## Answer Sheet

**Q1)**

**A legacy MapReduce job is running on EMR, and must ingest data that has been moved from HDFS to an S3 data lake.**

**Which is a viable option for connecting the S3 data to MapReduce on EMR?**

● Use EFS to connect MapReduce to the S3 bucket.
● MapReduce can talk natively to S3 using the s3a:// prefix
✓ Use EMRFS to connect MapReduce to S3, using the s3:// file prefix.
**Explanation:-**EMR extends Hadoop (which includes MapReduce) to use S3 as a storage backend instead of HDFS, using EMRFS.
● Use Apache Hive as an intermediary between MapReduce and S3

---

**Q2)**

**Your deep neural network has achieved 99% accuracy on your training data, but only 90% on your test data. Human-level performance for the same task is around 98%.**

**What are some possible conclusions (SELECT TWO)?**

● The model is overfitting and failing to generalize. Using more layers may help.
✓ The model is overfitting and failing to generalize. Using dropout regularization may help.
**Explanation:-**You'll need to understand how to deal with overfitting through regularization. Dropout regularization and using fewer layers are two ways to prevent overfitting in a neural network. Shuffling the input data every epoch would also help.
● The model is underfitting, and broader hidden layers are needed.
✓ The model is overfitting and failing to generalize. Using fewer layers may help.
**Explanation:-**You'll need to understand how to deal with overfitting through regularization. Dropout regularization and using fewer layers are two ways to prevent overfitting in a neural network. Shuffling the input data every epoch would also help.
● The model is underfitting, and more epochs will help.

---

**Q3)**

**A media company wishes to recommend movies to users based on the predicted rating of that movie for each user, using SageMaker and Factorization Machines.**

**What format should the rating data used for training the model be in?**

● CSV
✓ RecordIO/ protobuf in float32 format
**Explanation:-**RecordIO is usually the best choice on models that support it, as it allows for efficient processing and the use of Pipe mode. Factorization Machines are unusual in that they expect float32 data, not integers.
● LibSVM
● RecordIO / protobuf in integer format

---

**Q4)**

**You are tasked with developing a machine learning system that can detect the presence of your company's logo in an image. You have a large training set of images that do and do not contain your logo, but they are unlabeled.**

**How might you prepare this data prior to training a supervised learning model with it, with the least amount of development effort?**

● Use Amazon Rekognition
● Use Amazon Mechanical Turk
● Use SageMaker Object Detection
✓ Use Amazon SageMaker Ground Truth
**Explanation:-**While Ground Truth can use the Mechanical Turk workforce as an option, it is purpose built for this sort of task and can be set up very quickly. Rekognition won't know about your company logo, nor will Object Detection until you have trained it first.

---

**Q5)**

**While training your deep neural network using accuracy as a loss function, its accuracy is evaluated against a validation set at each epoch. The accuracy against the validation set continues to increase with each epoch until the 100th epoch, at which point the accuracy against the validation set begins to decrease while accuracy on the training set continues to increase.**

**What is likely happening?**

● This is not an indication of a real problem and may be ignored.
● The model is beginning to underfit after 100 epochs. Adding more layers to the network will help.
● A different activation function needs to be used.
✓ The model is beginning to overfit after 100 epochs. Early stopping will help.
**Explanation:-**Early stopping is one of the most widely used forms of neural network regularization.

---

**Q6)**

**You've set up a camera in Los Angeles, and want to be notified when a known celebrity is spotted.**

**Which services, used together, could accomplish this with the least development effort?**

● SageMaker Object Detection, Lambda, and SNS
● SageMaker Semantic Segmentation, SQS, and SNS
✓ Amazon Rekognition, IAM, and SNS

**Q7)**

**A convention wishes to install cameras that automatically detect when conference attendees are seen wearing a specific company shirt, as part of a contest.**

**Which is a viable approach?**

● Send raw video feeds into Amazon Rekognition to detect the company shirts
● Use RNN's embedded within DeepLens to detect shirts at the edge
● Use DeepLens and the DeepLens_Kinesis_Video module to analyze video in real time using the ImageNet CNN.
✓ Use DeepLens and the DeepLens_Kinesis_Video module to send video streams to a CNN trained with SageMaker using a labeled training set of videos of the company shirts.
**Explanation:-**DeepLens does integrate with Kinesis Video Streams, which in turn integrates with SageMaker. However, a pre-trained model such as ImageNet or Rekognition won't know about these specific company shirts - you need to train your own model first. CNN's and not RNN's, are appropriate for object detection.

