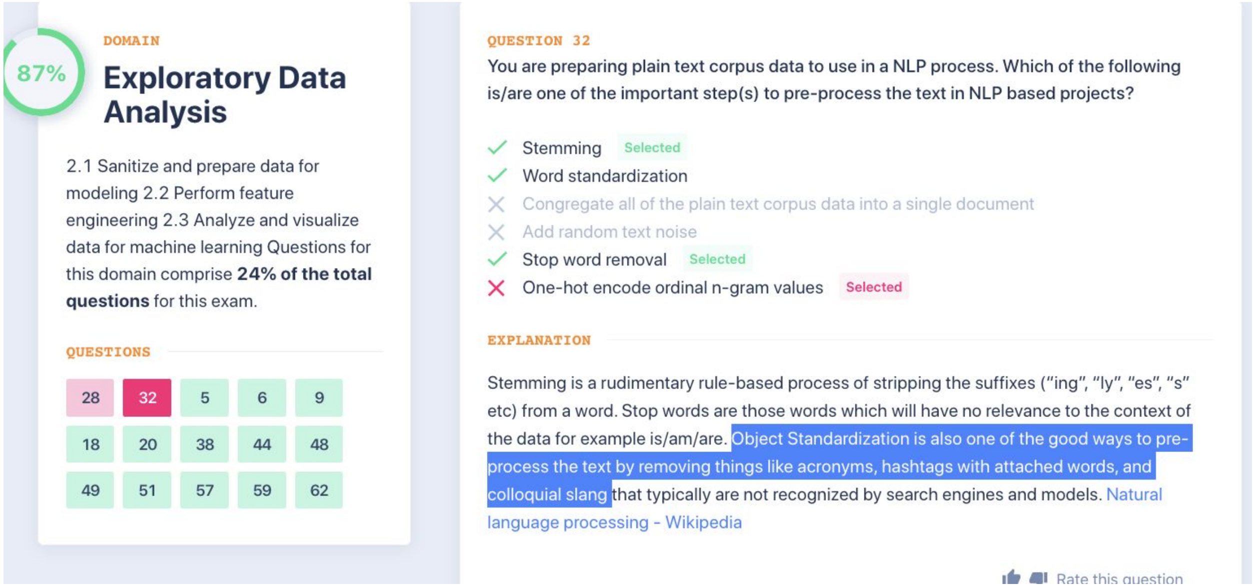
A machine learning model is being created using Amazon's Factorization Machines algorithm to help make click predictions and item recommendations for new customers. Which of the following would be candidates during the training process?

- ✓ Creating a binary classification model where the testing dataset is scored using Binary Cross Entropy (Log Loss), Accuracy, and F1 Score.
- ✗ Using sparse data in CSV format as training data.
- ✗ Making inferences to the model in application/csv format.
- ✓ Using sparse data in recordIO-protobuf format with Float32 tensors as training data. Selected
- ✓ Creating a regression model where the testing dataset is scored using Root Mean Square Error (RMSE). Selected
- ✗ Creating a multi-classification model where the testing dataset is scored using Area Under The Curve (AUC). Selected

EXPLANATION

The factorization machine algorithm can be run in either in binary classification mode or regression mode. In regression mode, the testing dataset is scored using Root Mean Square Error (RMSE). In binary classification mode, the test dataset is scored using Binary Cross Entropy (Log Loss), Accuracy (at threshold=0.5) and F1 Score (at threshold =0.5). For training, the factorization machines algorithm currently supports only the recordIO-protobuf format with Float32 tensors. CSV format is not a good candidate. For inference, factorization machines support the application/json and x-recordio-protobuf formats. Factorization Machines Algorithm - Amazon SageMaker

NLP Tasks



87%

DOMAIN

Exploratory Data Analysis

2.1 Sanitize and prepare data for modeling 2.2 Perform feature engineering 2.3 Analyze and visualize data for machine learning Questions for this domain comprise **24% of the total questions** for this exam.

QUESTIONS

28	32	5	6	9
18	20	38	44	48
49	51	57	59	62

QUESTION 32

You are preparing plain text corpus data to use in a NLP process. Which of the following is/are one of the important step(s) to pre-process the text in NLP based projects?

- ✓ Stemming Selected
- ✓ Word standardization
- ✗ Congregate all of the plain text corpus data into a single document
- ✗ Add random text noise
- ✓ Stop word removal Selected
- ✗ One-hot encode ordinal n-gram values Selected

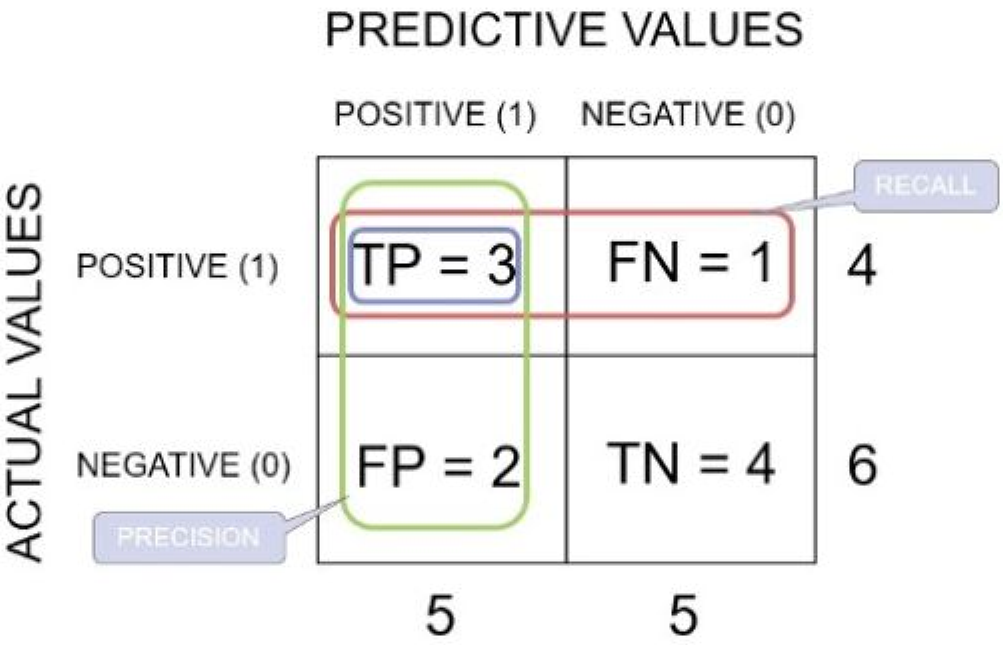
EXPLANATION

Stemming is a rudimentary rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word. Stop words are those words which will have no relevance to the context of the data for example is/am/are. Object Standardization is also one of the good ways to pre-process the text by removing things like acronyms, hashtags with attached words, and colloquial slang that typically are not recognized by search engines and models. Natural language processing - Wikipedia

Stemming is a rudimentary rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word.

Stop words are those words which will have no relevance to the context of the data for example is/am/are.

Object Standardization is also one of the good ways to pre-process the text by removing things like acronyms, hashtags with attached words, and colloquial slang that typically are not recognized by search engines and models. Natural language processing - Wikipedia



		PREDICTIVE VALUES		
		POSITIVE (1)	NEGATIVE (0)	
ACTUAL VALUES	POSITIVE (1)	TP = 3	FN = 1	4
	NEGATIVE (0)	FP = 2	TN = 4	6
		5	5	

Recall (type II): Recall gives us an idea about when it's actually yes, how often does it predict yes.
Recall = TP / (TP + FN) = 3/(3+1) = 0.75

Precision (type I): Precision tells us about when it predicts yes, how often is it correct.
Precision = TP / (TP + FP) = 3/(3+2) = 0.60

F-score:
F-score = (2*Recall*Precision)/(Recall+Precision) = (2*0.75*0.60)/(0.75+0.60) = 0.67

Specificity:
Specificity = TN / (TN + FP) = 4/(4+2) = 0.67

Hyper parameters vs parameters

- Hyper parameter: values set BEFORE the learning process
- Parameters: values DERIVED VIA the learning process

Data for SageMaker

- Most algorithms accept CSV
    - Content-type = "**text/csv**"
    - If unsupervised: label_size=0
    - If **supervised**: **label_size=1** (and label data in 1st column)
    - **No header**
    - Use optimized protobuf recordIO for optimal performance (with Pipe mode)
        - Data can be streamed from S3 to the learning instance, requiring less EBS space and faster startup

CreateTraingJob API for using Training images to train an algo (K-means, BlazingText…)

- **AlgorithmSpecification** registry path of the Docker image that contains the training algorithm and algorithm-specific metadata, including the input mode
- **OutputDataConfig** Specifies the path to the S3 location where you want to store model artifacts
- **ResourceConfig** The resources, including the ML compute instances and ML storage volumes, to use for model training.
- **RoleArn** The Amazon Resource Name (ARN) of an IAM role that Amazon SageMaker can assume to perform tasks on your behalf.
- **TrainingJobName** The name of the training job. The name must be unique within an AWS Region in an AWS account.
- **StoppingCondition** Specifies a limit to how long a model training job can run.
- Optionally:
    - **InputDataConfig** An array of `Channel` objects. Each channel is a named input source. `InputDataConfig` describes the input data and its location.
    - **Tags**
    - **HyperParameters**
    - **EnableNetworkIsolation** Isolates the training container. No inbound or outbound network calls can be made, except for calls between peers within a training cluster for distributed training.
    - **EnableManagedSpotTraining** To train models using managed spot training, choose `True`.
    - **EnableInterContainerTrafficEncryption** To encrypt all communications between ML compute instances in distributed training, choose `True`. Encryption provides greater security for distributed training, but training might take longer.
    - **CheckpointConfig** Contains information about the output location for managed spot training checkpoint data.

CreateModel

- **ExecutionRoleArn**
- **ModelName**
- Optionally:
    - **PrimaryContainer** The location of the primary docker image containing inference code, associated artifacts, and custom environment map that the inference code uses when the model is deployed for predictions.
    - **Tags**
    - **VpcConfig**

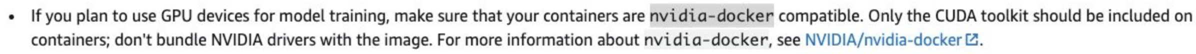
- Containers

:1 vs :latest container tags

:1 for using a stable version of the algorithm
:latest is the most recent but could cause backward compatible issue. Avoid using it for production ise

Leveraging NVidia GPUs for custom containers

https://docs.aws.amazon.com/sagemaker/latest/dg/your-algorithms-training-algo-dockerfile.html



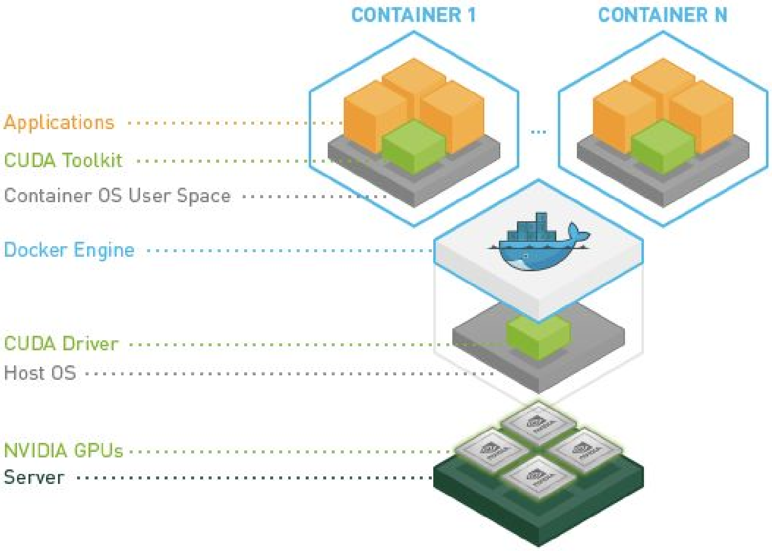
- If you plan to use GPU devices for model training, make sure that your containers are `nvidia-docker` compatible. Only the CUDA toolkit should be included on containers; don't bundle NVIDIA drivers with the image. For more information about `nvidia-docker`, see NVIDIA/nvidia-docker.

https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-dg.pdf (p. 55)

If you plan to use GPU devices, make sure that your containers are **nvidia-docker** compatible. Only the CUDA toolkit should be included on containers. Don't bundle NVIDIA drivers with the image.
For more information about nvidia-docker, see https://github.com/NVIDIA/nvidia-docker

NVIDIA Container Toolkit

license Apache-2.0 documentation wiki packages repository



Measuring Binary Classifier

https://docs.aws.amazon.com/machine-learning/latest/dg/binary-model-insights.html

=> Use AUC

Measuring ML Model Accuracy

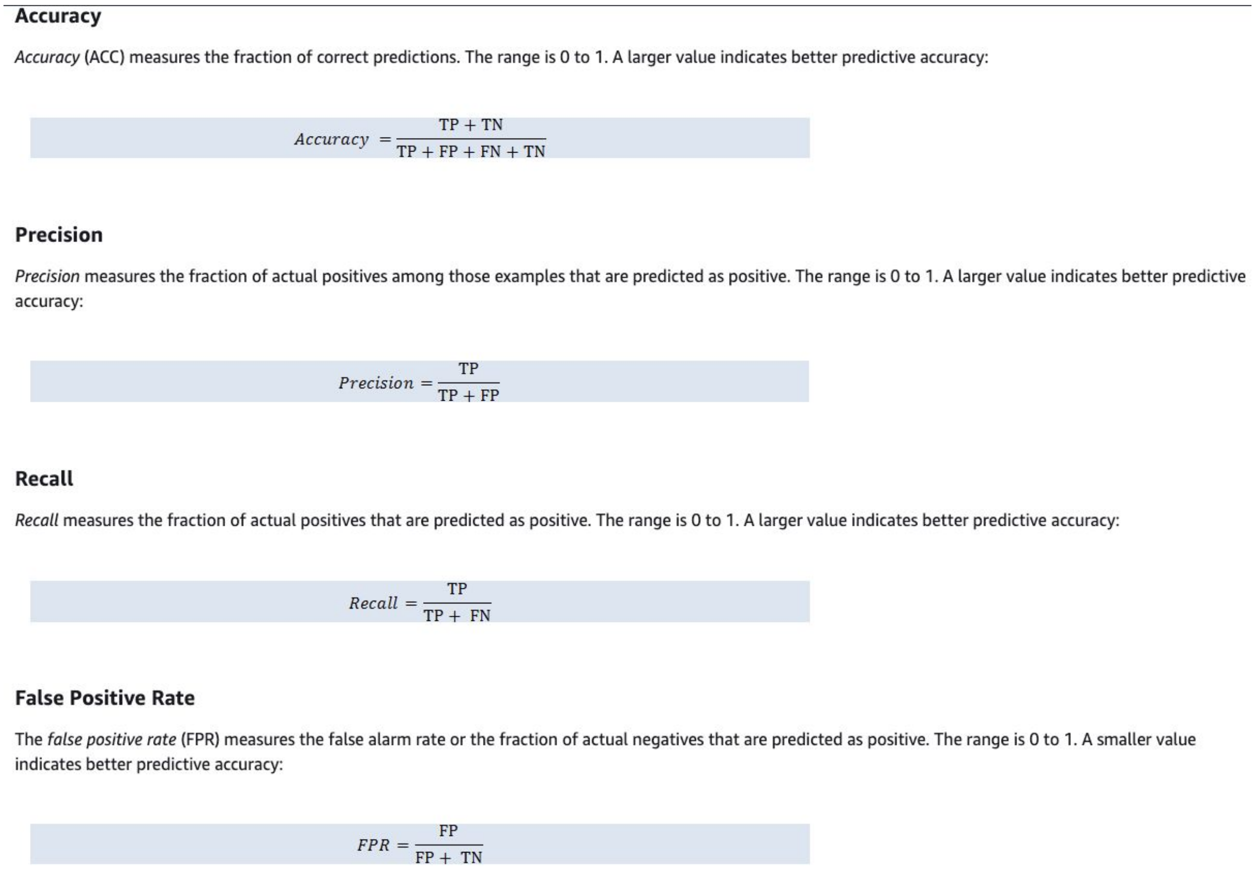
Amazon ML provides an industry-standard accuracy metric for binary classification models called Area Under the (Receiver Operating Characteristic) Curve (AUC). AUC measures the **ability of the model to predict a higher score for positive examples as compared to negative examples.** Because it is independent of the score cut-off, you can get a sense of the prediction accuracy of your model from the AUC metric without picking a threshold.

The AUC metric returns a decimal value from 0 to 1.

- AUC values near **1** indicate an ML model that is **highly accurate**.
- Values near **0.5** indicate an ML model that is **no better than guessing at random**.
- Values near **0** are unusual to see, and typically indicate a **problem with the data.**

Essentially, an AUC near 0 says that the ML model has learned the correct patterns, but is using them to make predictions that are flipped from reality ('0's are predicted as '1's and vice versa). For more information about AUC, go to the Receiver operating characteristic page on Wikipedia.

The **baseline** AUC metric for a binary model is **0.5**. It is the value for a hypothetical ML model that randomly predicts a 1 or 0 answer. **Your binary ML model should perform better than this value to begin to be valuable.**



Accuracy

Accuracy (ACC) measures the fraction of correct predictions. The range is 0 to 1. A larger value indicates better predictive accuracy:

$$Accuracy\ = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

Precision

Precision measures the fraction of actual positives among those examples that are predicted as positive. The range is 0 to 1. A larger value indicates better predictive accuracy:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Recall

Recall measures the fraction of actual positives that are predicted as positive. The range is 0 to 1. A larger value indicates better predictive accuracy:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \ \mathrm{FN}}$$

False Positive Rate

The *false positive rate* (FPR) measures the false alarm rate or the fraction of actual negatives that are predicted as positive. The range is 0 to 1. A smaller value indicates better predictive accuracy:

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \ \mathrm{TN}}$$

Residual analysis for Regression models

Residual = actual - predicted

https://towardsdatascience.com/how-to-use-residual-plots-for-regression-model-validation-c3c70e8ab378

A residual is a measure of how far away a point is vertically from the regression line. Simply, it is the error between a predicted value and the observed actual value.

Residual Plots

A typical residual plot has the residual values on the Y-axis and the independent variable on the x-axis. Figure 2 below is a good exampl how a typical residual plot looks like.

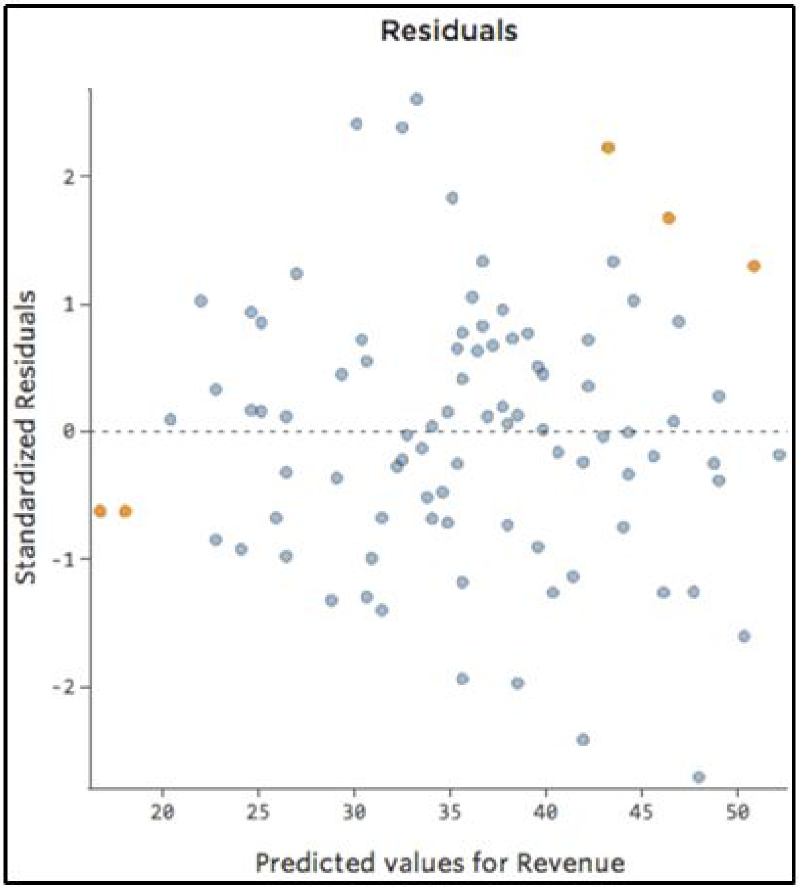


Fig. 2 [Credit]

Characteristics of Good Residual Plots

A few characteristics of a good residual plot are as follows:

1. It has a high density of points close to the origin and a low density of points away from the origin
2. It is symmetric about the origin

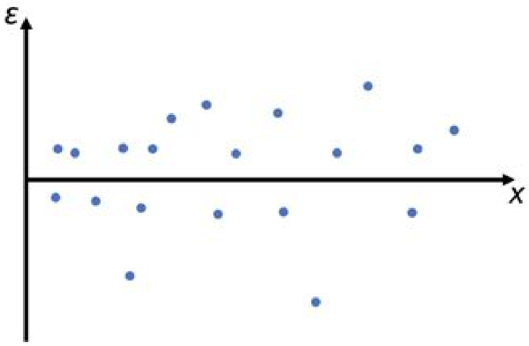


Fig. 3: Good Residual Plot

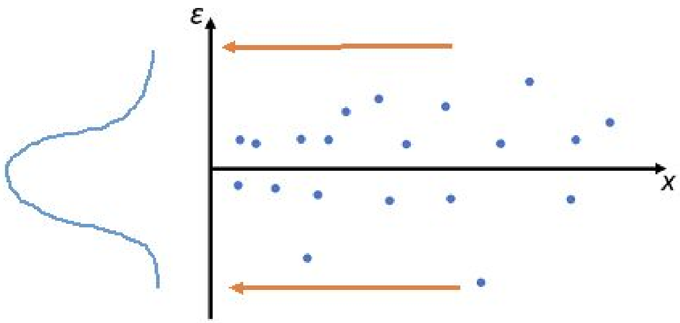


Fig. 3b: Project onto the y-axis

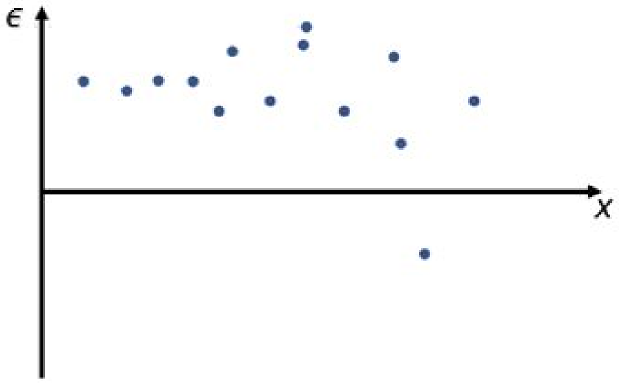


Fig. 4: Example of Bad Residual plot

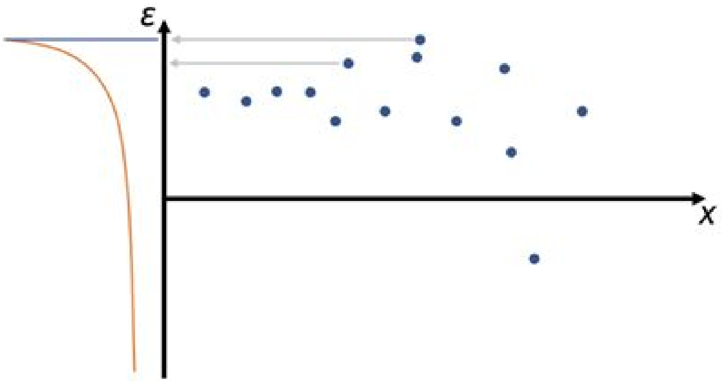
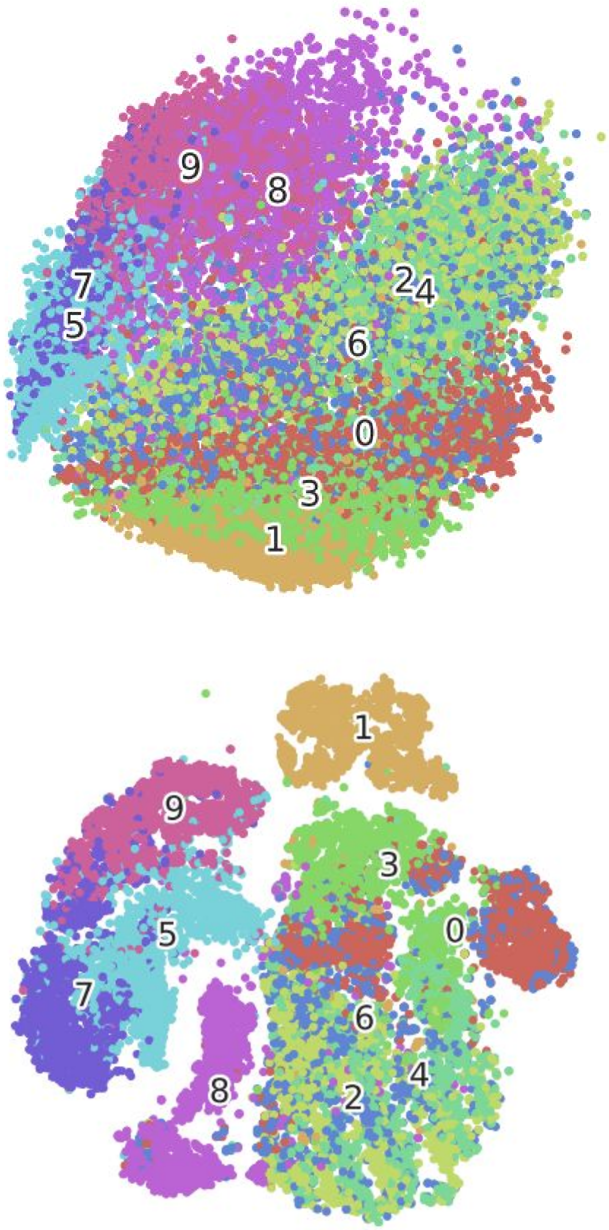


Fig. 4b: Project onto the y-axis

Using t-SNE

https://www.datacamp.com/community/tutorials/introduction-t-sne

t-Distributed Stochastic Neighbor Embedding (t-SNE)


9
8
7
5
2
4
6
0
3
1
1
9
3
5
0
7
6
4
8
2

Overfitting in image classification
=> Regularization
=> Data Augmentation
https://heartbeat.fritz.ai/overcoming-overfitting-in-image-classification-using-data-augmentation-9858c5cee986

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of **data** available for training models, without actually collecting new **data**. We can say data augmentation is a strategy focused on regeneration of more images from already-available ones.
This will help with model bias towards a particular class of dta. **Helps algorithm generalize well**

Optimizing Queries for Athena
Use **Parquet** (not recordIO)

CloudTrail integration with SM
https://docs.aws.amazon.com/sagemaker/latest/dg/logging-using-cloudtrail.html
Amazon SageMaker is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in SageMaker. **CloudTrail captures all API calls for SageMaker**, with the exception of InvokeEndpoint, as events.
The calls captured include calls from the SageMaker console and code calls to the SageMaker API operations.

- If you **create a trail**, you can enable continuous delivery of CloudTrail events to an Amazon **S3** bucket, including events for SageMaker.
- If you don't configure a trail, you can still view the **most recent** events in the CloudTrail console in **Event history**.

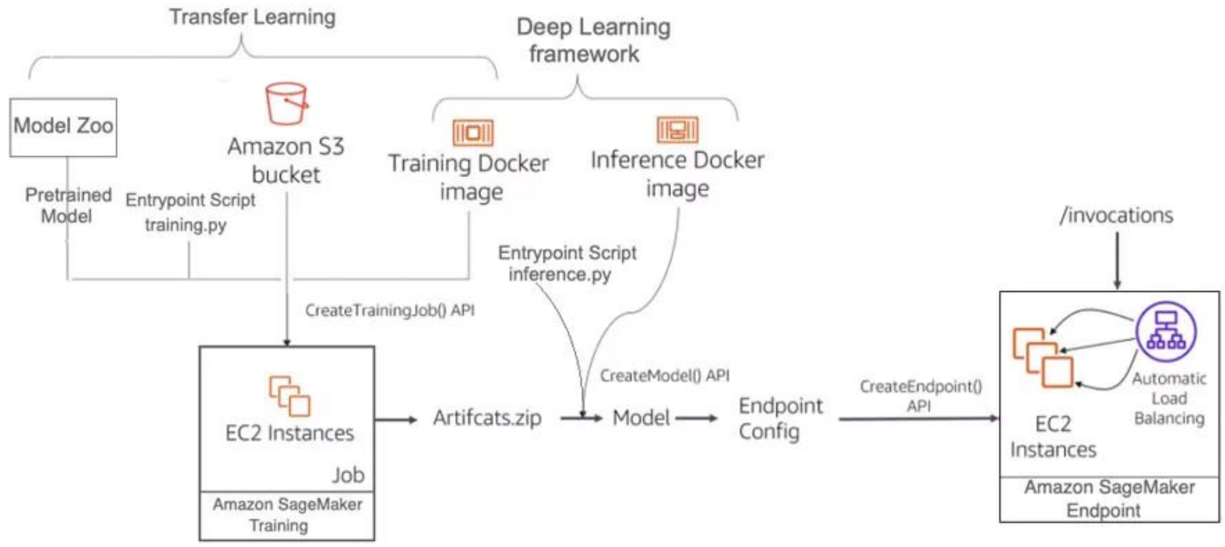
90 days - This is the **default** trail. Information in this trail is kept for the last 90 days in a rolling fashion

Using the information collected by CloudTrail, you can determine the **request that was made to SageMaker, the IP address from which the request was made, who made the request, when it was made, and additional details**.

All SageMaker actions, with the exception of InvokeEndpoint, are logged by CloudTrail and are documented in the Operations. For example, calls to the **CreateTrainingJob**, **CreateEndpoint** and **CreateNotebookInstance** actions generate entries in the CloudTrail log files.

SNS: You can be notified when CloudTrail publishes new log files to your Amazon S3 bucket. You manage notifications using Amazon Simple Notification Service (Amazon SNS).

ML Lifecycle

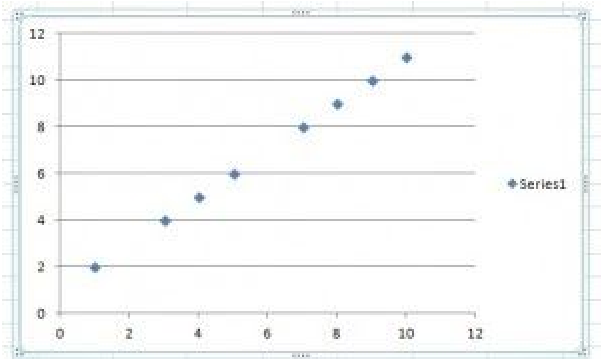


CreateTrainingJob() => CreateModel() => CreateEndpointConfig() => CreateEndpoint()

Discrete Vs Continuous Variable

What is a Discrete Variable?

Discrete variables are **countable** in a finite amount of time. For example, you can count the change in your pocket. You can count the money in your bank account. You could also count the amount of money in *everyone's* bank accounts. It might take you a long time to count that last item, but the point is—it's still countable.



Discrete variables on a scatter plot.

What is a Continuous Variable?

Continuous Variables would (literally) take **forever** to count. In fact, you would get to "forever" and never finish counting them. For example, take age. You can't count "age". **Why not?** Because it would literally take forever. For example, you could be:

25 years, 10 months, 2 days, 5 hours, 4 seconds, 4 milliseconds, 8 nanoseconds, 99 picosends…and so on.

Time is a continuous variable

Discrete Distribution

What Is Discrete Distribution?

A discrete distribution is a statistical distribution that shows the **probabilities of discrete (countable)** outcomes, such as 1, 2, 3...

Statistical distributions can be either discrete or continuous.

Discrete Probability Distributions

If a random variable is a discrete variable, its probability distribution is called a **discrete probability distribution**.

- Binomial probability distribution
- Hypergeometric probability distribution
- Multinomial probability distribution
- Negative binomial distribution
- Poisson probability distribution

Continuous Probability Distributions

If a random variable is a continuous variable, its probability distribution is called a **continuous probability distribution**.

A continuous probability distribution differs from a discrete probability distribution in several ways.

- The probability that a continuous random variable will assume a particular value is zero.
- As a result, a continuous probability distribution cannot be expressed in tabular form.
- Instead, an equation or formula is used to describe a continuous probability distribution.

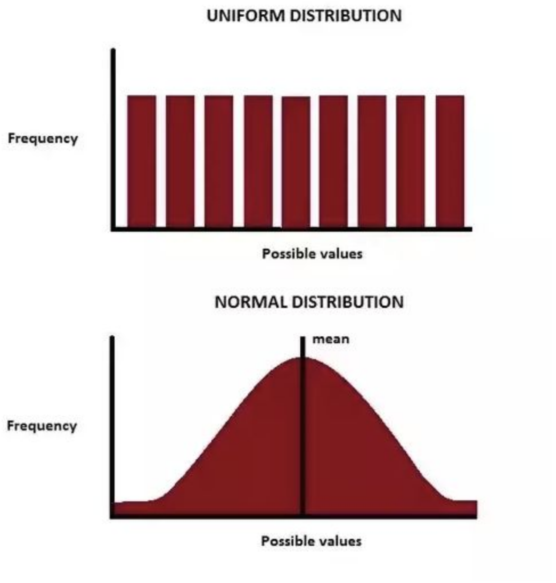
continuous probability distributions.

- Normal probability distribution
- Student's t distribution
- Chi-square distribution
- F distribution

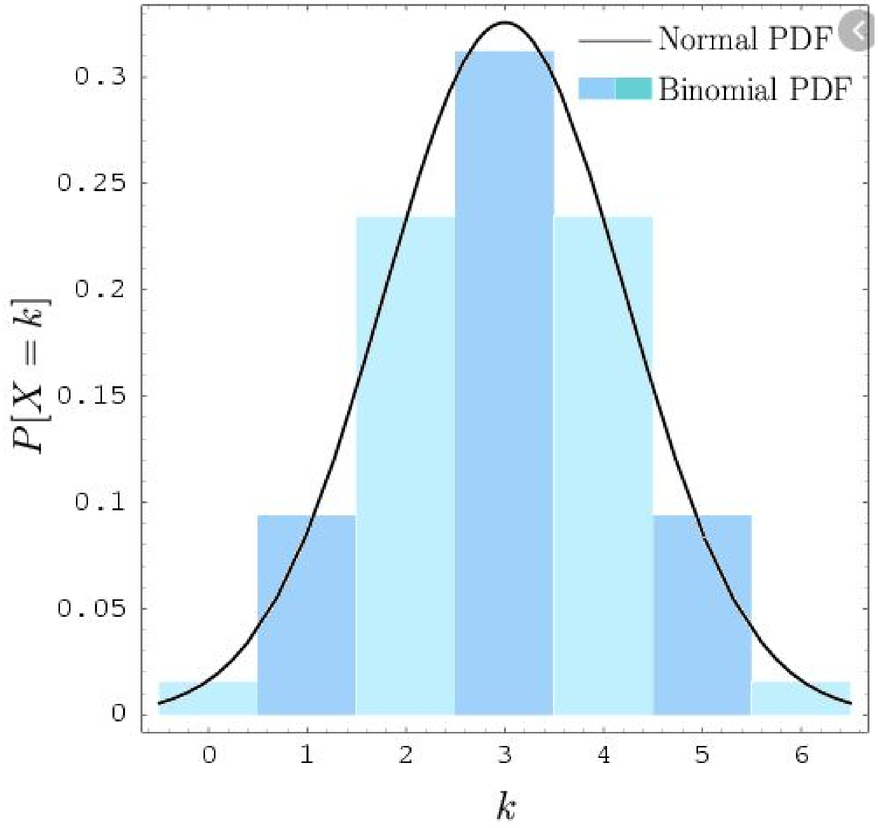
Difference Uniform distribution vs normal distribution

A **uniform distribution** is one in which all values are **equally** likely **within a range (and impossible beyond that range)**. For example, in a uniform distribution from 0 to 10, values from 0 to 1 have a 10% probability as do values from 5 to 6.

A **Normal distribution** is one in which **values** cluster **around the mean**, or average, a **outlying values are very unlikely** (unlike what I just described where outlying values are equally likely as values in the middle). While values far away from the mean are unlikely, the Normal Distribution’s **range is from minus infinity to infinity**, meaning that values in any range have some chance of occurring (even if it is infinitesimally small).



Difference between binomial distribution and normal distribution



Key Differences Between Binomial and Poisson Distribution

The differences between binomial and poisson distribution can be drawn clearly on the following grounds:

1. The binomial distribution is one in which the probability of repeated number of trials is studied. A probability distribution that gives the count of a number of independent events occur randomly within a given period, is called probability distribution.
2. Binomial Distribution is biparametric, i.e. it is featured by two parameters n and p whereas Poisson distribution is uniparametric, i.e. characterised by a single parameter m.
3. There are a **fixed number of attempts in the binomial distribution**. On the other hand, an **unlimited number of trials are there in a poisson distribution.**
4. The **success probability is constant in binomial distribution** but in poisson distribution, there are an extremely small number of success chances.

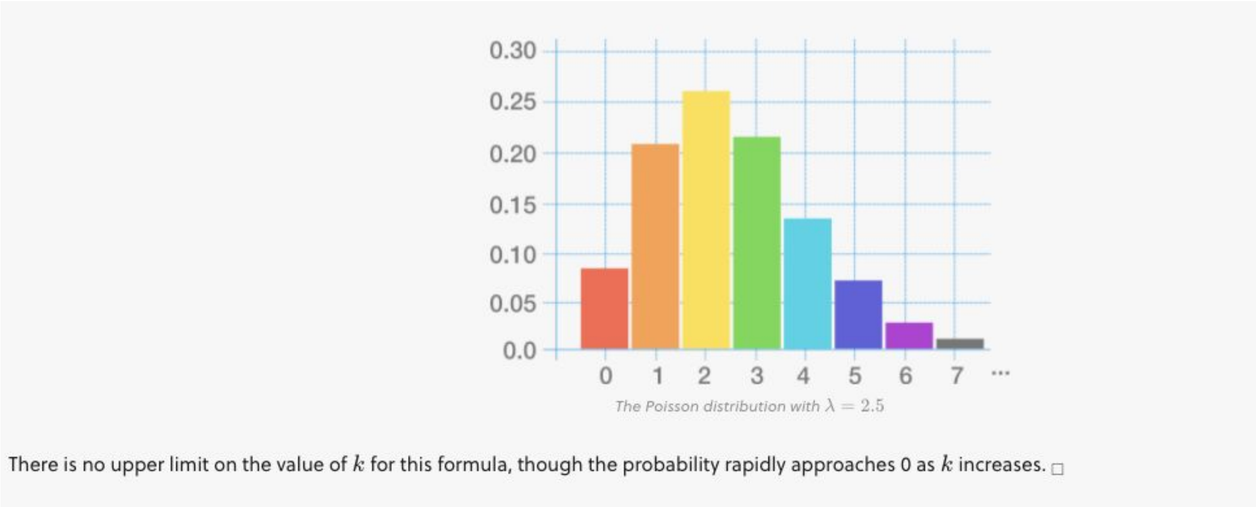
5. In a binomial distribution, there are only two possible outcomes, i.e. success or failure. Conversely, there are an unlimited number of possible outcomes in the case of poisson distribution.
6. In binomial distribution Mean > Variance while in poisson distribution mean = variance.

https://brilliant.org/wiki/poisson-distribution/
The **Poisson distribution** is the discrete probability distribution of the number of events occurring in a given time period, **given the average number of times the event occurs over that time period.**

A certain fast-food restaurant gets an average of 3 visitors to the drive-through per minute. This is just an average, however. The actual amount can vary.