**Q8)**

**You've developed a custom training model for SageMaker using TensorFlow, but a single GPU can't handle the data it needs to train with.**

**How would you go about scaling your algorithm to use multiple GPU's within SageMaker?**

● This isn't possible with Tensorflow; use Apache MXNet instead.
✓ Write your model with the Horovod distributed training framework, which is supported by SageMaker.
**Explanation:-**This is an example of where studying the SageMaker developer guide isn't enough. Read through the latest news from AWS related to SageMaker as well; there are a few questions where the answers are only found on AWS blog posts.
● Deploy your model to multiple EC2 P3 instances, and SageMaker will distribute it automatically
● Wrap your Tensorflow code with PySpark, and use sagemaker-spark to distribute it.

**Q9)**

**Your machine learning system needs to be re-trained nightly on millions of records. The training involves multiple long-running ETL jobs which need to execute in order and complete successfully, prior to being handed off to a machine learning training workflow.**

**Which is the simplest tool to help manage this process?**

● Amazon Simple Workflow Service
✓ AWS Step Functions
**Explanation:-**AWS Step Functions are designed for this use case. Depending on the details, AWS Data Pipeline could also be an appropriate choice, but it wasn't listed as an option. Simple Workflow Service might also do the job, but it would require a more complex approach. Using SQS would require implementing substantial functionality on top of SQS, and AWS Batch is only designed for scheduling and allocating the resources needed for batch processing.
● Amazon SQS
● AWS Batch

**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

**Q11)**

**You are training a Linear Learner model in SageMaker, with normalization enabled. However, your training is failing.**

**What might be a probable cause?**

● Normalization should be disabled.
● The data was shuffled prior to training.
✓ The data was not shuffled prior to training.
**Explanation:-**Training with unshuffled data may cause training to fail. Training data should be normalized and shuffled. Linear Learner supports both classification and regression tasks.
● You are attempting to perform classification instead of regression.

**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

**Q13)**

**You are receiving real-time data streams of raw data containing hundreds of columns of data on each record. Many of these columns are not needed for the machine learning system you are developing, and some of the remaining columns need to be concatenated or transformed in minor ways.**

**What is the simplest and most storage-efficient way to transform this data as it is received?**

● Use a Glue ETL job on the streaming data to transform it after it is stored in S3.
● Accept all of the data into S3, and drop the unneeded columns as part of preparing data for training.
✓ Process the data using Kinesis Data Streams and Amazon Kinesis Data Analytics. Within Kinesis Data Analytics, transform the data using SQL commands prior to sending the processed data to your analytics tools.
**Explanation:**-While other approaches may work, only Kinesis Data Analytics strips out the unneeded data before it is even stored.
● Use the spark-sagemaker library to process the data prior to training.

**Q14)**

**You have a massive S3 data lake containing clickstream data, and you wish to analyze and visualize this data in order to better understand the cleaning and feature engineering it might require.**

**Which AWS services could be used to analyze and visualize this data, without provisioning individual servers in the process?**

✓ S3, Glue, Athena, and QuickSight
**Explanation:**-RDS, Elasticsearch, and EMR all require the provisioning of servers. S3, Glue, Athena, and Quicksight are all serverless solutions.
● S3, DMS, RDS
● S3, EMR, Quicksight
● S3, Kinesis, Amazon Elasticsearch

**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

**Q16) What is the F1 score of the confusion matrix below?**

● 0.36
● 0.6
● 0.67
✓ 0.73
**Explanation:**-F1 is given by (2 x Precision x Recall) / (Precision + Recall). Precision is TP / (TP+FP) and Recall is TP / (TP+FN). In this example, Precision is 8/10 and Recall is 8/12, which ultimately works out to a F1 score of 0.73.

**Q17)**

**An online retailer wishes to predict website traffic over the coming year on an hourly basis. This traffic fluctuates signficantly with both time of day and time of year.**

**Which service could produce these recommendations with the least amount of development and operational overhead?**

● SageMaker DeepAR
● LSTM model deployed to EC2 with a machine learning AMI
✓ Amazon Forecast
**Explanation:**-New AWS machine learning services are launching all the time, Forecast being a recent one. Be warned that the free training videos offered by AWS Training aren't always up to date on these newer offerings; you need to research what's available today.
● Apache Spark MLLib

**Q18)**

**You wish to categorize terabytes' worth of articles into topics using SageMaker and LDA, but processing that much data at once is leading to difficulties in storage and training the model reliably.**

**What can be done to improve the performance of the system?**

● Configure SageMaker to use multiple instances for training LDA
✓ Convert the articles to RecordIO format, and use Pipe mode
**Explanation:-**Pipe mode allows you to stream in data, instead of copying the entire dataset to every machine you are training on. For large data sets this can make a big difference. With LDA, pip mode is only supported with RecordIO format, and LDA only supports training on a single-instance CPU.
● Convert the articles to CSV format, and use Pipe mode
● Configure SageMaker to use multiple GPU's for training LDA

**Q19)**

**A recommender engine developed using SageMaker has been deployed using a custom inference model. You've tested an improvement to your model offline, but wish to expose it to production traffic to see how real people respond to it.**

**How might you deploy this new model in a way that minimizes risk and operational effort?**

● Write logic within your custom inference model to randomly assign traffic to one underlying model or the other, and ramp up the traffic to the new model over time.
● This is just a bad idea, and you should evaluate this model offline instead using k-fold validation.
✓ Deploy the new model as a production variant behind the existing SageMaker endpoint. Increase the amount of traffic to the new model over time via SageMaker.
**Explanation:-**SageMaker's "production variants" are made for this sort of thing. Recommender systems in particular tend to behave differently in production deployments than in offline testing, so this is absolutely a valid thing to do.
● Deploy the model behind a second SageMaker endpoint, and use a load balancer to ramp up the traffic to the new model over time.

**Q20) When using SageMaker's BlazingText algorithm in Word2Vec mode, which of the following statements are true?**

✓ The order of words doesn't matter, as it uses skip-gram and continuous bag of words (CBOW) architectures.
**Explanation:-**A "continuous bag of words" can be thought of as a jumble of words in a bag. Order doesn't matter. BlazingText doesn't use LSTM or CNN; those are just there to throw you off.
● The order of words does not matter, as it uses a CNN internally.
● The order of words matters, because it uses LSTM internally
● The order of words does matter, as it uses skip-gram and continuous bag of words (CBOW) architectures.

**Q21) You have created a SageMaker notebook instance using its default IAM role. How is access to data in S3 managed?**

● No buckets are available by default; you must edit the default IAM role to explicitly allow access.
● Only S3 buckets with public access enabled are accessible
✓ Any bucket with "sagemaker" in the name is accessible with the default role
**Explanation:-**Unless you add a policy with S3FullAccess permission to the role, it is restricted to buckets with "sagemaker" in the bucket name. Strange but true.
● The default IAM role allows access to any S3 bucket, regardless of name

**Q22)**

**A financial services company needs to automate the analysis of each day's transaction costs, execution reporting, and market performance. They have developed their own Big Data tools to perform this analysis, which require the scheduling and configuration of their underlying computing resources.**

**Which tool provides the simplest approach for configuring the resources and scheduling the data analytic workloads?**

● Amazon Simple Workflow Service
✓ AWS Batch
**Explanation:-**
This use-case comes straight from the AWS Batch homepage. See aws.amazon.com/batch/ for more.

● AWS Step Functions
● Amazon SQS

**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

**Q24) Which probability distribution would describe the likelihood of flipping a coin "heads"?**

● Normal Distribution
● Bernoulli Distribution
● Poisson Distribution
✓ Binomial Distribution
**Explanation:-**You could probably guess this one based on the name, but make sure you understand what all of these distributions are and how they are used. Some are specific to discrete time-series data, some are specific to continuous data, and they all describe different situations. A good

reference I found is https://www.analyticsvidhya.com/blog/2017/09/6-probability-distributions-data-science/

**Q25)**

**A large retail chain receives dumps of sales data from each store on a daily basis into S3. Analysts need to run daily reports covering trends over the past 30 days using this data. After 90 days, the data may be archived.**

**Which architecture allows for fast querying and archiving of this data, with the least cost?**

● Copy the data nightly into Redshift, with a table for each day. Run a nightly script to drop tables older than 90 days.
● Store all data into a single partition, and use Glue ETL to identify and discard data older than 90 days.
✓ Organize the data by prefixes that indicate the day the data covers, and use Glue to create tables partitioned by date. Use S3 lifecycle policies to automatically archive the data to Glacier after 90 days.
**Explanation:-**Organizing data by date using S3 prefixes allows Glue to partition the data by date, which leads to faster queries done on date ranges. S3 lifecycle policies can automate the process of archiving old data to Glacier.
● Organize the data into unique S3 buckets for each date, and use S3 lifecycle policies to archive the data after 90 days.

**Q26)**

**You are using SageMaker and XGBoost to classify millions of videos into genres, based on each video's attributes. Prior to training your model, the video attribute data must be cleaned and transformed into LibSVM format.**

**Which are viable approaches for pre-processing the data? (SELECT TWO)**

✓ Use PySpark with the XGBoostSageMakerEstimator to prepare the data using Spark, and then pass off the training to SageMaker.
**Explanation:-**For data of this size, you want to parallelize its processing. And that's something Apache Spark is good for. The sagemaker-spark package allows you to integrate SageMaker and Spark together, and you can also just use the two systems separately. Neither Glue ETL nor Kinesis Analytics can convert to LibSVM format, and scikit-learn is not a distributed solution.
● Use Kinesis Analytics to transform the data as it is received into LibSVM format, then train with SageMaker.
● Use Glue ETL to transform the data into LibSVM format, and then train with SageMaker.
● Use scikit-learn in your SageMaker notebook to pre-process the data, and then train it.
✓ Use Spark on EMR to pre-process the data, and store the processed results in an S3 bucket accessible to SageMaker.
**Explanation:-**For data of this size, you want to parallelize its processing. And that's something Apache Spark is good for. The sagemaker-spark package allows you to integrate SageMaker and Spark together, and you can also just use the two systems separately. Neither Glue ETL nor Kinesis Analytics can convert to LibSVM format, and scikit-learn is not a distributed solution.