A Poisson distribution can be used to analyze the probability of various events regarding how many customers go through the drive-through. It can allow one to calculate the probability of a lull in activity (when there are 0 customers coming to the drive-through) as well as the probability of a flurry of activity (when there are 5 or more customers coming to the drive-through). This information can, in turn, help a manager plan for these events with staffing and scheduling.

In the World Cup, an average of 2.5 goals are scored each game. Modeling this situation with a Poisson distribution, what is the probability that k goals are scored in a game?



The Poisson distribution with $\lambda = 2.5$

There is no upper limit on the value of k for this formula, though the probability rapidly approaches 0 as k increases. □

Security - Notebooks to secure with no internet traffic
"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"

=> 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.

Note: NAT gateway COULD GO OUT TO THE INTERNET AND DOWNLOAD BACK MALICIOUS
D
C is correct
We must use the VPC endpoint (either Gateway Endpoint or Interface Endpoint)to comply with this requirement "Data communication traffic must stay within the AWS network".
https://docs.aws.amazon.com/sagemaker/latest/dg/notebook-interface-endpoint.html

You can connect to your notebook instance from your VPC through an **interface endpoint** in your Virtual Private Cloud (VPC) instead of connecting over the internet. When you use a VPC interface endpoint, communication between your VPC and the notebook instance is conducted entirely and securely within the AWS network.

https://docs.aws.amazon.com/sagemaker/latest/dg/host-vpc.html

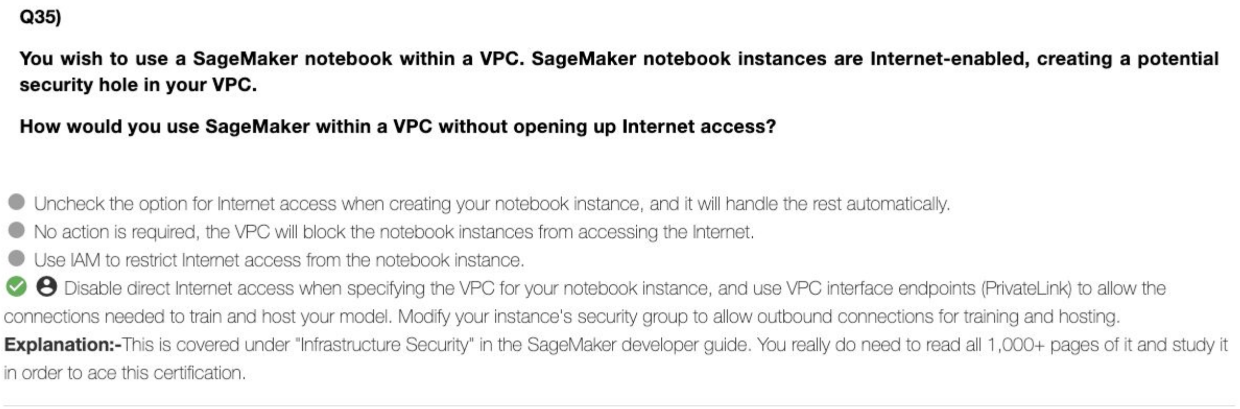
If you configure your VPC so that it doesn't have internet access, models that use that VPC do not have access to resources outside your VPC. If your model needs access to resources outside your VPC, provide access with one of the following options:

- If your model needs **access to an AWS service that supports interface VPC endpoints**, create an **endpoint** to connect to that service. For a list of services that support interface endpoints, see VPC Endpoints in the Amazon VPC User Guide. For information about creating an interface VPC endpoint, see Interface VPC Endpoints (AWS PrivateLink) in the Amazon VPC User Guide.

- If your model needs access to an AWS service that doesn't support interface VPC endpoints or to a **resource outside of AWS**, create a **NAT gateway** and configure your security groups to allow outbound connections. For information about setting up a NAT gateway for your VPC, see Scenario 2: VPC with Public and Private Subnets (NAT) in the Amazon Virtual Private Cloud User Guide.

Security - use SM without optming up Internet

- Disable internet access when specifying VPC for your network
- Use VPC interface endpoint (Private Link) to allow connections needed to train and host yourmodel
- Modify your instance's security group to allow outbound connections training and hosting.
-

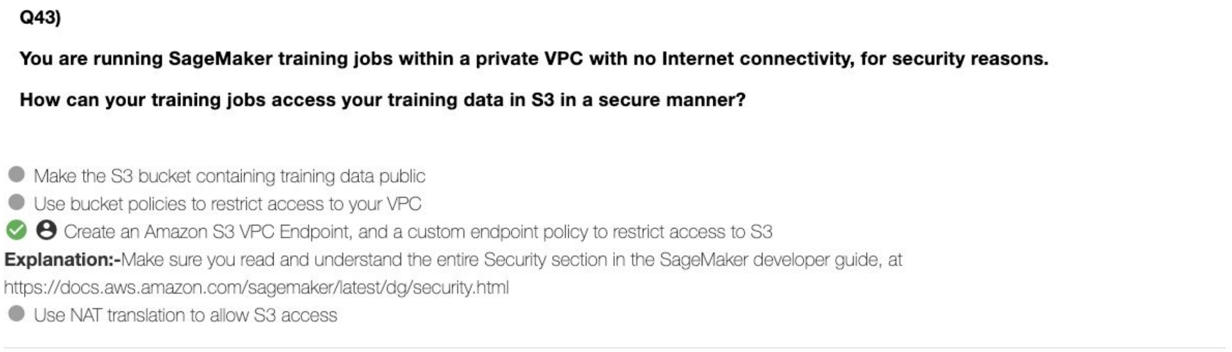


Q35)

You wish to use a SageMaker notebook within a VPC. SageMaker notebook instances are Internet-enabled, creating a potential security hole in your VPC.

How would you use SageMaker within a VPC without opening up Internet access?

● Uncheck the option for Internet access when creating your notebook instance, and it will handle the rest automatically.
● No action is required, the VPC will block the notebook instances from accessing the Internet.
● Use IAM to restrict Internet access from the notebook instance.
✓ Disable direct Internet access when specifying the VPC for your notebook instance, and use VPC interface endpoints (PrivateLink) to allow the connections needed to train and host your model. Modify your instance's security group to allow outbound connections for training and hosting.
Explanation:-This is covered under "Infrastructure Security" in the SageMaker developer guide. You really do need to read all 1,000+ pages of it and study it in order to ace this certification.



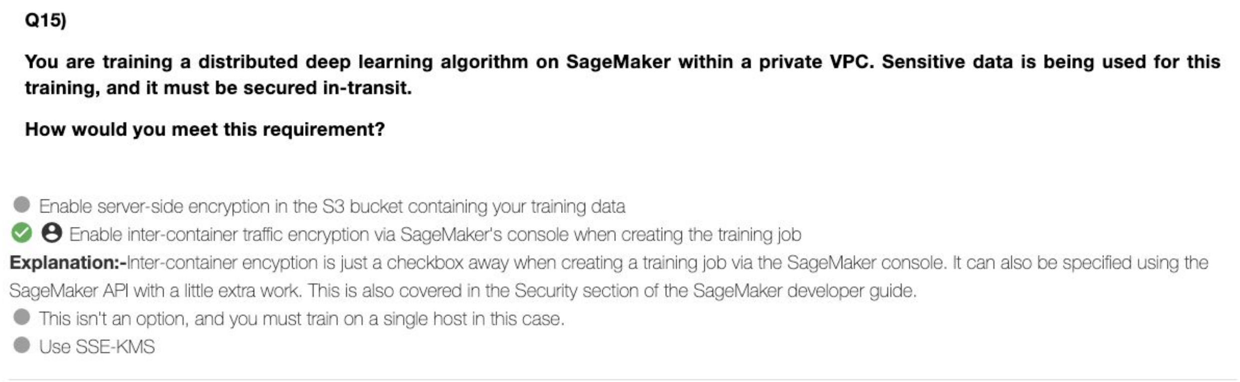
Q43)

You are running SageMaker training jobs within a private VPC with no Internet connectivity, for security reasons.

How can your training jobs access your training data in S3 in a secure manner?

● Make the S3 bucket containing training data public
● Use bucket policies to restrict access to your VPC
✓ Create an Amazon S3 VPC Endpoint, and a custom endpoint policy to restrict access to S3
Explanation:-Make sure you read and understand the entire Security section in the SageMaker developer guide, at https://docs.aws.amazon.com/sagemaker/latest/dg/security.html
● Use NAT translation to allow S3 access

Security - protecting data in-transit
=> Inter-container encryption is just a checkbox away when creating a training job via SM console. It can also be specified via SM API.



Q15)

You are training a distributed deep learning algorithm on SageMaker within a private VPC. Sensitive data is being used for this training, and it must be secured in-transit.

How would you meet this requirement?

● Enable server-side encryption in the S3 bucket containing your training data

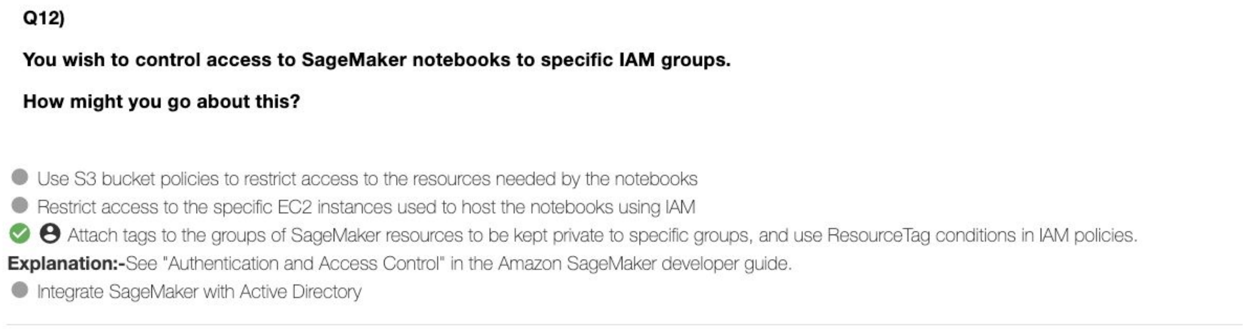
✓ Enable inter-container traffic encryption via SageMaker's console when creating the training job

Explanation:-Inter-container encyption is just a checkbox away when creating a training job via the SageMaker console. It can also be specified using the SageMaker API with a little extra work. This is also covered in the Security section of the SageMaker developer guide.

● This isn't an option, and you must train on a single host in this case.

● Use SSE-KMS

Security - control access to SageMaker notebooks to specific IAM groups
=> Attach **tags** to the groups of SM resources to be kept private to specific groups, and use **ResourceTag** conditions in IAM policies.



Q12)

You wish to control access to SageMaker notebooks to specific IAM groups.

How might you go about this?

● Use S3 bucket policies to restrict access to the resources needed by the notebooks

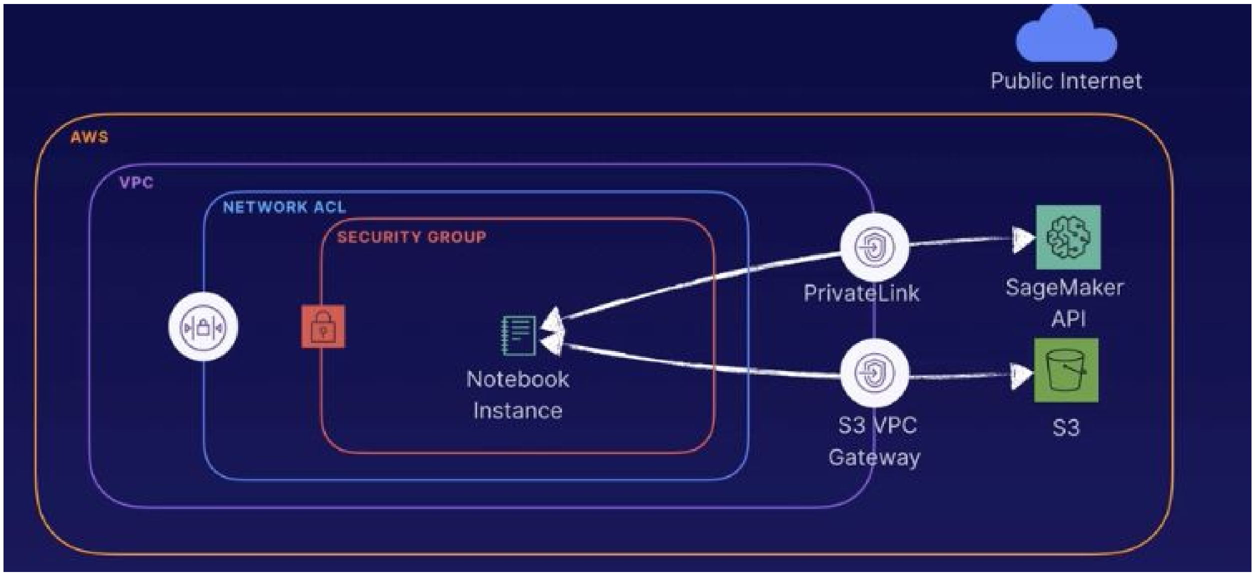
● Restrict access to the specific EC2 instances used to host the notebooks using IAM

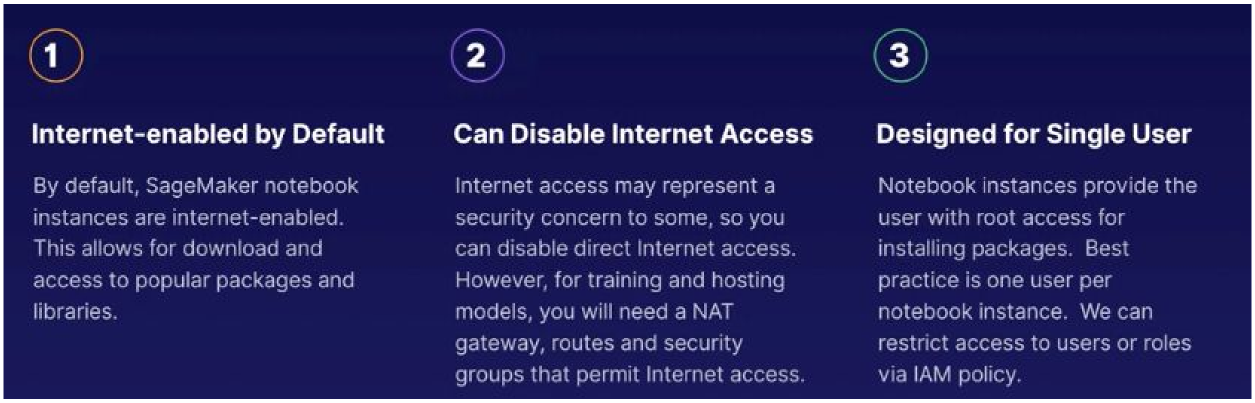
✓ Attach tags to the groups of SageMaker resources to be kept private to specific groups, and use ResourceTag conditions in IAM policies.

Explanation:-See "Authentication and Access Control" in the Amazon SageMaker developer guide.

● Integrate SageMaker with Active Directory

Securing ML resources, not going over Public internet





1

Internet-enabled by Default

By default, SageMaker notebook instances are internet-enabled. This allows for download and access to popular packages and libraries.

2

Can Disable Internet Access

Internet access may represent a security concern to some, so you can disable direct Internet access. However, for training and hosting models, you will need a NAT gateway, routes and security groups that permit Internet access.

3

Designed for Single User

Notebook instances provide the user with root access for installing packages. Best practice is one user per notebook instance. We can restrict access to users or roles via IAM policy.

L1 (Lasso) and L2 (Ridge)

The **key difference** between these techniques is that **L1 (Lasso)** shrinks the less important feature’s coefficient to zero thus, **removing** some feature altogether. So, this works well for **feature selection** in case we have a huge number of features.

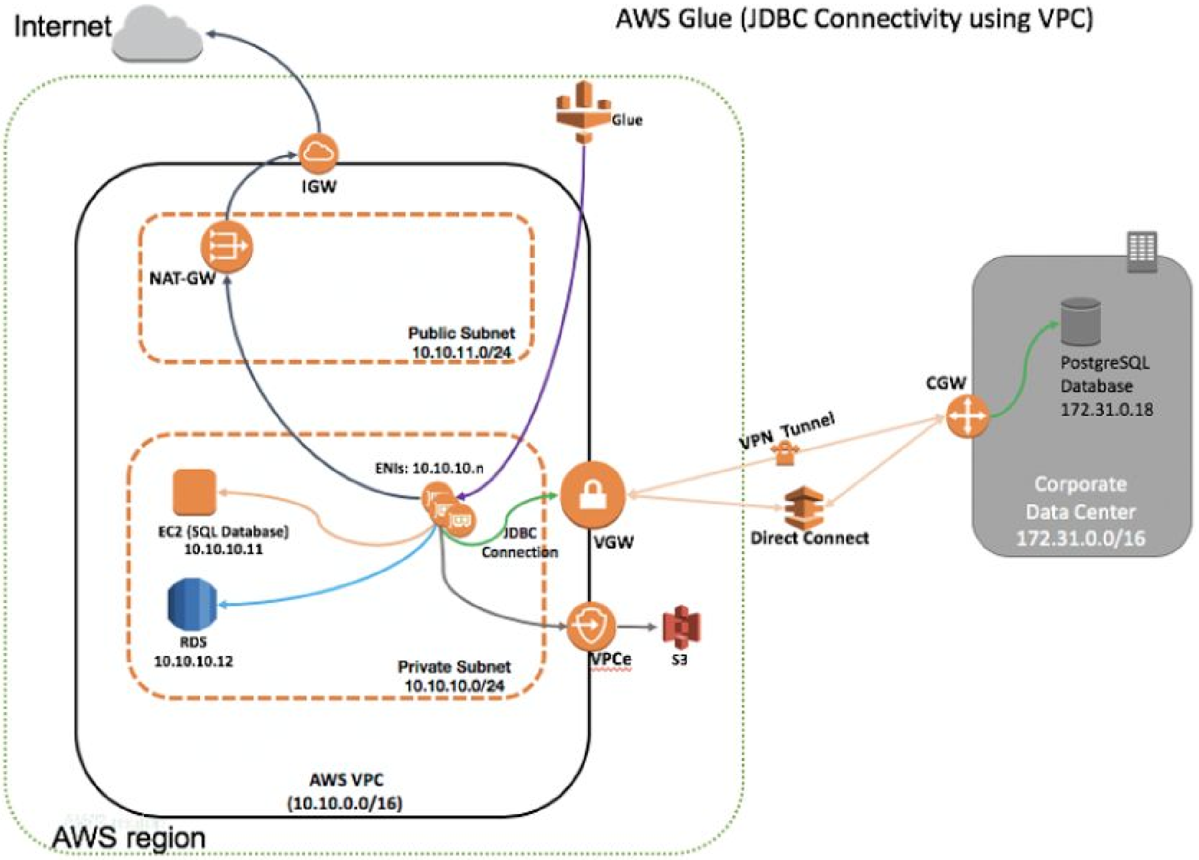
The main difference between these techniques is that L1 shrinks the less important feature’s coefficient down to zero. Causing it inexplicitly removed the feature, it can be used in feature analysis. This model works well if the dataset has super wide columns as the sparsity in the outputs. **Ridge regularisation**, on the other hand, **L2** tends to **keep the coefficients less important features tiny rather than zero them out.**

AWS Glue and on-prem access

https://aws.amazon.com/blogs/big-data/how-to-access-and-analyze-on-premises-data-stores-using-aws-glue/#:~:text=AWS%20Glue%20can%20also%20connect,Microsoft%20SQL%20Server%2C%20and%20MariaDB.&text=AWS%20Glue%20jobs%20extract%20data,data%20stores%20as%20a%20target.

AWS Glue can also connect to a variety of **on-premises JDBC** data stores such as PostgreSQL, MySQL, Oracle, Microsoft SQL Server, and MariaDB.

The following diagram shows the architecture of using AWS Glue in a hybrid environment, as described in this post. The solution uses JDBC connectivity using the elastic network interfaces (ENIs) in the Amazon VPC. The ENIs in the VPC help connect to the on-premises database server over a virtual private network (VPN) or AWS Direct Connect (DX).