**Q27) The graph below plots predicted and actual website views over time. Based on this graph, would you say the prediction model:**

● Captures seasonality and trends well
✓ Captures seasonality and trends poorly
**Explanation:-**Seasonality refers to periodic changes, while trends are longer-term changes over time. A trend across seasonal data would result in periodic seasonal spikes and valleys increasing or decreasing over time.
● Captures seasonality well, but trends poorly
● Captures seasonality poorly, and trends well

**Q28)**

**Your XGBoost model has high accuracy on its training set, but poor accuracy on its validation set, suggesting overfitting.**

**Which hyperparameter would be most likely to improve the situation?**

● csv_weights
✓ subsample
**Explanation:-**Only the "subsample" parameter directly addresses overfitting out of these choices, but other parameters such as eta, gamma, lambda, and alpha may also have an effect. Refer to https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost_hyperparameters.html - yes, you will be expected to have this level of detail on a few questions.
● grow_policy
● booster

**Q29)**

**You want to create AI-generated music, by training some sort of neural network on existing music and getting it to predict additional notes going forward.**

**What architecture might be appropriate?**

✓ RNN
**Explanation:-**Music is fundamentally a time-series problem, which RNN's (recurrent neural networks) are best suited for. You might see the term LSTM used as well, which is a specific kind of RNN.
● MLP
● CNN
● ResNet50

**Q30) The ROC curve below was generated from a classifier. What can we say about this classifier?**

● This model has perfect accuracy
✓ This model has no discrimination capacity to distinguish between positive and negative classes.
**Explanation:-**This graph presents a classifier that is no better than random chance. A good ROC curve would be curved up toward (0,1) and not

linear like this. The AUC (area under the curve) in this case is actually 0.5; a perfect AUC would be 1.0.

● The AUC is 1.0<br>
● This model has a perfect measure of separability

---

**Q31)**

**You increased the learning rate on your deep neural network in order to speed up its convergence, but suddenly the accuracy of the model has suffered as a result.**

**What is a likely cause?**

● Too many layers are being used

✓ The true minimum of your loss function was overshot while learning

**Explanation:-**A learning rate that is too large may overshoot the true minima, while a learning rate that is too small will slow down convergence.

● Shuffling should not be used

● SGD got stuck in local minima

---

**Q32)**

**A large retailer wishes to publish revenue forecasts in a graphical manner for consumption by executives. Historical revenue data contains anomalies, such as spikes due to price drops.**

**What is the simplest solution for providing these forecasts with the least amount of development effort and ongoing maintenance?**

● Publish sales data into S3, use SageMaker to produce forecasts, and visualize with QuickSight

● Publish sales data into an RDS database, and produce forecasts and visualizations using Tableau

● Publish sales data into an EMR cluster, produce forecasts with Spark, and visualize with QuickSight

✓ Publish sales data into S3, and produce forecasts and visualizations with Amazon QuickSight

**Explanation:-**QuickSight's ML Insights feature allows forecasting using QuickSight itself. This is a serverless solution that contains the least number of components.

---

**Q33)**

**A medical company is building a model to predict the occurrence of thyroid cancer. The training data contains 900 negative instances (people who don't have cancer) and 100 positive instances. The resulting model has 90% accuracy, but extremely poor recall.**

**What steps can be used to improve the model's performance? (SELECT TWO)**

● Over-sample instances from the negative (no cancer) class

✓ Generate synthetic samples using SMOTE

**Explanation:-**The fundamental issue is an imbalanced training set; there are too many negative samples and not enough positive ones. If you can collect more positive samples to improve the balance, that's the best option. Synthetic samples may also be created using SMOTE. Over-samping negatives or under-sampling positives just makes matters worse, although the opposite approaches would be reasonable. Bagging has nothing to do with it!

✓ Collect more data for the positive case

**Explanation:-**The fundamental issue is an imbalanced training set; there are too many negative samples and not enough positive ones. If you can collect more positive samples to improve the balance, that's the best option. Synthetic samples may also be created using SMOTE. Over-samping negatives or under-sampling positives just makes matters worse, although the opposite approaches would be reasonable. Bagging has nothing to do with it!