Using Athena for ETL

https://aws.amazon.com/blogs/big-data/extract-transform-and-load-data-into-s3-data-lake-using-ctas-and-insert-into-statements-in-amazon-athena/

Amazon Athena is an interactive query service that makes it easy to analyze the data stored in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

This blog post discusses how to use Athena for extract, transform and load (ETL) jobs for data processing

Amazon Athena recently added support for federated queries and user-defined functions (UDFs), both in Preview.

Using Amazon Athena to simplify ETL workflows and enable quicker analytics

Athena, a fully managed serverless interactive service for querying data in Amazon S3 using SQL, has been rapidly adopted by multiple departments across our organization. For our use case, we did not require an always-on EMR cluster waiting for an analytics query. Athena's serverless nature is perfect for our use case. Along the way we discovered that we could use Athena to run extract, transform, and load (ETL) jobs.

However, Athena is a lot more than an interactive service for querying data in Amazon S3. We also found Athena to be a robust, powerful, reliable, scalable, and cost-effective ETL tool. The ability to schedule SQL statements, along with support for Create Table As Select (CTAS) and INSERT INTO statements, helped us accelerate our ETL workloads.

With the addition of query federation and UDFs to Athena, Jornaya has been able to replace many of our unstable data pipelines with Athena to extract and transform data from DynamoDB and write it to Amazon S3.

Cost - Inference is 90% of the cost of compute cost
=> use **Elastic Inference**
https://aws.amazon.com/blogs/big-data/simplify-etl-data-pipelines-using-amazon-athenas-federated-queries-and-user-defined-functions/

https://docs.aws.amazon.com/sagemaker/latest/dg/ei.html
By using Amazon Elastic Inference (EI), you can **speed up the throughput and decrease the latency** of getting real-time inferences from your deep learning models that are deployed as Amazon SageMaker hosted models, but **at a fraction of the cost of using a GPU** instance for your endpoint.

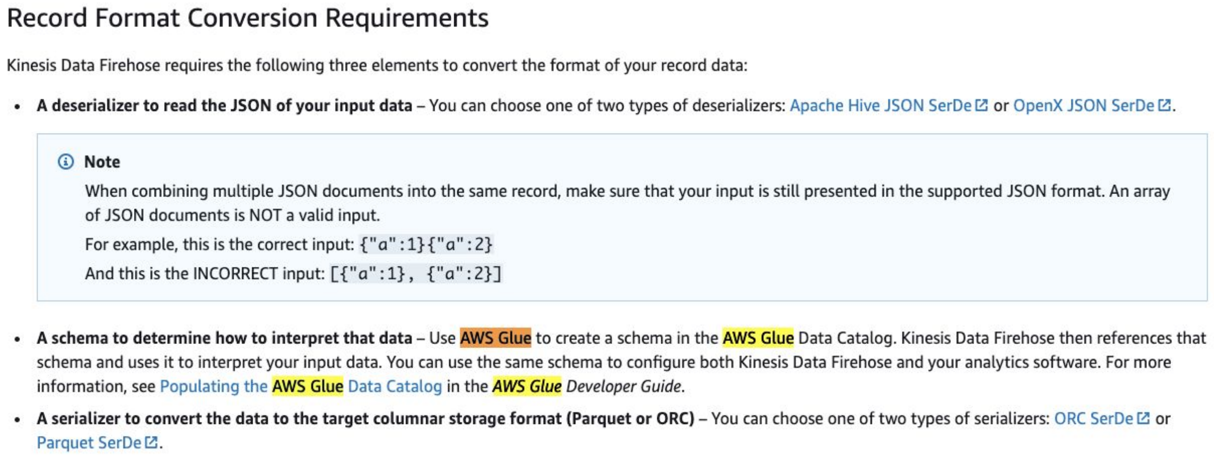
EI allows you to add **inference acceleration to a hosted endpoint** for a fraction of the cost of using a full GPU instance. Add an EI accelerator in one of the available sizes to a deployable model in addition to a CPU instance type, and then add that model as a production variant to an endpoint configuration that you use to deploy a hosted endpoint.
You can also add an EI accelerator to a SageMaker notebook instance so that you can test and evaluate inference performance when you are building your models.

Elastic Inference accelerates inference by allowing you to attach fractional GPUs to any SageMaker instance. You can select the client instance to run your application and attach an Elastic Inference accelerator to use the right amount of GPU acceleration for your inference needs.

Kinesis Firehose to convert JSON to Parquet (use AWS Glue)

“Use AWS Glue to create a schema in the AWS Glue Data Catalog. Kinesis Data Firehose then references that schema and uses it to interpret your input data”
https://docs.aws.amazon.com/firehose/latest/dev/record-format-conversion.html

Amazon Kinesis Data Firehose can convert the format of your input data from JSON to Apache Parquet or Apache ORC before storing the data in Amazon S3. Parquet and ORC are columnar data formats that save space and enable faster queries compared to row-oriented formats like JSON. If you want to convert an input format other than JSON, such as comma-separated values (CSV) or structured text, you can use AWS Lambda to transform it to JSON first. F



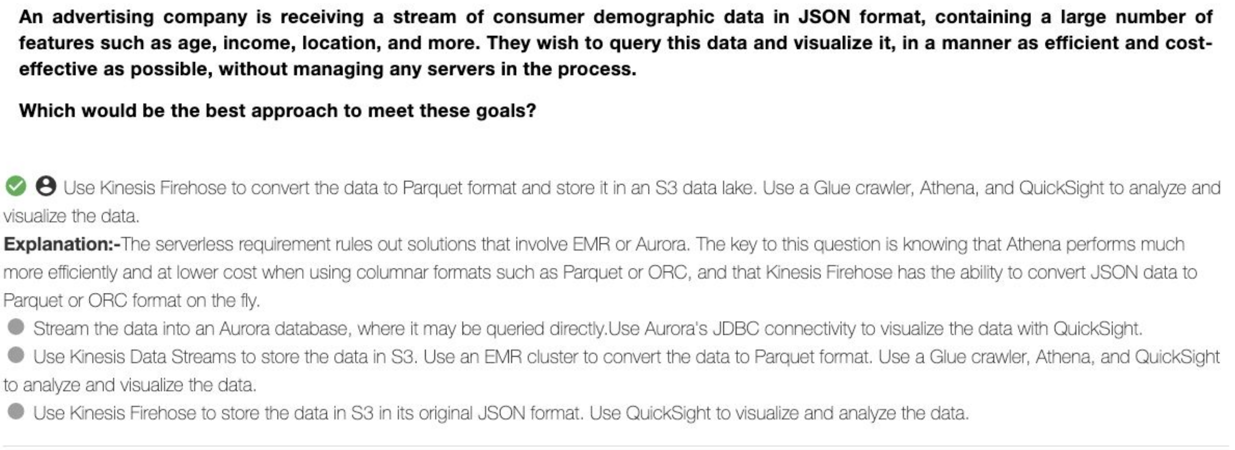
Record Format Conversion Requirements

Kinesis Data Firehose requires the following three elements to convert the format of your record data:

- **A deserializer to read the JSON of your input data** – You can choose one of two types of deserializers: Apache Hive JSON SerDe or OpenX JSON SerDe.

> **Note**
>
> When combining multiple JSON documents into the same record, make sure that your input is still presented in the supported JSON format. An array of JSON documents is NOT a valid input.
>
> For example, this is the correct input: `{"a":1}{"a":2}`
>
> And this is the INCORRECT input: `[{"a":1}, {"a":2}]`

- **A schema to determine how to interpret that data** – Use AWS Glue to create a schema in the AWS Glue Data Catalog. Kinesis Data Firehose then references that schema and uses it to interpret your input data. You can use the same schema to configure both Kinesis Data Firehose and your analytics software. For more information, see Populating the AWS Glue Data Catalog in the *AWS Glue Developer Guide*.
- **A serializer to convert the data to the target columnar storage format (Parquet or ORC)** – You can choose one of two types of serializers: ORC SerDe or Parquet SerDe.



An advertising company is receiving a stream of consumer demographic data in JSON format, containing a large number of features such as age, income, location, and more. They wish to query this data and visualize it, in a manner as efficient and cost-effective as possible, without managing any servers in the process.

Which would be the best approach to meet these goals?

✓ ● Use Kinesis Firehose to convert the data to Parquet format and store it in an S3 data lake. Use a Glue crawler, Athena, and QuickSight to analyze and visualize the data.

Explanation:-The serverless requirement rules out solutions that involve EMR or Aurora. The key to this question is knowing that Athena performs much more efficiently and at lower cost when using columnar formats such as Parquet or ORC, and that Kinesis Firehose has the ability to convert JSON data to Parquet or ORC format on the fly.

● Stream the data into an Aurora database, where it may be queried directly.Use Aurora's JDBC connectivity to visualize the data with QuickSight.

● Use Kinesis Data Streams to store the data in S3. Use an EMR cluster to convert the data to Parquet format. Use a Glue crawler, Athena, and QuickSight to analyze and visualize the data.

● Use Kinesis Firehose to store the data in S3 in its original JSON format. Use QuickSight to visualize and analyze the data.

Numerical Data Imputation with KNN

https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/

k-nearest neighbor (KNN) algorithm has proven to be generally effective, often referred to as “*nearest neighbor imputation*.

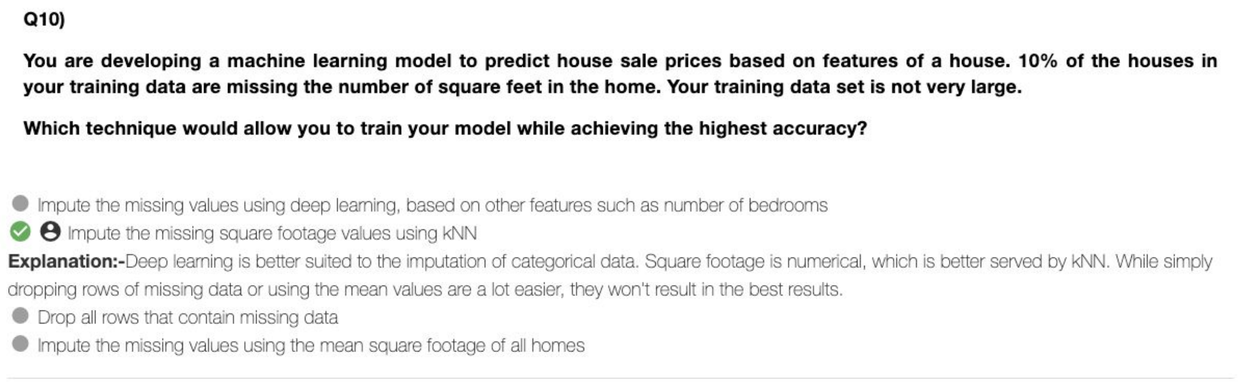
If input variables are **numeric**, then **regression** models can be used for prediction, and this case is quite common. A range of different models can be used, although a **simple k-nearest neighbor (KNN) model has proven to be effective** in experiments. The use of a KNN model to predict or fill missing values is referred to as “*Nearest Neighbor Imputation*” or “*KNN imputation*.”

Categorical Data Imputation

Options:

- Encode the data then apply numerical data imputation like Mean, KNN,...
- most common class
- Replace values with “Unkniwn” or “Misisng” => Fillna (“Unknown”)

Deep learning is better suited to the imputation of **categorical** data



Q10)

You are developing a machine learning model to predict house sale prices based on features of a house. 10% of the houses in your training data are missing the number of square feet in the home. Your training data set is not very large.

Which technique would allow you to train your model while achieving the highest accuracy?

● Impute the missing values using deep learning, based on other features such as number of bedrooms

✓ 👤 Impute the missing square footage values using kNN

Explanation:-Deep learning is better suited to the imputation of categorical data. Square footage is numerical, which is better served by kNN. While simply dropping rows of missing data or using the mean values are a lot easier, they won't result in the best results.

● Drop all rows that contain missing data

● Impute the missing values using the mean square footage of all homes https://www.countants.com/blogs/heres-how-you-can-configure-automatic-imputation-of-missing-data/

What are some of the common techniques used for missing data imputations?

Here are a few:

Mean, Median, and Mode technique

In this technique, the **mean**, **median**, or **mode** value is used as the imputation value used for statistical estimation of the missing data. For example, in the mean method, the mean value of a particular variable (containing numeric data) replaces the missing value of the same variable.

On a similar note, in the **median** technique, the median value of a variable (with **skewed** distribution pattern) is used as the missing value. For the **mode** technique, the missing value is replaced with the **most frequently** occurring variable (with **non-numerical value)** in the dataset.

Last Observation Carried Forward (LOCF)

This is a common imputation technique used in a **time-based dataset**. A missing value is replaced with the latest or last observed value in the dataset.

Next Observation Carried Backward (NOCB)

This imputation technique works in the reverse order to the LOCF method. That means the missing value is replaced with the next observed value in the dataset.

Multiple imputations

For this imputation technique, **a distributed set of observed data is used to estimate** a set of imputation values for the missing data. In this method, multiple datasets are created and then individually analyzed for estimating imputation values.

An example of multiple imputations is the **Multiple Imputation by Chained Equations or MICE** method. This method is executed through multiple **regression models** followed by modeling each missing value conditionally based on the observed values of the dataset.

Clustering Vs Collaborative filtering
https://www.quora.com/What-is-the-difference-between-Clustering-and-Collaborative-Filtering
Collaborative Filtering is a generic approach that can be summarized as "using information from similar users or items to predict affinity to a given item". There are many techniques that can be used for Collaborative Filtering. The two that are most well-known and discussed in the literature are **Nearest Neighbors (knn) and Matrix Factorization (MF)**. Knn is clearly a **supervised** method. As for MF, depending on the details of its usage one can call it supervised, unsupervised, or semi-supervised.

Clustering is usually defined as the (**unsupervised**) task of **grouping similar items together.**

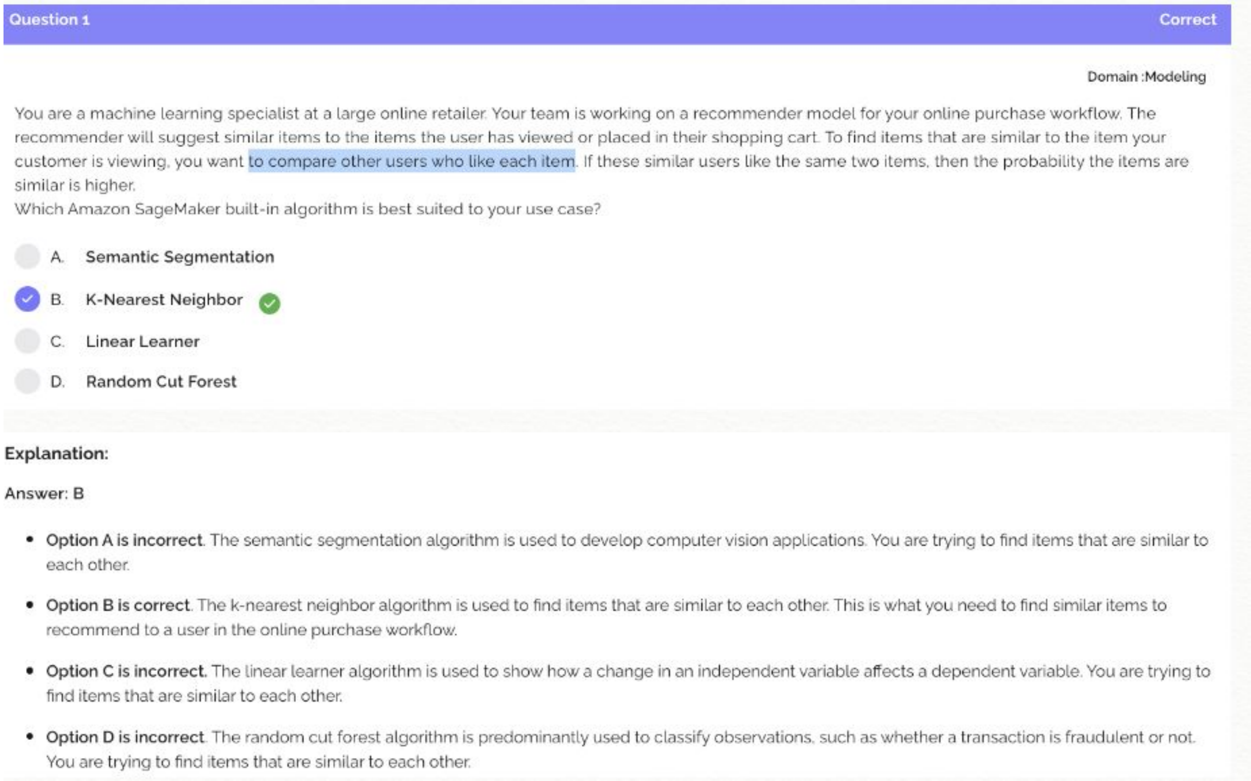
It is a basic technique of Machine Learning that **simply groups datapoints based on the similarity of each other**. The similarity is generally measured by any distance function eg. euclidean distance. Clustering is an unsupervised learning techinique wherein the data learns and corrects by itself without any prior knowledge. Finally, it converges to an output of groups of similar data points. The data points from one group to another are of different characteristics.

Well, it turns out that **most clustering methods can be used to implement Collaborative Filtering**. For most practical applications, you will need to combine clustering with something else since clustering is purely unsupervised. But you can still do at least primitive forms of CF based mostly on clustering. In order to do this you could, for example, cluster users and produce the final recommendations for all users in a given cluster as the "average preference" for all of them (e.g. rank items by how many users in that cluster have "liked" each item).

In **collaborative filtering,** we are given **partial information**, and the task is to fill up the missing entries (e.g. **Netflix** problem).

In **clustering**, typically **entire information is made available**. In other words, features and for all objects are given. The task is to **group objects together** with the criteria that objects with similar features must be assigned to the same group (any deviation from this will be penalized, and we must find a grouping with minimum loss)

Example with KNN (collaborative filtering)
KNN is used to find items that are similar to each other. This is what you need to find similar items to recommend to a user in the online workflow (same idea as Netflix recommendation)



Question 1 Correct

Domain :Modeling

You are a machine learning specialist at a large online retailer. Your team is working on a recommender model for your online purchase workflow. The recommender will suggest similar items to the items the user has viewed or placed in their shopping cart. To find items that are similar to the item your customer is viewing, you want to compare other users who like each item. If these similar users like the same two items, then the probability the items are similar is higher.
Which Amazon SageMaker built-in algorithm is best suited to your use case?

○ A. Semantic Segmentation

● B. K-Nearest Neighbor ✓

○ C. Linear Learner

○ D. Random Cut Forest

Explanation:

Answer: B

- **Option A is incorrect**. The semantic segmentation algorithm is used to develop computer vision applications. You are trying to find items that are similar to each other.
- **Option B is correct**. The k-nearest neighbor algorithm is used to find items that are similar to each other. This is what you need to find similar items to recommend to a user in the online purchase workflow.
- **Option C is incorrect**. The linear learner algorithm is used to show how a change in an independent variable affects a dependent variable. You are trying to find items that are similar to each other.
- **Option D is incorrect**. The random cut forest algorithm is predominantly used to classify observations, such as whether a transaction is fraudulent or not. You are trying to find items that are similar to each other.

Horovod: Single GPU Training to Multiple
https://github.com/horovod/horovod
The primary motivation for this project is to make it easy to take a single-GPU TensorFlow program and successfully train it on many GPUs faster.

AUC can generalize to multi-class
https://www.datascienceblog.net/post/machine-learning/performance-measures-multi-class-problems/
The area under the ROC curve (AUC) is a useful tool for evaluating the quality of class separation for soft classifiers. In the multi-class setting, we can visualize the performance of multi-class models according to their one-vs-all precision-recall curves. The AUC can also be generalized to the multi-class setting.

Input data to Training
https://docs.aws.amazon.com/sagemaker/latest/dg/cdf-training.html

---

Using CSV Format

Many Amazon SageMaker algorithms support training with data in CSV format. To use data in CSV format for training, in the input data channel specification, specify `text/csv` as the `ContentType`. Amazon SageMaker requires that a CSV file d**oes not have a header record and that the target variable is in the first column.**

To run **unsupervised** learning algorithms that don't have a target, specify the number of label columns in the content type. For example, in this case `'content_type=text/csv;label_size=0'`.