● Under-sample instances from the positive (has cancer) class

● Use Bagging

---

**Q34) A classifier that predicts if an image is of a cat or a dog results in the confusion matrix below. What is the precision of this classifier?**

● 0.7

● 0.75

● 0.67

✓ 0.8

**Explanation:-**Precision is defined as TP / (TP + FP). A confusion matrix has TP and FP on the top row, and FN and TN on the bottom. Refer: https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/Rekognition%20Custom%20Labels.pdf

---

**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.

---

**Q36)**

**Your training data contains hundreds of features, many of which are correlated. You are having difficulty converging on a useful model due to the sparsity caused by hundreds of dimensions.**

**How might you best pre-process this data to avoid this "curse of dimensionality?"**

● Drop half of the feature columns
✅ Apply PCA to the training data
**Explanation:-**PCA is a powerful dimensionality reduction technique that will find the best dimensions to arrange your data by. Dropping or concatenating columns would also reduce dimensionality, but in a haphazard manner. Factorization machines are relevant to handling sparse data, but they don't perform dimensionality reduction per se.
● Apply a factorization machine to the training data
● Concatenate columns together

**Q37)**

**You are ingesting a data feed of subway ridership in near-real-time. Your incoming data is timestamped by the minute, and includes the total number of riders at each station for that minute.**

**What is the simplest approach for automatically sending alerts when an unusually high or low number of riders is observed?**

● Ingest the data with Kinesis Firehose, and use Amazon CloudWatch to alert when anomalous data is detected.
✅ Ingest the data with Kinesis Data Firehose, and use Random Cut Forest in Kinesis Data Analytics to detect anomalies. Use AWS Lambda to process the output from Kinesis Data Analytics, and issue an alert via SNS if needed.
**Explanation:-**Random Cut Forest is Amazon's own algorithm for anomaly detection, and is usually the right choice when anomaly detection is asked for on the exam. It is implemented within both Kinesis Data Analytics and SageMaker, but only Kinesis works in the way described.
● Ingest the data with Kinesis Data Streams directly into S3, and use Random Cut Forest in SageMaker to detect anomalies in real-time. Integrate SageMaker with SNS to issue alarms.
● Publish data directly into S3, and use Glue to detect anomalies and pass on alerts to SNS.

**Q38) If you wanted to build your own Alexa-type device that converses with customers using speech, which Amazon services might you use?**

● Amazon Transcribe -> Amazon Comprehend -> Amazon Polly
● Amazon Comprehend -> Amazon Lex -> Amazon Polly
✅ Amazon Transcribe -> Amazon Lex -> Amazon Polly
**Explanation:-**Transcribe can convert a customer's speech to text, which could then be fed into Lex for handling the chatbot logic. The output from Lex could be read back to the customer using Polly. Under the hood, more services would likely be needed as well to support Lex, such as Lambda and DynamoDB.
● Amazon Polly -> Amazon Lex -> Amazon Transcribe

**Q39)**

**A dataset representing a clinical trial includes many features, including Mean Arterial Pressure (MAP). The various features are not well correlated, and less than 1% of the data is missing MAP information. Apart from some outliers, the MAP data is fairly evenly distributed. All other features contain complete information.**

**Which is the best choice for handling this missing data?**

● Populate the missing MAP values with random noise
✅ Impute the missing values with the median MAP value.
**Explanation:-**A rough imputation method such as mean or median can be a resonable choice when only a handful of values are missing, and there aren't large relationships between features that we might compromise. Due to the outliers mentioned, median is a better choice than mean.
● Impute the missing values with the mean MAP value.
● Drop the MAP column

**Q40)**

**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.

**Q41)**

**A system designed to classify financial transactions into fraudulent and non-fraudulent transactions results in the confusion matrix below.**

**What is the recall of this model?**

● 0.5
● 0.74
● 0.6667
✓ 0.9
**Explanation:-**Recall is defined as true positives / (true positives + false negatives). This works out to 90/(90+10) in this example, or 90%. 66.67% is the precision (true positives / (true postives + false positives.) Recall is an important metric in situations where classifications are highly imbalanced, and the positive case is rare. Accuracy tends to be misleading in these cases.

---

**Q42)**

**You are developing a computer vision system that can classify every pixel in an image based on its image type, such as people, buildings, roadways, signs, and vehicles.**

**Which SageMaker algorithm would provide you with the best starting point for this problem?**

● Object Detection
✓ Semantic Segmentation
**Explanation:-**Semantic Segmentation produces segmentation masks that identify classifications for each individual pixel in an image. It uses MXNet and the ResNet architecture to do this.
● Rekognition
● Object2Vec

---

**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

---

**Q44)**

**You are training an XGBoost model on SageMaker with millions of rows of training data, and you wish to use Apache Spark to pre-process this data at scale.**