Using RecordIO Format

Most Amazon SageMaker algorithms work best when you use the **optimized protobuf** recordIO data format for training. Using this format allows you to take advantage of ***Pipe mode***.

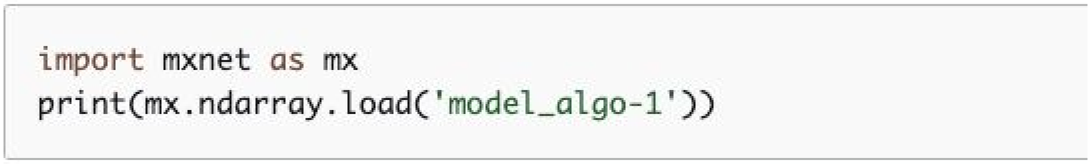
In *Pipe mode*, your training job streams data directly from Amazon Simple Storage Service (Amazon S3). Streaming can provide faster start times for training jobs and better throughput.

This is in contrast to *File mode*, in which your data from Amazon S3 is stored on the training instance volumes. File mode uses disk space to store both your final model artifacts and your full training dataset.

By streaming in your data directly from Amazon S3 in Pipe mode, **you reduce the size of Amazon Elastic Block Store volumes of your training instances.** Pipe mode needs only enough disk space to store your final model artifacts. See the `AlgorithmSpecification` for additional details on the training input mode.

Trained Model Deserialization

Amazon SageMaker models are stored as **model.tar.gz** in the S3 bucket specified in `OutputDataConfig` `S3OutputPath` parameter of the `create_training_job` call. You can specify most of these model artifacts when creating a hosting model. You can also open and review them in your notebook instance. When `model.tar.gz` is untarred, it contains `model_algo-1`, which is a serialized Apache MXNet object. For example, you use the following to load the k-means model into memory and view it:



```
import mxnet as mx
print(mx.ndarray.load('model_algo-1'))
```

Local run of SageMaker (when no internet)
https://aws.amazon.com/blogs/machine-learning/use-the-amazon-sagemaker-local-mode-to-train-on-your-notebook-instance/
Amazon SageMaker recently launched support for local training using the pre-built TensorFlow and MXNet containers
They've recently been open sourced, which means you can pull the containers into your working environment and use custom code built into the Amazon SageMaker Python SDK to test your algorithm **locally**, **just by changing a single line of code**.
This means that you can iterate and test your work without having to wait for a new training or hosting cluster to be built each time.
The local mode in the Amazon SageMaker Python SDK can emulate CPU (single and multi-instance) and GPU (single instance) SageMaker training jobs by changing a single argument in the TensorFlow or MXNet estimators. To do this, it uses Docker compose and NVIDIA Docker. It

will also pull the Amazon SageMaker TensorFlow or MXNet containers from Amazon ECS, so you'll need to be able to access a public Amazon ECR repository from your local environment. If you choose to use a SageMaker notebook instance as your local environment, this script will install the necessary prerequisites. Otherwise, you can install them yourself, and make sure you've upgraded to the latest version of the SageMaker Python SDK with `pip install -U sagemaker`.

Kinesis Analytics - on gzip files leveraging lambda

https://aws.amazon.com/about-aws/whats-new/2017/10/amazon-kinesis-analytics-can-now-pre-process-data-prior-to-running-sql-queries/

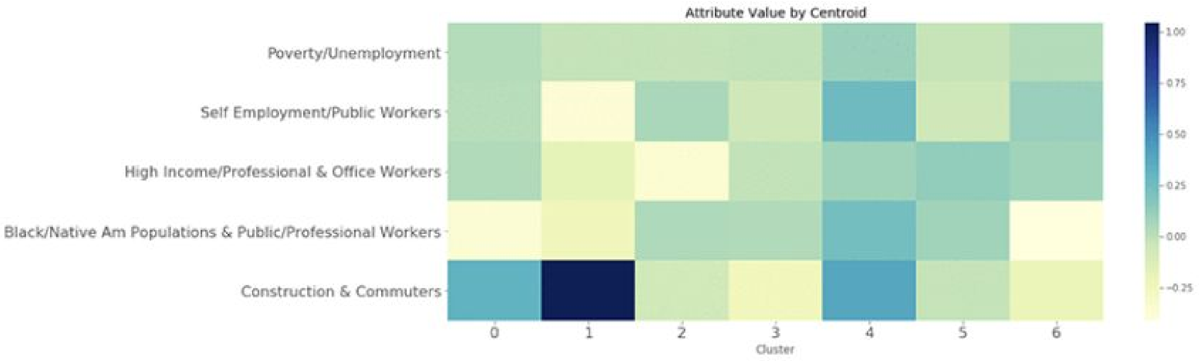
You can now configure your Amazon Kinesis Analytics applications to **transform data before it is processed by your SQL code**. This new feature allows you to use AWS Lambda to convert formats, enrich data, filter data, and more. Once the data is transformed by your function, Kinesis Analytics sends the data to your application's SQL code for real-time analytics.

To get started, simply select an **AWS Lambda function** from the Kinesis Analytics application source page in the AWS Management console. Your **Kinesis Analytics application will automatically process your raw data records using the Lambda function, and send transformed data to your SQL code for further processing.**

Census data - use PCA and Kmeans

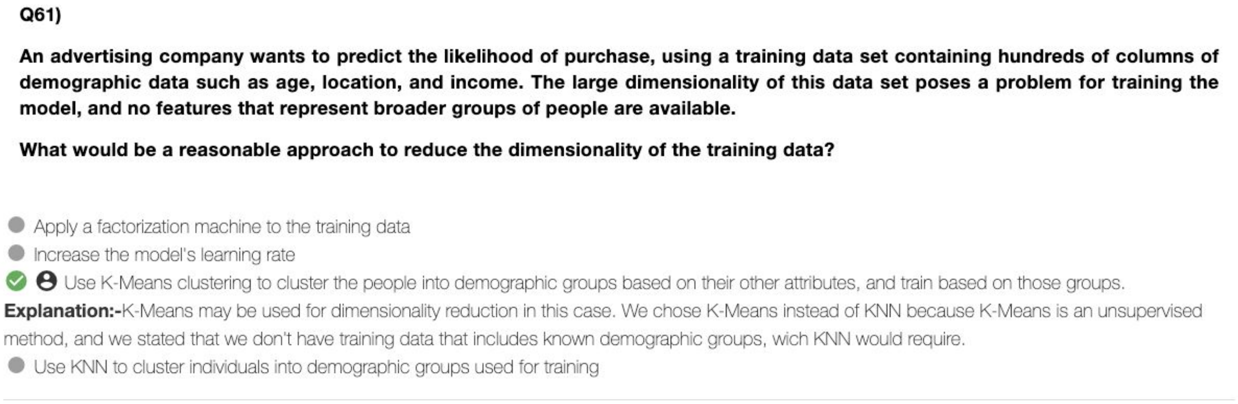
https://aws.amazon.com/blogs/machine-learning/analyze-us-census-data-for-population-segmentation-using-amazon-sagemaker/

We will be using principal component analysis (**PCA**) to reduce the dimensionality of our data.
Now, we'll use the **Kmeans** algorithm to segment the population of counties by the 5 PCA attributes we have created. Kmeans is a clustering algorithm that identifies clusters of similar counties based on their attributes. Since we have ~3000 counties and 34 attributes in our original dataset, the large feature space may have made it difficult to cluster the counties effectively. Instead, we have reduced the feature space to 5 PCA components, and we'll cluster on this transformed dataset.



KNN Vs K-Means clustering for predicting likelihood to purchase

K-means used here for dimensionality reduction.
We chose K-Means instead of KNN because K-means is an unsupervised method, and we stated that we don't have training data that includes known demographic groups, which KNN would require.



Q61)

An advertising company wants to predict the likelihood of purchase, using a training data set containing hundreds of columns of demographic data such as age, location, and income. The large dimensionality of this data set poses a problem for training the model, and no features that represent broader groups of people are available.

What would be a reasonable approach to reduce the dimensionality of the training data?

● Apply a factorization machine to the training data
● Increase the model's learning rate
✓ ● Use K-Means clustering to cluster the people into demographic groups based on their other attributes, and train based on those groups.
Explanation:-K-Means may be used for dimensionality reduction in this case. We chose K-Means instead of KNN because K-Means is an unsupervised method, and we stated that we don't have training data that includes known demographic groups, wich KNN would require.
● Use KNN to cluster individuals into demographic groups used for training

DeepAR

https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html
"...When your dataset contains hundreds of related time series, DeepAR outperforms the standard ARIMA and ETS methods.
You can also use the trained model to generate forecasts for new time series that are **similar to the ones it has been trained on**."

Using multiple GPU

https://aws.amazon.com/blogs/machine-learning/the-importance-of-hyperparameter-tuning-for-scaling-deep-learning-training-to-multiple-gpus/
In each training iteration, typically a small subset of the dataset, called a mini-batch, is processed. When a single GPU is available, processing of the mini-batch in each training iteration is handled by this GPU. When training with multiple GPUs, the mini-batch is split across available GPUs to evenly spread the processing load. To ensure that you **fully use each GPU**, you must **increase the mini-batch size linearly with each additional GPU**.
Mini-batch size has an impact not only on training speed, but also on the quality of the trained model. **As you increase the mini-batch size**, it is important to tune other hyperparameters to ensure faster training with similar model quality.
Interestingly enough, the technique of **increasing the learning rate with an increase in mini-batch size has been shown to reduce the impact of having a large mini-batch on model quality.**

Oversampling:

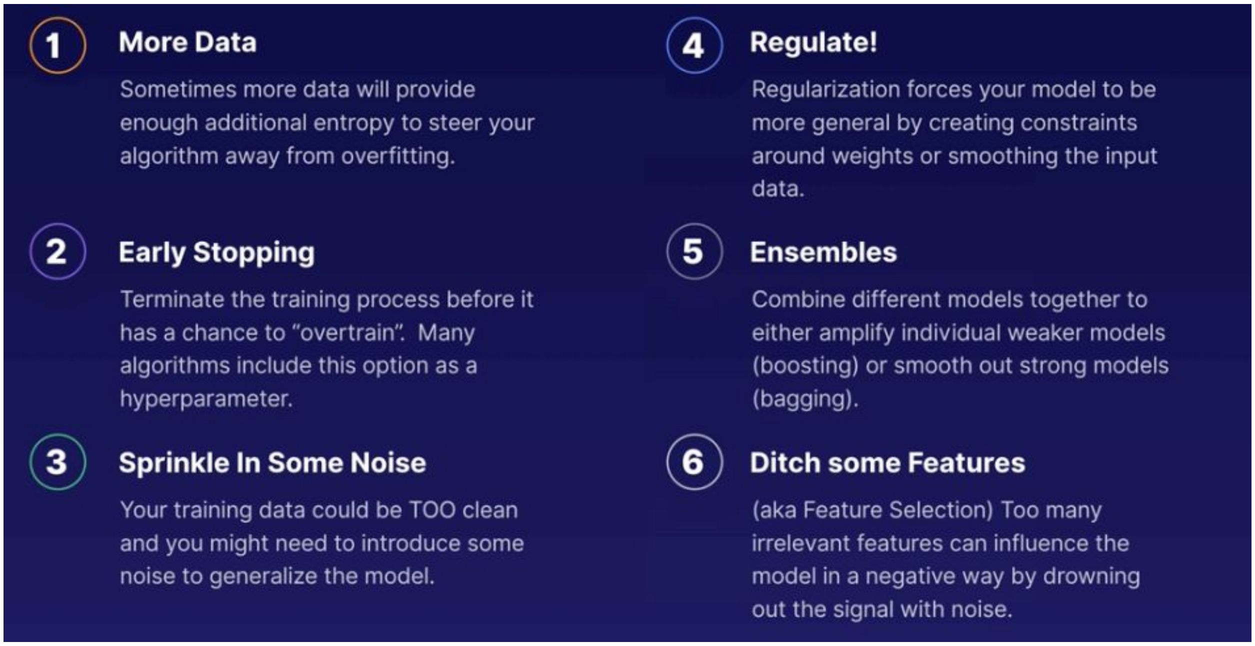
- **SMOTE** (Synthetic Minority Oversampling Technique) - creates new observations of the underrepresented class. Synthetic observations are almost identical to the original set. This technique is expeditious but the types of synthetic observations produced are not as useful as other techniques like GAN
- **GAN** (Generative Adversial Networks) - generates unique observations that more closely resemble the minority without being so similar that they are almost identical.

Techniques to reduce underfitting :

1. Increase model complexity.
2. Increase number of features, performing feature engineering.
3. Remove noise from the data.
4. Increase the number of epochs or increase the duration of training to get better results.

Techniques to reduce overfitting :

1. Increase training data.
2. Reduce model complexity.
3. Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
4. Ridge(L2) Regularization and Lasso (L1) Regularization
5. Use dropout for neural networks to tackle overfitting.



1 **More Data**

Sometimes more data will provide enough additional entropy to steer your algorithm away from overfitting.

2 **Early Stopping**

Terminate the training process before it has a chance to "overtrain". Many algorithms include this option as a hyperparameter.

3 **Sprinkle In Some Noise**

Your training data could be TOO clean and you might need to introduce some noise to generalize the model.

4 **Regulate!**

Regularization forces your model to be more general by creating constraints around weights or smoothing the input data.

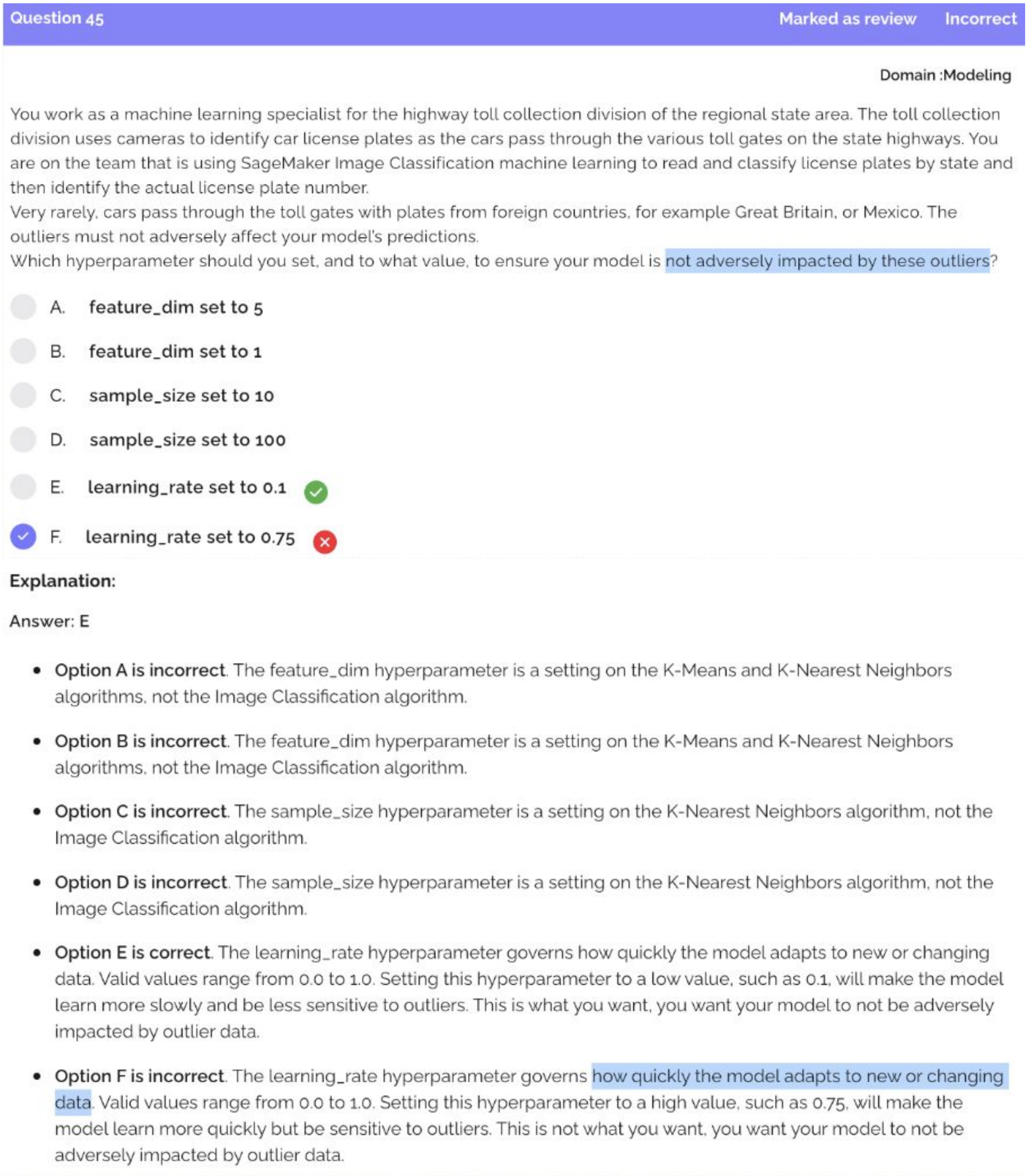
5 **Ensembles**

Combine different models together to either amplify individual weaker models (boosting) or smooth out strong models (bagging).

6 **Ditch some Features**

(aka Feature Selection) Too many irrelevant features can influence the model in a negative way by drowning out the signal with noise.

Hyper parameters to not be impacted by outliers
learning_Rate to be low as learning rate governs how quickly a model adapts to new or changing data. If it’s set high, the model will learn quickly but be sensible to outliers. This is not what we want.



Question 45 Marked as review Incorrect

Domain :Modeling

You work as a machine learning specialist for the highway toll collection division of the regional state area. The toll collection division uses cameras to identify car license plates as the cars pass through the various toll gates on the state highways. You are on the team that is using SageMaker Image Classification machine learning to read and classify license plates by state and then identify the actual license plate number.

Very rarely, cars pass through the toll gates with plates from foreign countries, for example Great Britain, or Mexico. The outliers must not adversely affect your model's predictions.

Which hyperparameter should you set, and to what value, to ensure your model is not adversely impacted by these outliers?