**What is the simplest architecture that achieves this?**

● Use Amazon EMR to pre-process your data using Spark, and use the same EMR instances to host your SageMaker notebook.
● Use Sparkmagic to pre-process your data within a SageMaker notebook, transform the resulting Spark DataFrames into RecordIO format, and then use Spark's XGBoost algorithm to train the model.
✓ Use sagemaker_pyspark and XGBoostSageMakerEstimator to use Spark to pre-process, train, and host your model using Spark on SageMaker.
**Explanation:-**The SageMakerEstimator classes allow tight integration between Spark and SageMaker for several models including XGBoost, and offers the simplest solution. You can't deploy SageMaker to an EMR cluster, and XGBoost actually requires LibSVM or CSV input, not RecordIO.
● Use Amazon EMR to pre-process your data using Spark, and then use AWS Data Pipelines to transfer the processed training data to SageMaker

---

**Q45)**

**You are developing an autonomous vehicle that must classify images of street signs with extremely low latency, processing thousands of images per second.**

**What AWS-based architecture would best meet this need?**

● Use Amazon Rekognition in edge mode
● Use Amazon Rekognition on AWS DeepLens to identify specific street signs in a self-contained manner.
● Develop your classifier using SageMaker Object Detection, and use Elastic Inference to accelerate the model's endpoints called over the air from the vehicle.
✓ Develop your classifier with TensorFlow, and compile it for an NVIDIA Jetson edge device using SageMaker Neo, and run it on the edge with IoT GreenGrass.
**Explanation:-**SageMaker Neo is designed for compiling models using TensorFlow and other frameworks to edge devices such as Nvidia Jetson. The low latency requirement requires an edge solution, where the classification is being done within the vehicle itself and not over the air. Rekognition (which doesn't have an "edge mode," but does integrate with DeepLens) can't handle the very specific classification task of identifying different street signs and what they mean.

---

**Q46)**

**Your automatic hyperparameter tuning job in SageMaker is consuming more resources than you would like, and coming at a high cost.**

**What are TWO techniques that might reduce this cost?**

✓ Use less concurrency while tuning
**Explanation:-**Since the tuning process learns from each incremental step, too much concurrency can actually hinder that learning. Logarithmic ranges tend to find optimal values more quickly than linear ranges. Inference pipelines are a thing, but have nothing to do with this problem.

● Use more concurrency while tuning

● Use inference pipelines

✓ Use logarithmic scales on your parameter ranges

**Explanation:-**Since the tuning process learns from each incremental step, too much concurrency can actually hinder that learning. Logarithmic ranges tend to find optimal values more quickly than linear ranges. Inference pipelines are a thing, but have nothing to do with this problem.

● Use linear scales on your parameter ranges

**Q47)**

**Your company wishes to monitor social media, and perform sentiment analysis on Tweets to classify them as positive or negative sentiment. You are able to obtain a data set of past Tweets about your company to use as training data for a machine learning system, but they are not classified as positive or negative.**

**How would you build such a system?**

✓ Use SageMaker Ground Truth to label past Tweets as positive or negative, and use those labels to train a neural network on SageMaker.

**Explanation:-**A machine learning system needs labeled data to train itself with; there's no getting around that. Only the Ground Truth answer produces the positive or negative labels we need, by using humans to create that training data initially. Another solution would be to use natural language processing through a service such as Amazon Comprehend.

● Stream both old and new tweets into an Amazon Elasticsearch Service cluster, and use Elasticsearch machine learning to classify the tweets.

● Use Amazon Machine Learning with a binary classifier to assign positive or negative sentiments to the past Tweets, and use those labels to train a neural network on an EMR cluster.

● Use RANDOM_CUT_FOREST to automatically identify negative tweets as outliers.

**Q48)**

**A large news website needs to produce personalized recommendations for articles to its readers, by training a machine learning model on a daily basis using historical click data. The influx of this data is fairly constant, except during major elections when traffic to the site spikes considerably.**

**Which system would provide the most cost-effective and reliable solution?**

✓ Publish click data into Amazon S3 using Kinesis Firehose, and process the data nightly using Apache Spark and MLLib using spot instances in an EMR cluster. Publish the model's results to DynamoDB for producing recommendations in real-time.

**Explanation:-**The use of spot instances in response to anticipated surges in usage is the most cost-effective approach for scaling up an EMR cluster. Kinesis streams is over-engineering because we do not have a real-time streaming requirement. Elasticsearch doesn't make sense because Elasticsearch is not a recommender engine. reference - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html

● Publish click data into Amazon Elasticsearch using Kinesis Firehose, and query the Elasticsearch data to produce recommendations in real-time.

● Publish click data into Amazon S3 using Kinesis Firehose, and process the data nightly using Apache Spark and MLLib using reserved instances in an EMR cluster. Publish the model's results to DynamoDB for producing recommendations in real-time.

● Publish click data into Amazon S3 using Kinesis Streams, and process the data in real time using Splunk on an EMR cluster with spot instances added as needed. Publish the model's results to DynamoDB for producing recommendations in real-time.