○ A. feature_dim set to 5

○ B. feature_dim set to 1

○ C. sample_size set to 10

○ D. sample_size set to 100

○ E. learning_rate set to 0.1 ✓

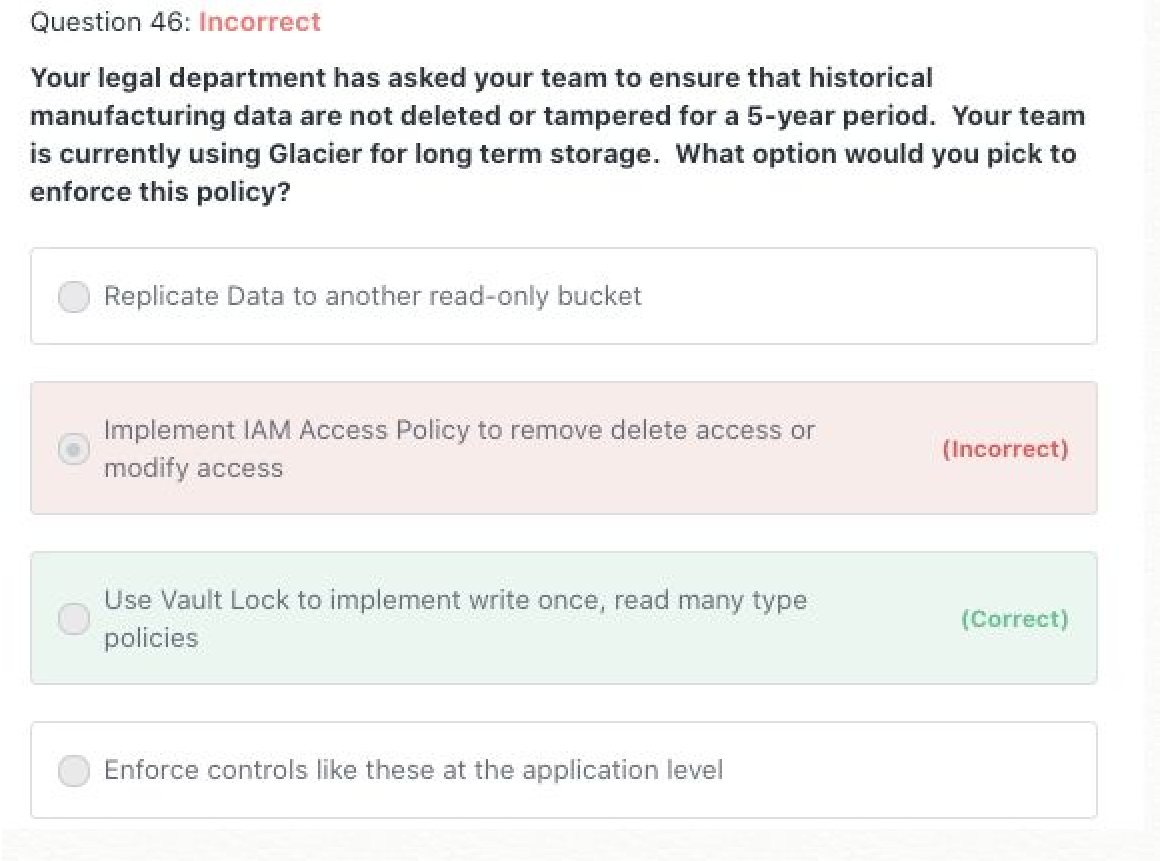
● F. learning_rate set to 0.75 ✗

Explanation:

Answer: E

- **Option A is incorrect**. The feature_dim hyperparameter is a setting on the K-Means and K-Nearest Neighbors algorithms, not the Image Classification algorithm.
- **Option B is incorrect**. The feature_dim hyperparameter is a setting on the K-Means and K-Nearest Neighbors algorithms, not the Image Classification algorithm.
- **Option C is incorrect**. The sample_size hyperparameter is a setting on the K-Nearest Neighbors algorithm, not the Image Classification algorithm.
- **Option D is incorrect**. The sample_size hyperparameter is a setting on the K-Nearest Neighbors algorithm, not the Image Classification algorithm.
- **Option E is correct**. The learning_rate hyperparameter governs how quickly the model adapts to new or changing data. Valid values range from 0.0 to 1.0. Setting this hyperparameter to a low value, such as 0.1, will make the model learn more slowly and be less sensitive to outliers. This is what you want, you want your model to not be adversely impacted by outlier data.
- **Option F is incorrect**. The learning_rate hyperparameter governs how quickly the model adapts to new or changing data. Valid values range from 0.0 to 1.0. Setting this hyperparameter to a high value, such as 0.75, will make the model learn more quickly but be sensitive to outliers. This is not what you want, you want your model to not be adversely impacted by outlier data.

Vault Lock for enforcing policies



Question 46: Incorrect

Your legal department has asked your team to ensure that historical manufacturing data are not deleted or tampered for a 5-year period. Your team is currently using Glacier for long term storage. What option would you pick to enforce this policy?

○ Replicate Data to another read-only bucket

● Implement IAM Access Policy to remove delete access or modify access **(Incorrect)**

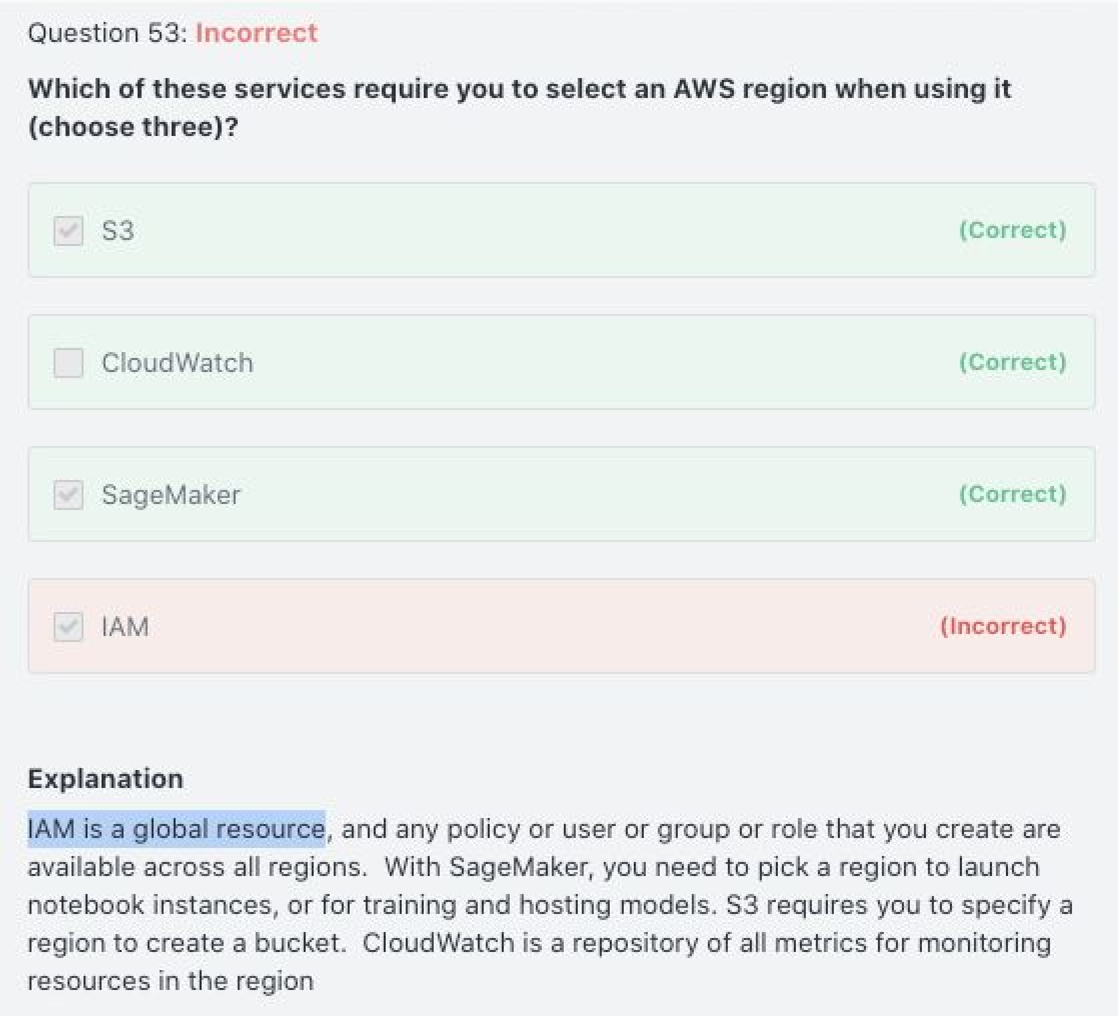
○ Use Vault Lock to implement write once, read many type policies **(Correct)**

○ Enforce controls like these at the application level

Explanation

Vault Lock allows you to set immutable policies to enforce compliance controls. With the IAM Access policy, you can define who has access to storage and type of access. However, the IAM policy on its own is not sufficient for compliance-related controls as someone could change the policy to grant write permissions

Global vs local services to a region



Question 53: Incorrect

Which of these services require you to select an AWS region when using it (choose three)?

☑ S3 (Correct)

☐ CloudWatch (Correct)

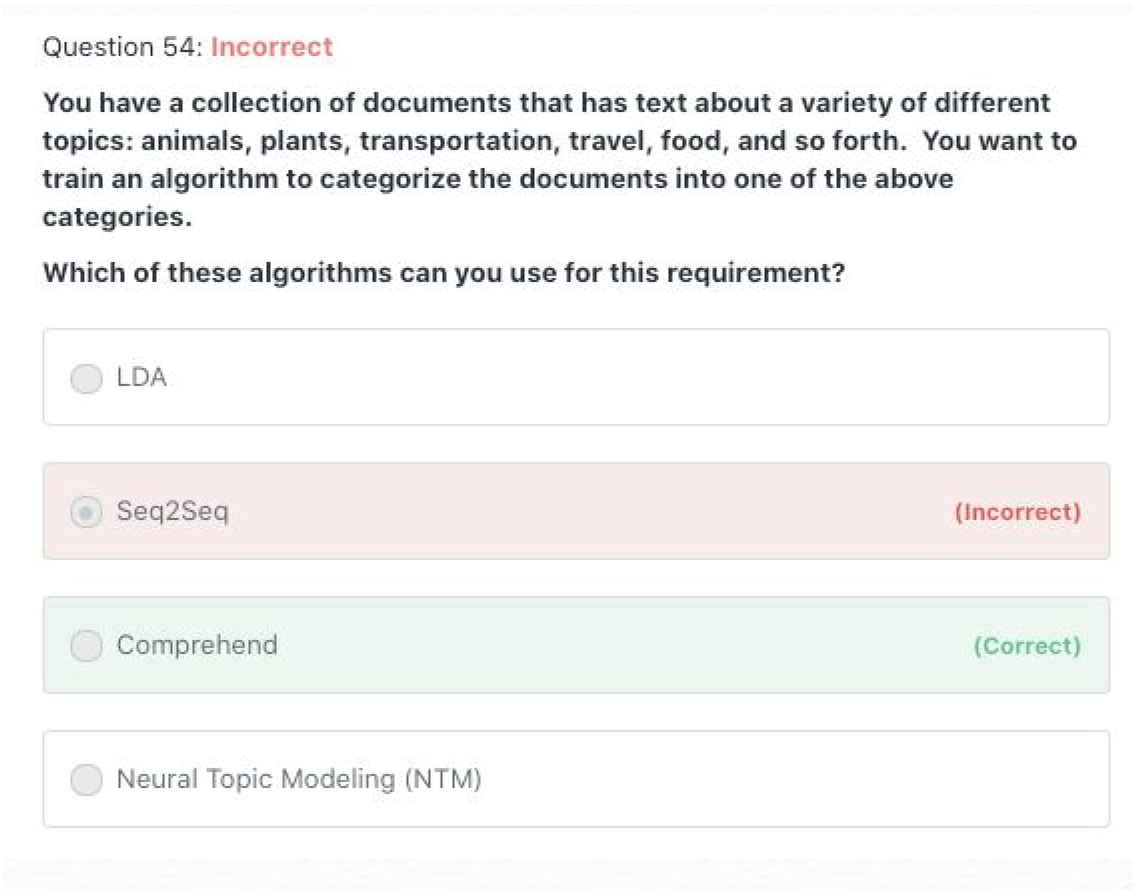
☑ SageMaker (Correct)

☑ IAM (Incorrect)

Explanation

IAM is a global resource, and any policy or user or group or role that you create are available across all regions. With SageMaker, you need to pick a region to launch notebook instances, or for training and hosting models. S3 requires you to specify a region to create a bucket. CloudWatch is a repository of all metrics for monitoring resources in the region

Categorize documents



Question 54: Incorrect

You have a collection of documents that has text about a variety of different topics: animals, plants, transportation, travel, food, and so forth. You want to train an algorithm to categorize the documents into one of the above categories.

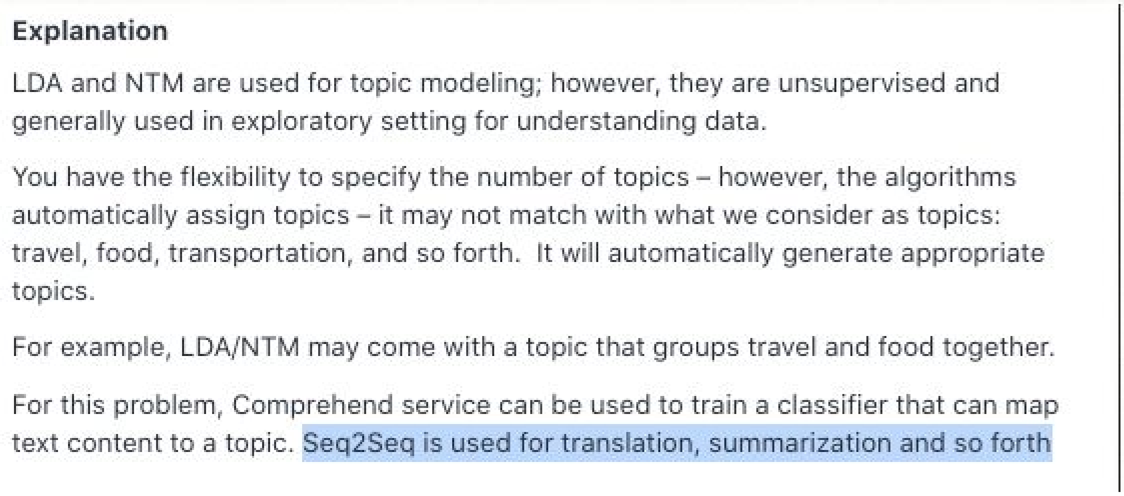
Which of these algorithms can you use for this requirement?

○ LDA

● Seq2Seq **(Incorrect)**

○ Comprehend **(Correct)**

○ Neural Topic Modeling (NTM)



Explanation

LDA and NTM are used for topic modeling; however, they are unsupervised and generally used in exploratory setting for understanding data.

You have the flexibility to specify the number of topics – however, the algorithms automatically assign topics – it may not match with what we consider as topics: travel, food, transportation, and so forth. It will automatically generate appropriate topics.

For example, LDA/NTM may come with a topic that groups travel and food together.

For this problem, Comprehend service can be used to train a classifier that can map text content to a topic. Seq2Seq is used for translation, summarization and so forth

=> Use **Comprehend** and **train a custom classifier** using [CreateDocumentClassifier](https://docs.aws.amazon.com/comprehend/latest/dg/how-document-classification-training.html):
https://docs.aws.amazon.com/comprehend/latest/dg/how-document-classification-training.html

Seq2Seq
https://docs.aws.amazon.com/sagemaker/latest/dg/seq-2-seq.html
supervised learning algorithm where the input is a sequence of tokens (for example, text, audio) and the output generated is another sequence of tokens. Example applications include:

- **machine translation** (input a sentence from one language and predict what that sentence would be in another language)
- **text summarization** (input a longer string of words and predict a shorter string of words that is a summary)
- **speech-to-text** (audio clips converted into output sentences in tokens).

Use Cases

- Translation
- Text Summarization
- Video captioning
- Speech recognition

Example:
https://github.com/aws/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/seq2seq_translation_en-de/SageMaker-Seq2Seq-Translation-English-German.ipynb

Input: **RecordIO-Protobuf** format. However, the tokens are expected as **integers**, not as floating points

Notes about SageMaker Algorithms

Algo	Training Input	Input mode	Channels	Inference	Instance class	Use Cases
XGboost (S)	**CSV** **libSVM (default)** Libsvm: assumes label is in the first column. Subsequent columns contain the zero-based index value pairs for features: <label> <index0>:<value0> <index1>:<value1> ... CSV: algorithm assumes the target variable is in the first column and that the CSV does not have a header record.	File Pipe	must: train optional: validation	`Csv` `libsvm`	**CPU only** (need enough memory to hold trainign set) M5 better than C4 **No GPU**	**Regression** **Classification** **Ranking**

Seq2seq
(S)
	RecordIO-Protobuf
	Integers
	(not float)
	
	sequence of tokens (for example, text, audio)
		File	must:
			train
			validation
			vocab
				`json`
				and
				`x-recordio-protobuf`
					GPU only
					Single instance
						Translation
						
						Text Summarization
						
						Video captioning
						
						Speech recognition

LDA (U)	**CSV** (dense) **RecordIO**-wrapped-Protobuf (dense and sparse)	File Pipe	must: train optional: test	`csv`, `json`, and `x-recordio-protobuf`	CPU **Single** Instance	**Topic modeling** Discover user-specified number of topics shared by documents within a text corpus. Used to figure out how similar documents are based on the frequency of similar words Ex: recommend articles based on similar topics we read or rated before
NTM (U)	**CSV** (dense) **RecordIO**-wrapped-Protobuf (dense and sparse)	File Pipe	must: train optional: test validation: auxiliary	`csv`, `json`, and `x-recordio-protobuf`	**GPU or CPU**	**Topic modeling** Ex: Organize a set of documents into topics (not known in advance): tag a document as belonging to a medical category based on the terms used in the document.
K-Means **(U)**	**CSV** **RecordIO**-wrapped-Protobuf	File Pipe	must: train optional: test	`csv`, `json`, and `x-recordio-protobuf`	**CPU** **or 1 GPU *P*.xl)** with only 1 gpu per instance	**Clustering** **Grouping** We specify the number of clusters and which attributes indicate similarity and the algo will attempt to group them together Ex: Group similar objects/data together: find high-, medium-, and low-spending customers from their transaction histories
K-NN (S)	**CSV** **RecordIO**-wrapped-Protobuf	File Pipe	must: train optional: test	`csv`, `json`, and `x-recordio-protobuf`	**CPU** **or GPU *P*.xl)** with	**Classification** **Regression** ***Lazy* algorithm:** does not use training data

					only 1 gpu per instance	to generalize but rather uses them to figure out who’s nearby Predicts the value or classification based on that which are closest. Need to specify K Ex: group people together for credit risk based on attributes they share with others of known credit usage. Based on what someone likes, recommend similar items they might also like
Object2Vec (S)	**JSON** discrete token: `[10]`. sequences of discrete tokens: `[0,12,10,13]`.	File	must: train optional: test	`json`	**CPU** **or GPU**	Embeddings: convert highdimensional objects into lowdimensionalspace . Use embeddings for downstream supervised tasks such as classification or regression. . Ex: Improve the data embeddings of the high-dimensional objects: identify duplicate support tickets or find the corrrect routing based on similarity of text in the tickets Ex: predict the rating a person is likely to give a movie based on similarity to other’s movie ratings
BlazingText (**S**: Text Classifica	**TEXT** with space-separated tokens. Each line is a sentence	File Pipe with augme	must: train	`json`	**CPU** **or GPU** single instance	**Text Classification** Ex: Assign pre-defined categories to

tion **U**: cbow, skip-gram, batch skip-gram)	For supervised (Text classification), add a label at the beginning: ***__label_4***	nted manifest json file				documents in a corpus: categorize books in a library into academic disciplines BlazingText in Word2Vec can only finds embeddings against WORDS, not full sentences. Use Object2Vec for sentences.
Factorization Machine (S)	**RecordIO**-Protobuf **Float32** NO CSV	File Pipe	must: train optional: test	`json` and `x-recordio-protobuf`	**CPU GPU**	**Binary Classification Regression** (extension of linear learner) Used for **Item recommendations or click predictions** on sparse data NO multi class Only analyze relationship of 2 pairs of features at a time Needs lots of data because of sparseness Ex: good choice for tasks dealing with high dimensional sparse datasets, such as click prediction and item recommendation
Linear Learner (S)	**CSV** **RecordIO**-Protobuf **Float32**	File Pipe	must: train optional: validation Test If validation:	`Json, csv` and `x-recordio-protobuf`	**CPU GPU**	**Classification Regression** Well suited for discrete or continuous inference

			`S3DataDistributionType = FullyReplicated`			Built in tuning hyperparameters Ex: breast cancer, MNIST dataset,...
[RCF](#) (U)	**CSV** **RecordIO**-Protobuf **Float32**	File Pipe	must: train optional: test	`Json, csv` and `x-recordio-protobuf`	**CPU** **No GPU**	**Anomaly detection** Low score is normal High score is an anomaly

Object2Vec

https://aws.amazon.com/blogs/machine-learning/introduction-to-amazon-sagemaker-object2vec/?ref=Welcome.AI

Embeddings are an important feature engineering technique in machine learning (ML). They convert high dimensional vectors into low-dimensional space to make it easier to do machine learning with large sparse vector inputs. Embeddings also capture the semantics of the underlying data by placing similar items closer in the low-dimensional space. This makes the features more effective in training downstream models. One of the well-known embedding techniques is *Word2Vec*, which provides embeddings for words. It has been widely used in many use cases, such as sentiment analysis, document classification, and natural language understanding. See the following diagram for a conceptual representation of word embeddings in the feature space.