**Q49)**

**After training a deep neural network over 100 epochs, it achieved high accuracy on your training data, but lower accuracy on your test data, suggesting the resulting model is overfitting.**

**What are TWO techniques that may help resolve this problem?**

✓ Use early stopping

**Explanation:-**Early stopping is a simple technique for preventing neural networks from training too far, and learning patterns in the training data that can't be generalized. Dropout regularization forces the learning to be spread out amongst the artificial neurons, further preventing overfitting. Removing layers, rather than adding them, might also help prevent an overly complex model from being created - as would using fewer features, not more.

✓ Use dropout regularization

**Explanation:-**Early stopping is a simple technique for preventing neural networks from training too far, and learning patterns in the training data that can't be generalized. Dropout regularization forces the learning to be spread out amongst the artificial neurons, further preventing overfitting. Removing layers, rather than adding them, might also help prevent an overly complex model from being created - as would using fewer features, not more.

● Use more features in the training data

● Employ gradient checking

● Use more layers in the network

**Q50)**

**You are ingesting images and text from a social media feed, and wish to label them with the subjects represented by each image or text post.**

**Which services might play a role in this system? (SELECT TWO)**

✓ Amazon Comprehend

**Explanation:-**Rekognition can identify common objects in images right out of the box. Comprehend could be used to produce topics for the text in the posts. The remaining choices are relevant to speech and chatbots, not this task.

✓ Amazon Rekognition

**Explanation:-**Rekognition can identify common objects in images right out of the box. Comprehend could be used to produce topics for the text in the posts. The remaining choices are relevant to speech and chatbots, not this task.

● Amazon Transcribe

● Amazon Lex

● Amazon Polly

**Q51)**

**A random forest classifier was used to classify handwriten digits 0-9 into the numbers they were intended to represent. The confusion matrix below was generated from the results.**

**Based on the matrix, which number was predicted with the least accuracy?**

● zero<br>
✓ 8<br>
**Explanation:-**The choice with the lightest color along the diagonal axis is the correct one, as it represents the lowest number of correct predictions. You should understand how to read and interpret confusion matrices in depth for the exam.<br>
● 3<br>
● 6

**Q52)**

**A company maintains a large data lake in Amazon S3 containing a combination of structured and unstructured CSV data. Some of this data must be transformed and cleaned as it is received, and analysts within the company wish to analyze it using SQL queries.**

**What solution would require the least amount of development work, lowest cost, and least ongoing maintenance?**

● Transform the data using Apache Spark on EMR, and query it with RDS<br>
● Tranform the data periodically with SageMaker, and query it with Redshift.<br>
● Use AWS Glue to transform the data, and query it using Redshift.<br>
✓ Use AWS Glue to organize the unstructured data and transform it, and Amazon Athena to query the data.<br>
**Explanation:-**Glue and Glue ETL can impart structure to unstructured data, and perform transformations on that data as it is received. Athena is a serverless solution that can query S3 data lakes directly when paired with Glue. Redshift, Aurora, and RDS are all more complex and expensive solutions - and both Spark/EMR and SageMaker require provisioning servers of your own.

**Q53)**

**You are deploying your own custom inference container on Amazon SageMaker.**

**Which of the following are requirements of your container? (SELECT TWO)**

● Respond to both /invocations and /ping on port 80<br>
✓ Respond to both /invocations and /ping on port 8080<br>
**Explanation:-**Your inference container responds to port 8080, and must respond to ping requests in under 2 seconds. Model artifacts need to be compressed in tar format, not zip.<br>
✓ Accept all socket connection requests within 250 ms.<br>
**Explanation:-**Your inference container responds to port 8080, and must respond to ping requests in under 2 seconds. Model artifacts need to be compressed in tar format, not zip.<br>
● Respond to GET requests on the /ping endpoint in under 5 seconds.<br>
● Must be compressed in ZIP format

**Q54)**

**An e-commerce company needs to pre-process large amounts of consumer behavior data stored in HDFS using Apache Spark on EMR prior to analysis on a daily basis. The volume of data is highly seasonal, and can surge during holidays and big sales.**

**What is the most cost-effective option for handling these sporadic demands, without incurring data loss or needing to terminate the entire cluster?**

● Use EC2 Spot instances for core and task nodes, and reserved instances for the master node<br>
● Use reserved instances for task nodes, and spot instances for core nodes<br>
✓ Use EC2 Spot instances for Spark task nodes only.<br>
**Explanation:-**While you can use Spot instances on any node type, a Spot interruption on the Master node requires terminating the entire cluster, and on a Core node, it can lead to HDFS data loss.<br>
● Use EC2 spot instances for all node types.

**Q55)**

**You are training a regression model using SageMaker's Linear Learner, to predict individual incomes as a function of age and years in school. The training data was gathered from several distinct groups.**

**What pre-processing should be performed to ensure good results? (SELECT TWO)**

✓ Normalize the feature data to have a mean of zero and unit standard deviation<br>
**Explanation:-**Since age and number of years in school are values that span very different ranges, they must be normalized prior to training with Linear Learner. Shuffling is also recommended with Linear Learner.<br>
● Use SMOTE to impute additional data<br>
● Add some random noise to the training data<br>
✓ Shuffle the input data<br>
**Explanation:-**Since age and number of years in school are values that span very different ranges, they must be normalized prior to training with Linear Learner. Shuffling is also recommended with Linear Learner.<br>
● Scale the feature data to match the range of income data

**Q56) What is an appropriate choice of an instance type for training XGBoost in SageMaker?**

● C4
✓ M4
**Explanation:-**XGBoost is a CPU-only algorithm, and won't benefit from the GPU's of a P3 or P2. It is also memory-bound, making M4 a better choice than C4.
● P2
● P3

---

**Q57)**

**An image recognition model using a CNN is capable of identifying flowers in an image, but you need an image recognition model that can identify specific species of flowers as well.**

**How might you accomplish this effectively while minimizing training time?**

● Use transfer learning by training the entire model with new labels
● Use incremental training on Amazon Rekognition
● Train a new CNN from scratch with only your flower species labels
✓ Use transfer learning by training a new classification layer on top of the existing model
**Explanation:-**Transfer learning generally involves using an existing model, or adding additional layers on top of one. Retraining the whole thing isn't transfer learning, and incremental training isn't something Rekognition supports (using incremental training with Sagemaker's image classification model would be a valid approach, though.)

---

**Q58)**

**You are developing a machine translation model using SageMaker's seq2seq model.**

**What format must your training data be provided in?**

● RecordIO-protobuf with floating point tokens
✓ RecordIO-protobuf format with integer tokens
**Explanation:-**For machine translation you first need to tokenize your words into integers, which refer to vocabulary files you also must provide. The seq2seq example notebook contains a script to convert data to the required format.
● Text in CSV format
● Text in JSON format

---

**Q59)**

**A classifier predicts if insurance claims are fraudulent or not. The cost of paying a fraudulent claim is higher than the cost of investigating a claim that is suspected to be fraudulent.**

**Which metric should we use to evaluate this classifier?**

● Specificity
● F1
● Precision
✓ Recall
**Explanation:-**Recall (TP / (TP+FN)) is important when the cost of a false negative is higher than that of a false positive. Be sure to have a deep understanding of precision, recall, and F1 for the exam, how to compute them, and when to use them. Remember a "positive" result is whatever the classifier is trying to predict, good or bad. In this case, "positive" means "is fraudulent."

---

**Q60)**

**A training dataset contains several columns of features from census data, all of which are correlated in some way. One column, representing a person's age, is missing 10% of its values.**

**What is the best way to handle these missing values to maximize accuracy in the resulting model?**

● Drop the column that contains missing values
✓ Build a separate supervised model that predicts the value of the missing column based on the other columns, and use that to estimate the missing values.
**Explanation:-**A machine learning model could capture the relationships between the features, allowing you to impute missing values more accurately than via other means.
● Populate the missing values with random values
● Impute the missing values based on the mean of the entire column

---

**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

**Q62)**

**A regression model on a dataset including many features includes L1 regularization, and the resulting model appears to be underfitting.**

**Which steps might lead to better accuracy? (SELECT TWO)**

✓ Decrease the L1 regression term

**Explanation:-**L1 effectively removes features that are unimportant, and doing this too aggressively can lead to underfitting. L2 weighs each feature instead of removing them entirely, which can lead to better accuracy. Removing more features would only make underfitting worse, and L0 regulaization isn't a thing.

✓ Try L2 instead of L1 regularization

**Explanation:-**L1 effectively removes features that are unimportant, and doing this too aggressively can lead to underfitting. L2 weighs each feature instead of removing them entirely, which can lead to better accuracy. Removing more features would only make underfitting worse, and L0 regulaization isn't a thing.

● Increase the L1 regularization term

● Use L0 regularization

● Remove features

**Q63)**

**You decreased the batch size used to train your deep neural network, and found that the accuracy of the model suddenly suffered as a result.**

**What is a likely cause?**

● Shuffling should not be used

✓ The small batch size caused training to get stuck in local minima

**Explanation:-**Large batch sizes lead to faster training, and small batch sizes run the risk of getting stuck in localized minima instead of finding the true one.

● Too many layers are being used

● The small batch size caused training to overshoot the true minima

**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

**Q65)**

**You are developing a deep learning model that categorizes handwritten digits 0-9 into the numbers they represent.**

**How should you pre-process the label data?**

● Normalization

✓ One-hot encoding

**Explanation:-**Categorical features need to be converted into one-hot, binary representations prior to use in a neural network.

● Hexadecimal

● Use integer values