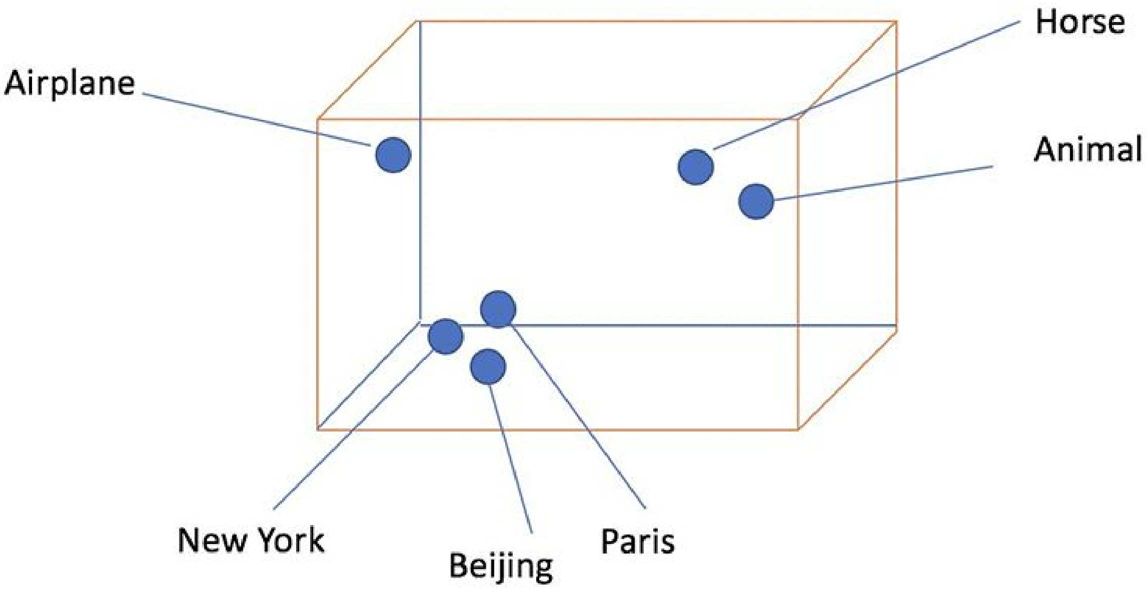


Figure 1: Word2Vec embeddings: words that are semantically similar are close together in the embedding space.

In addition to word embeddings, there are also use cases where we want to learn the embeddings of more general-purpose objects such as sentences, customers, and products. This is so we can build practical applications for **information retrieval, product search, item matching, customer profiling based on similarity or as inputs for other supervised tasks.**

Blazing Text

The Amazon SageMaker BlazingText algorithm provides highly optimized implementations of the Word2vec and text classification algorithms.

To summarize, the following modes are supported by BlazingText on different types instances:

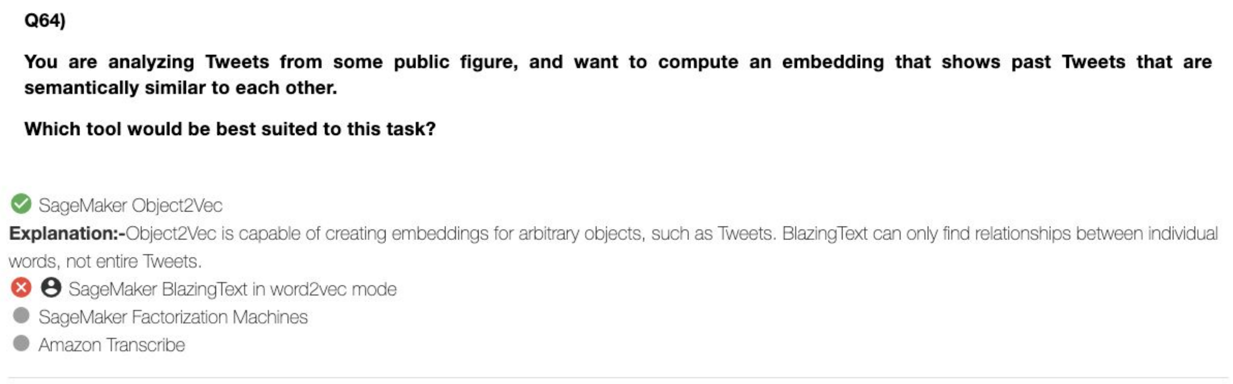
Modes	Word2Vec (Unsupervised Learning)	Text Classification (Supervised Learning)

Single CPU instance	`cbow`
	`Skip-gram`
	`Batch Skip-gram`
		`supervised`

Single GPU instance (with 1 or more GPUs)	`cbow` `Skip-gram`	`supervised` with one GPU
Multiple CPU instances	`Batch Skip-gram`	None

Blazing Text vs Object2Vec for Tweeter feeds

- **Object2Vec** is capable of creating embeddings for **arbitrary objects, such as Tweets**
- **BlazingText** can only find relationships between **individual WORDS**, not entire tweets.



Q64)

You are analyzing Tweets from some public figure, and want to compute an embedding that shows past Tweets that are semantically similar to each other.

Which tool would be best suited to this task?

✓ SageMaker Object2Vec

Explanation:-Object2Vec is capable of creating embeddings for arbitrary objects, such as Tweets. BlazingText can only find relationships between individual words, not entire Tweets.

✗ 👤 SageMaker BlazingText in word2vec mode

● SageMaker Factorization Machines

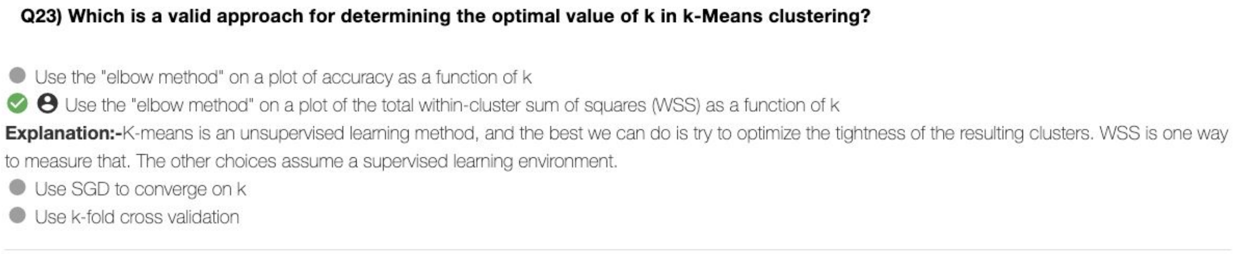
● Amazon Transcribe

LDA

To improve performances, use PIPE mode with RecordIO format.
DON’T use multiple GPU - LDA only supports training on a **single-instance CPU**

K-MEANS - determine optimal value of K

Use the **“elbow method”** on a plot of the total **within-cluster sum of squares** (WSS) as a function of k



Q23) Which is a valid approach for determining the optimal value of k in k-Means clustering?

● Use the "elbow method" on a plot of accuracy as a function of k

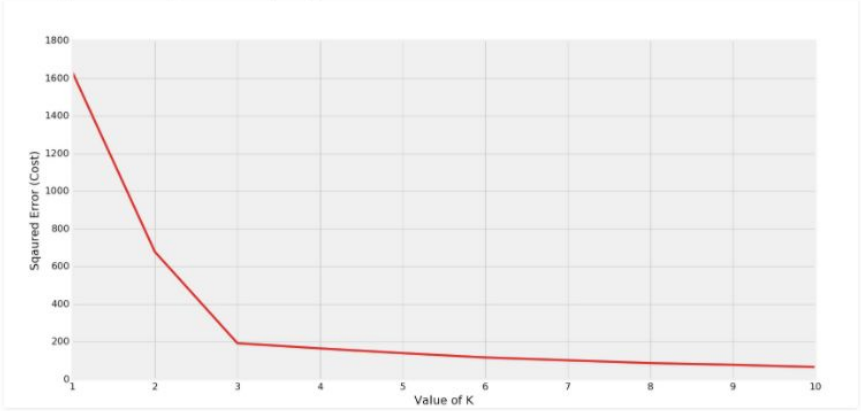
✓ Use the "elbow method" on a plot of the total within-cluster sum of squares (WSS) as a function of k

Explanation:-K-means is an unsupervised learning method, and the best we can do is try to optimize the tightness of the resulting clusters. WSS is one way to measure that. The other choices assume a supervised learning environment.

● Use SGD to converge on k

● Use k-fold cross validation

In the above figure, its clearly observed that the distribution of points are forming 3 clusters. Now, let’s see the plot for the squared error(Cost) for different values of K.



Elbow is forming at K=3

XGBoost

Key hyper parameters

Required

- **num_round:** number of rounds to turn the training

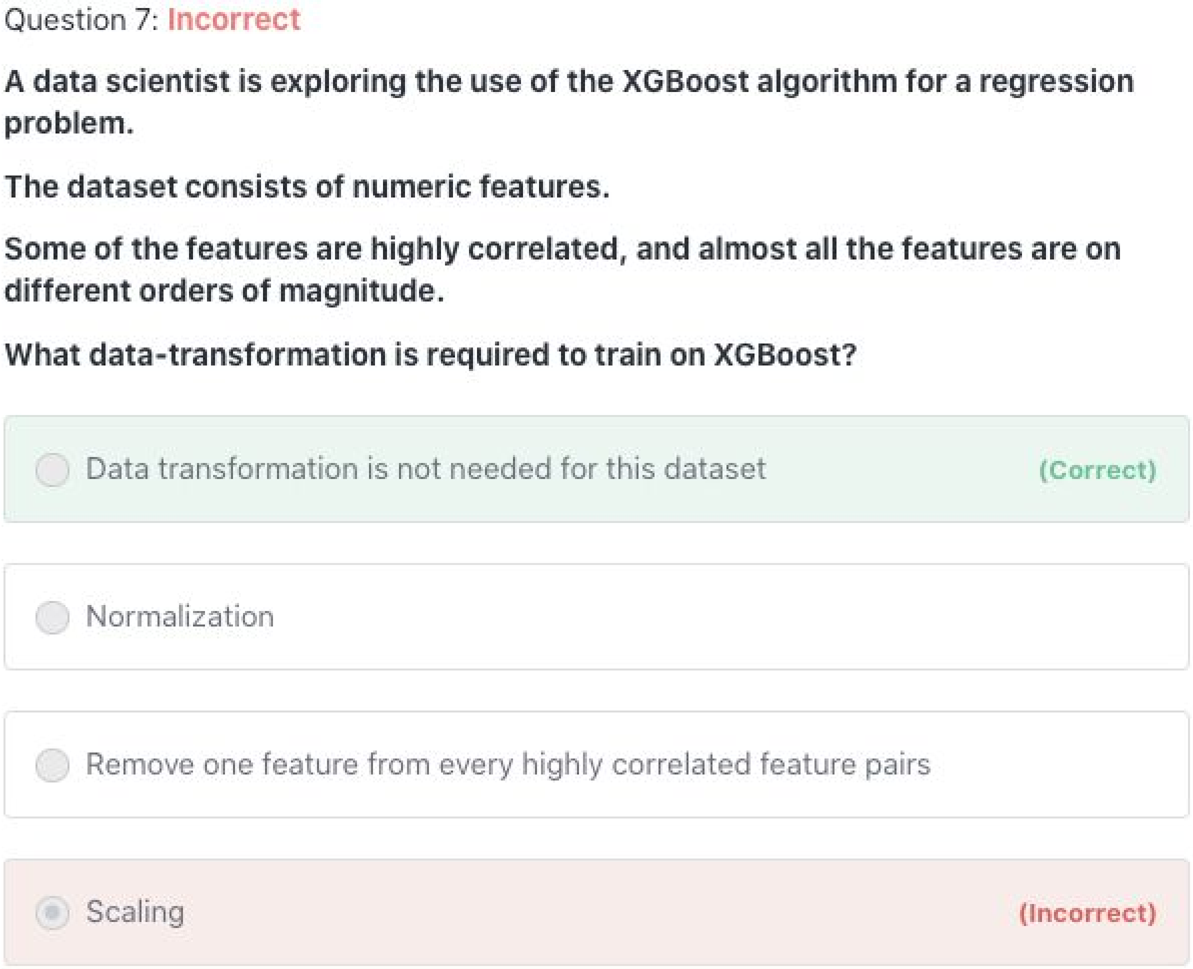
Required when objective is set to **ulti:softmax (or multi:softprob)**

- **num_class:** number of classes

Optional:

- **alpha:** L1 (lasso) regularization term on weight. Increasing this value makes the model more conservative
- **lambda**: L2 (ridge) regularization term on weight. Increasing this value makes the model more conservative
- **gamma:** minimum loss reduction to make a further partition on a leaf node of the tree. The larger, the more conservative the algorithm is.

- **booster**: “gbtree” by default or “dart” using tree-based model while “gblinear” uses a linear function
- **Csv_weight**:0 or 1. If 1 (enabled), XGboost differentiates the importance of instances for csv input by taking the second column (after labels) in training dat aas the instance weights.
- **early_stopping_rounds**: model trains until the validation score stops improving. Validation error needs to decrease at least every early_stopping_rounds to continue training.
- **eta**: [0,1] range with 0.3 default. step size shrinkage used in updates to **prevent overfitting**. It shrinks the feature weights to make the boosting process more conservative.
- **eval_metric**: “**rmse**” for regression, “**error**” for classification and “map” for ranking.
- **subsample:** [0,1] range. Default is 1. subsample ratio of the training instance. Setting it to 0.5 means that XGboost randomly collects half of the data instances to grow trees. This **prevents overfitting.**



Question 7: Incorrect

A data scientist is exploring the use of the XGBoost algorithm for a regression problem.

The dataset consists of numeric features.

Some of the features are highly correlated, and almost all the features are on different orders of magnitude.

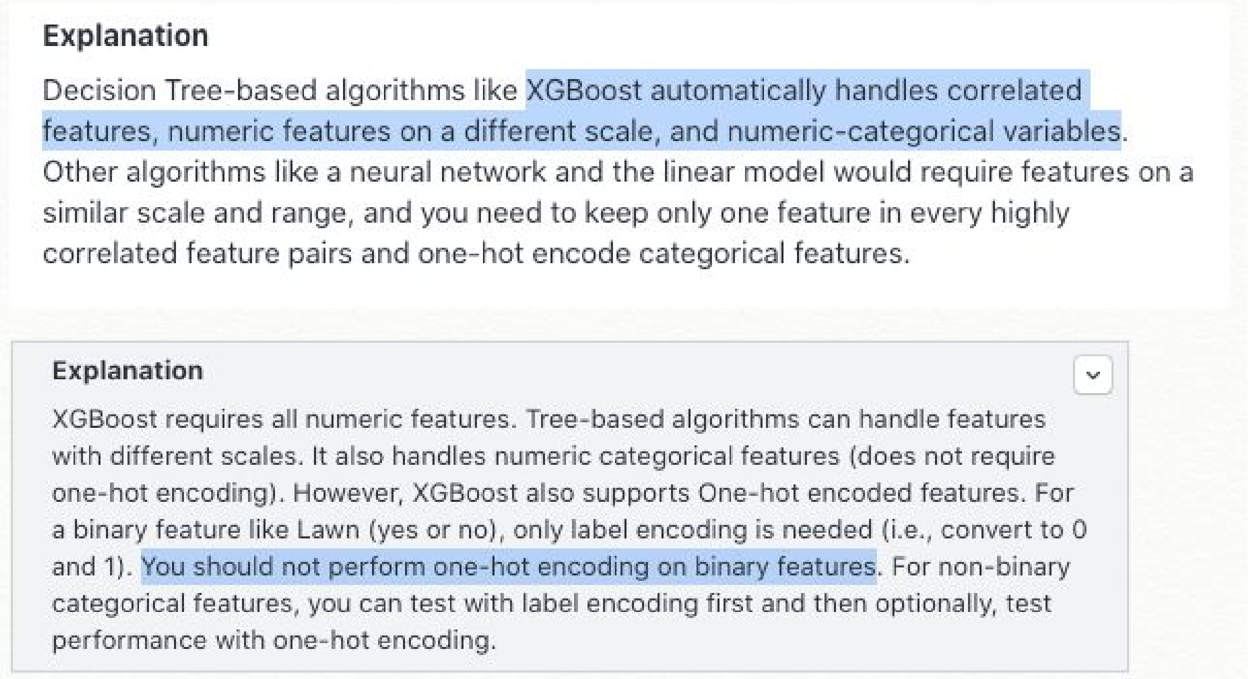
What data-transformation is required to train on XGBoost?

○ Data transformation is not needed for this dataset **(Correct)**

○ Normalization

○ Remove one feature from every highly correlated feature pairs

● Scaling **(Incorrect)**



Explanation

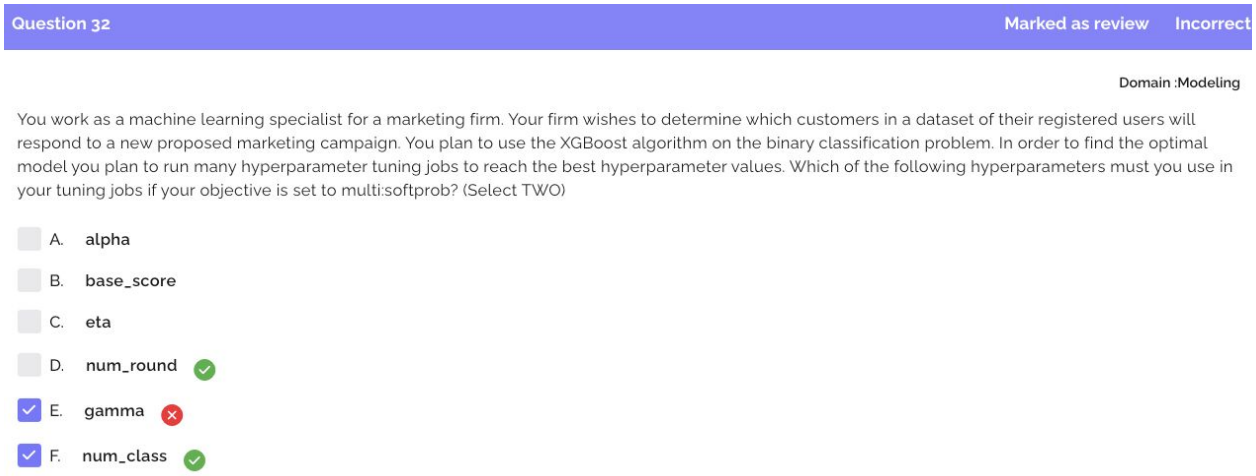
Decision Tree-based algorithms like XGBoost automatically handles correlated features, numeric features on a different scale, and numeric-categorical variables. Other algorithms like a neural network and the linear model would require features on a similar scale and range, and you need to keep only one feature in every highly correlated feature pairs and one-hot encode categorical features.

Explanation

XGBoost requires all numeric features. Tree-based algorithms can handle features with different scales. It also handles numeric categorical features (does not require one-hot encoding). However, XGBoost also supports One-hot encoded features. For a binary feature like Lawn (yes or no), only label encoding is needed (i.e., convert to 0 and 1). You should not perform one-hot encoding on binary features. For non-binary categorical features, you can test with label encoding first and then optionally, test performance with one-hot encoding.

XGBoosts with multi-softprob

=> **num_class** and **num_round** must be specified as hyper parameters



Question 32 | Marked as review | Incorrect

Domain :Modeling

You work as a machine learning specialist for a marketing firm. Your firm wishes to determine which customers in a dataset of their registered users will respond to a new proposed marketing campaign. You plan to use the XGBoost algorithm on the binary classification problem. In order to find the optimal model you plan to run many hyperparameter tuning jobs to reach the best hyperparameter values. Which of the following hyperparameters must you use in your tuning jobs if your objective is set to multi:softprob? (Select TWO)

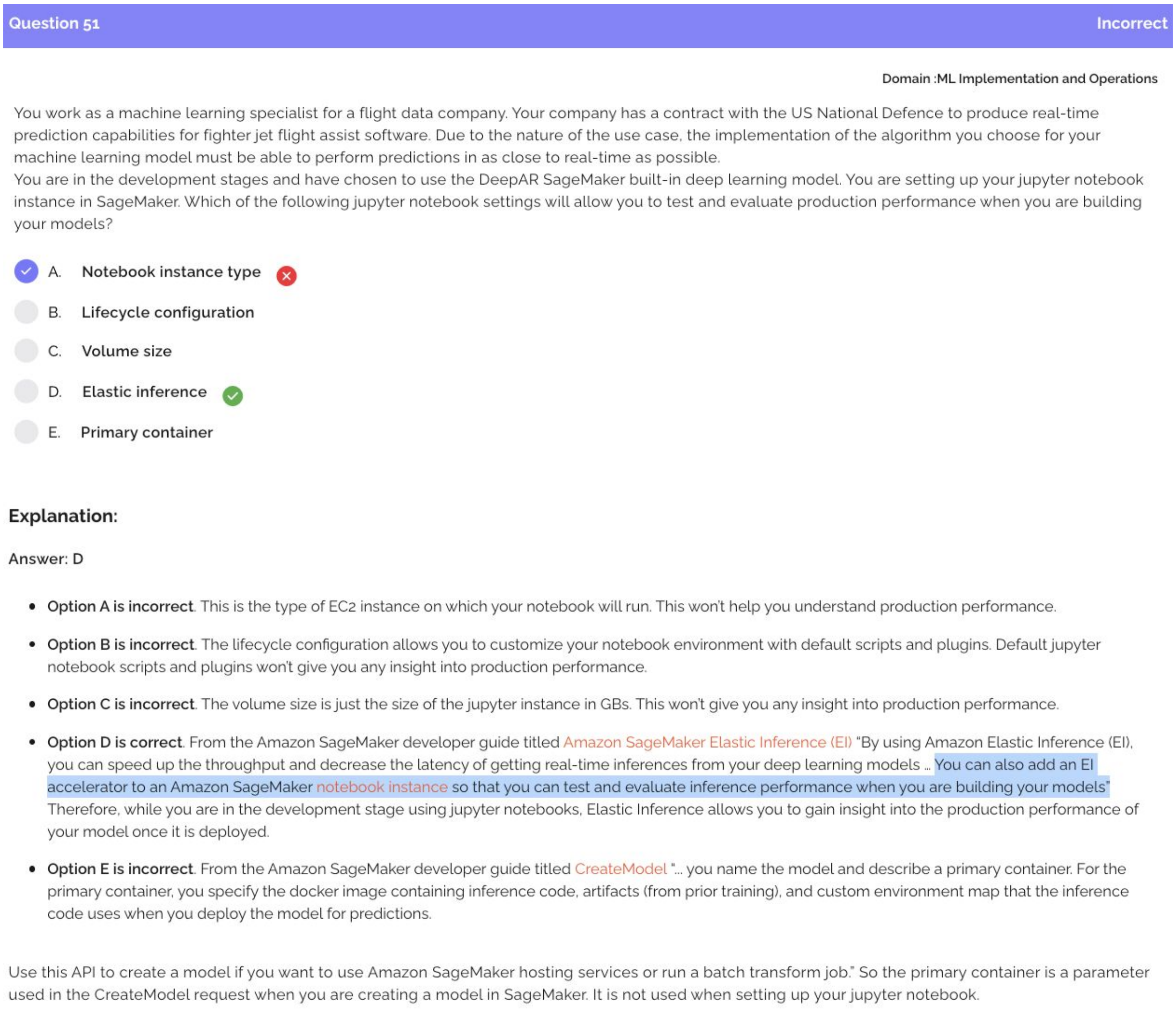
- □ A. alpha
- □ B. base_score
- □ C. eta
- □ D. num_round ✓
- ☑ E. gamma ✗
- ☑ F. num_class ✓

Explanation:

Answers: D and F

- **Option A is incorrect**. The alpha hyperparameter is used to adjust the L1 regulation term on weights. This term is optional. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)
- **Option B is incorrect**. The base_score hyperparameter is used to set the initial prediction score of all instances. This term is optional. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)
- **Option C is incorrect**. The eta hyperparameter is used to prevent overfitting. This term is optional. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)
- **Option D is correct**. The num_round hyperparameter is used to set the number of rounds to run in your hyperparameter tuning jobs. This term is required. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)
- **Option E is incorrect**. The gamma hyperparameter is used to set the minimum loss reduction required to make a further partition on a leaf node of the tree. This term is optional. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)
- **Option F is correct**. This hyperparameter is used to set the number of classes. This term is required if the objective is set to multi:softmax or multi:softprob. (See the Amazon SageMaker developer guide titled XGBoost Hyperparameters)

Using EI in a notebook to evaluate production performance



Question 51 **Incorrect**

Domain :ML Implementation and Operations

You work as a machine learning specialist for a flight data company. Your company has a contract with the US National Defence to produce real-time prediction capabilities for fighter jet flight assist software. Due to the nature of the use case, the implementation of the algorithm you choose for your machine learning model must be able to perform predictions in as close to real-time as possible.
You are in the development stages and have chosen to use the DeepAR SageMaker built-in deep learning model. You are setting up your jupyter notebook instance in SageMaker. Which of the following jupyter notebook settings will allow you to test and evaluate production performance when you are building your models?

- ● A. Notebook instance type ✗
- ○ B. Lifecycle configuration
- ○ C. Volume size
- ○ D. Elastic inference ✓
- ○ E. Primary container

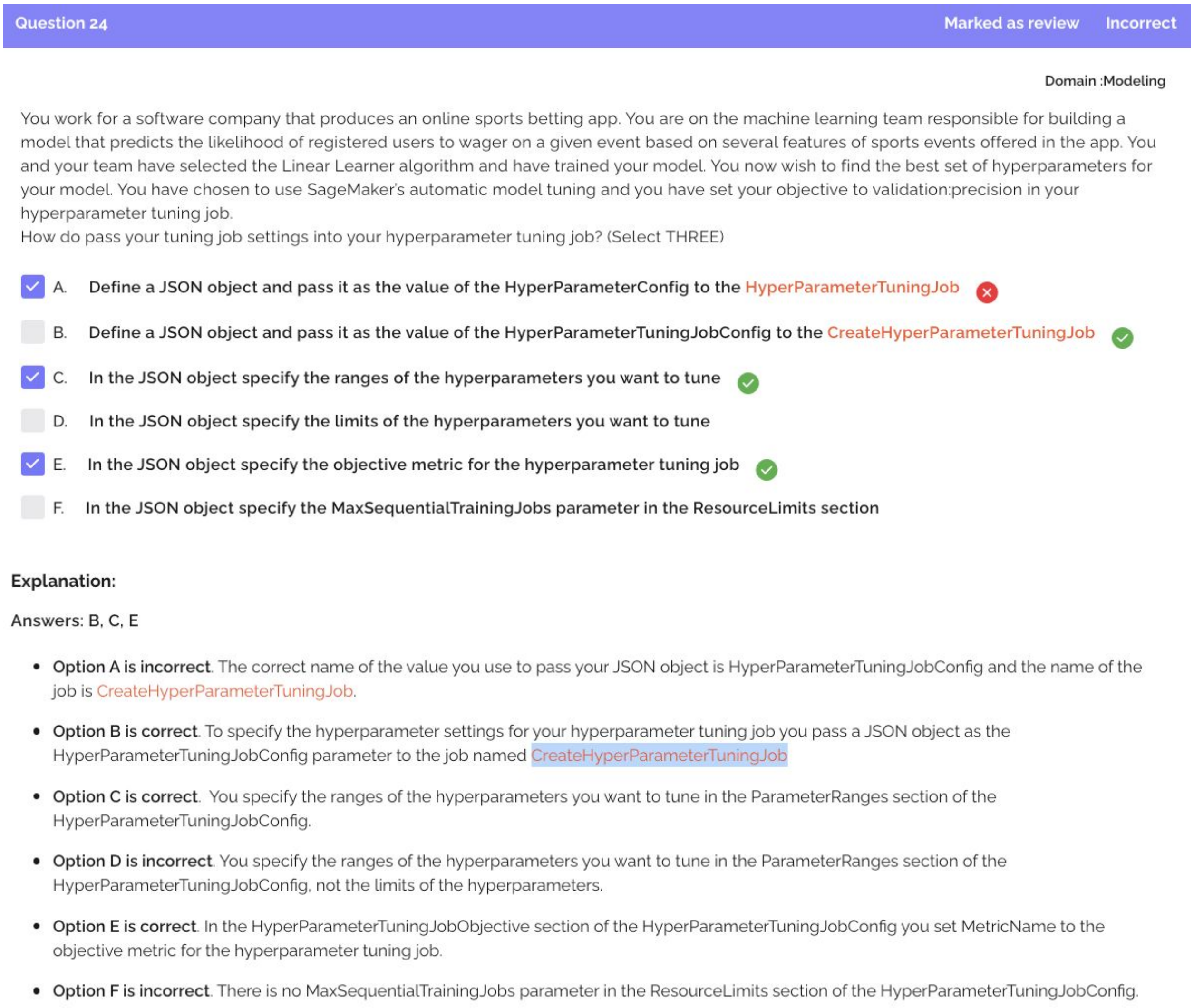
Explanation:

Answer: D

- **Option A is incorrect**. This is the type of EC2 instance on which your notebook will run. This won't help you understand production performance.
- **Option B is incorrect**. The lifecycle configuration allows you to customize your notebook environment with default scripts and plugins. Default jupyter notebook scripts and plugins won't give you any insight into production performance.
- **Option C is incorrect**. The volume size is just the size of the jupyter instance in GBs. This won't give you any insight into production performance.
- **Option D is correct**. From the Amazon SageMaker developer guide titled Amazon SageMaker Elastic Inference (EI) "By using Amazon Elastic Inference (EI), you can speed up the throughput and decrease the latency of getting real-time inferences from your deep learning models ... You can also add an EI accelerator to an Amazon SageMaker notebook instance so that you can test and evaluate inference performance when you are building your models" Therefore, while you are in the development stage using jupyter notebooks, Elastic Inference allows you to gain insight into the production performance of your model once it is deployed.
- **Option E is incorrect**. From the Amazon SageMaker developer guide titled CreateModel "... you name the model and describe a primary container. For the primary container, you specify the docker image containing inference code, artifacts (from prior training), and custom environment map that the inference code uses when you deploy the model for predictions.

Use this API to create a model if you want to use Amazon SageMaker hosting services or run a batch transform job." So the primary container is a parameter used in the CreateModel request when you are creating a model in SageMaker. It is not used when setting up your jupyter notebook.

API for Hyper parameter: CreateHyperParameterTuningJob ()



Question 24 Marked as review Incorrect

Domain :Modeling

You work for a software company that produces an online sports betting app. You are on the machine learning team responsible for building a model that predicts the likelihood of registered users to wager on a given event based on several features of sports events offered in the app. You and your team have selected the Linear Learner algorithm and have trained your model. You now wish to find the best set of hyperparameters for your model. You have chosen to use SageMaker's automatic model tuning and you have set your objective to validation:precision in your hyperparameter tuning job.
How do pass your tuning job settings into your hyperparameter tuning job? (Select THREE)

☑ A. **Define a JSON object and pass it as the value of the HyperParameterConfig to the HyperParameterTuningJob** ✗

☐ B. **Define a JSON object and pass it as the value of the HyperParameterTuningJobConfig to the CreateHyperParameterTuningJob** ✓

☑ C. **In the JSON object specify the ranges of the hyperparameters you want to tune** ✓

☐ D. **In the JSON object specify the limits of the hyperparameters you want to tune**

☑ E. **In the JSON object specify the objective metric for the hyperparameter tuning job** ✓

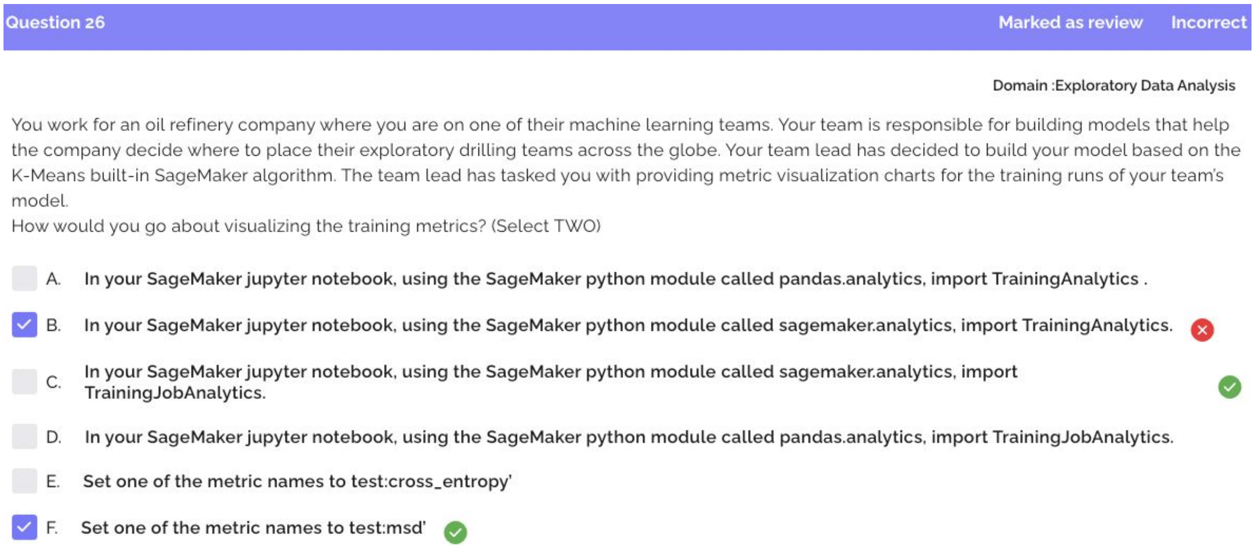
☐ F. **In the JSON object specify the MaxSequentialTrainingJobs parameter in the ResourceLimits section**

Explanation:

Answers: B, C, E

- **Option A is incorrect**. The correct name of the value you use to pass your JSON object is HyperParameterTuningJobConfig and the name of the job is CreateHyperParameterTuningJob.
- **Option B is correct**. To specify the hyperparameter settings for your hyperparameter tuning job you pass a JSON object as the HyperParameterTuningJobConfig parameter to the job named CreateHyperParameterTuningJob
- **Option C is correct**. You specify the ranges of the hyperparameters you want to tune in the ParameterRanges section of the HyperParameterTuningJobConfig.
- **Option D is incorrect**. You specify the ranges of the hyperparameters you want to tune in the ParameterRanges section of the HyperParameterTuningJobConfig, not the limits of the hyperparameters.
- **Option E is correct**. In the HyperParameterTuningJobObjective section of the HyperParameterTuningJobConfig you set MetricName to the objective metric for the hyperparameter tuning job.
- **Option F is incorrect**. There is no MaxSequentialTrainingJobs parameter in the ResourceLimits section of the HyperParameterTuningJobConfig.

Import TrainingJobAnalytics from SageMaker.analytics



Question 26 Marked as review Incorrect

Domain :Exploratory Data Analysis

You work for an oil refinery company where you are on one of their machine learning teams. Your team is responsible for building models that help the company decide where to place their exploratory drilling teams across the globe. Your team lead has decided to build your model based on the K-Means built-in SageMaker algorithm. The team lead has tasked you with providing metric visualization charts for the training runs of your team's model.
How would you go about visualizing the training metrics? (Select TWO)

- □ A. **In your SageMaker jupyter notebook, using the SageMaker python module called pandas.analytics, import TrainingAnalytics .**
- ☑ B. **In your SageMaker jupyter notebook, using the SageMaker python module called sagemaker.analytics, import TrainingAnalytics.** ✗
- □ C. **In your SageMaker jupyter notebook, using the SageMaker python module called sagemaker.analytics, import TrainingJobAnalytics.** ✓
- □ D. **In your SageMaker jupyter notebook, using the SageMaker python module called pandas.analytics, import TrainingJobAnalytics.**
- □ E. **Set one of the metric names to test:cross_entropy'**
- ☑ F. **Set one of the metric names to test:msd'** ✓

Kinesis Firehose with Lambda transformation - status and results
Lambda status:

- **Ok** (record was transformed successfully)
- **Dropped** (record was dropped intentionally by transformation logic)
- **ProcessingFailed** (record could not be transformed

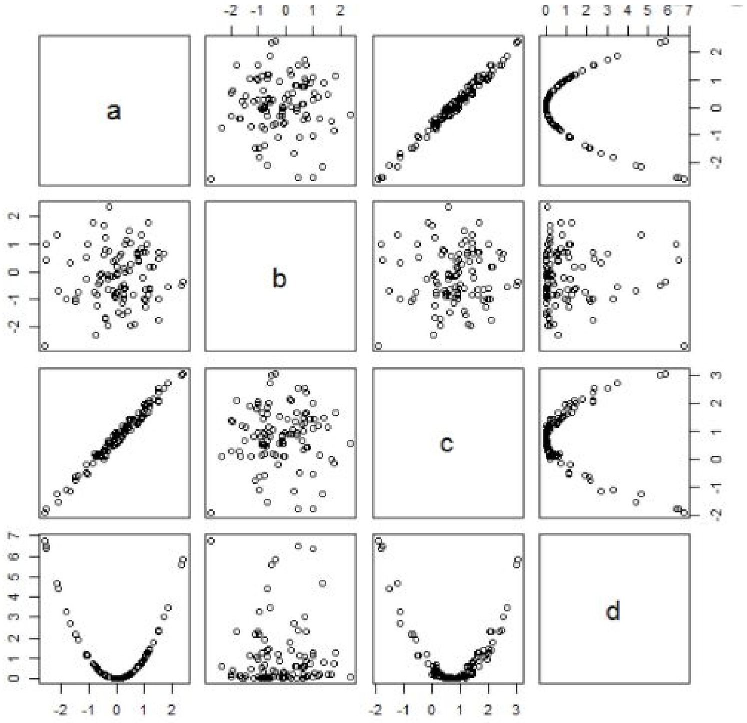
Firehose tranformed records:

- **recordId**
- **Result**
- **data**

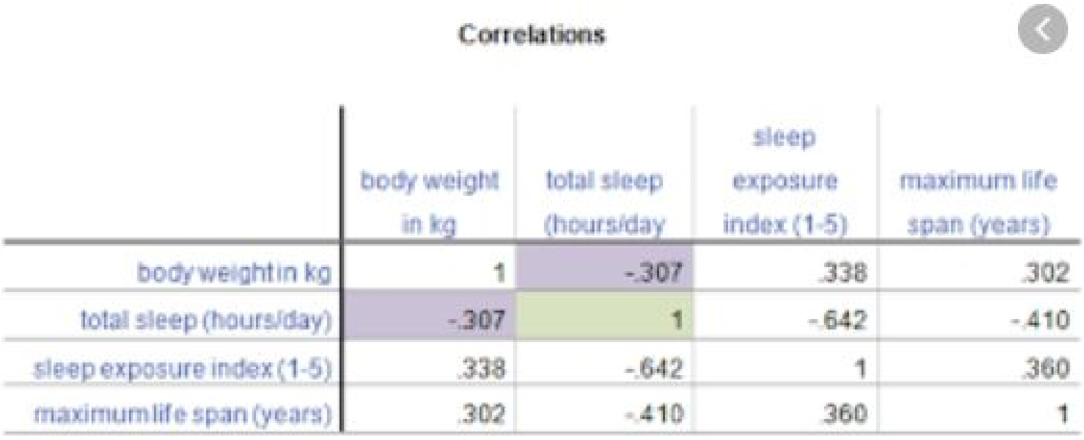
Covariance Matrix and Pairs Plot

- **Pairs plot:** show the relationship between pairs of features as well as the distribution of one of the variables in relation to the other.
- **Covariance matrix** shows the degree ot correlation between 2 features

Pairs plot:



Covariance matrix:



Correlations

	body weight in kg	total sleep (hours/day	sleep exposure index (1-5)	maximum life span (years)
body weight in kg	1	-.307	.338	.302
total sleep (hours/day)	-.307	1	-.642	-.410
sleep exposure index (1-5)	.338	-.642	1	.360
maximum life span (years)	.302	-.410	360	1