**AWS Certified Machine Learning** - **Specialty Practice Questions**

**Requirement**: Create & Share 10 ML Specialty practice questions. These are for “**Section Tests**” to be placed below the existing practice:

**Topic**

* Questions around core machine learning concepts such as Pearson coefficient and Bayesian classification.

**Delivery Timeline**: July 6

**Question Response Types**

There are two types of questions:

* Multiple Choice Single Response – **1** correct answer **3** incorrect responses (distractors).
* Multiple Choice Multiple Response – **2** or more correct answers out of **5** or more options.

**Important Note**

* Do write Question Number for quick identification. Q# 1, Q# 2 …. & so on.
* Please mention Domain (based on ML Specialty exam blueprint), Topic & Sub-Topic (If Applicable) with every question.
* Note that we’re expecting standard scenario based questions & NOT straight-forward definition kind of questions.
* The options should not have any obviously incorrect option. We need to word the options so that all of them should appear correct for the students, but a subtle point should mark the correct answer without any ambiguity. So, one answer should be the best choice without any doubt.
* The answer / explanation section should contain explanations on why the answer is correct and others are incorrect. It should also contain the relevant resource link (for details) preferably from AWS documentation.
  + Example
    - Option A is incorrect because..
    - Option B is CORRECT because…
    - Option C is incorrect because..
    - Option D is incorrect because..
* Try to balance the domains based on weightage % defined in the exam blueprint.
* Any AWS service or feature must be approximately 6 months old to figure out in Practice Tests. Put a note in the explanation for any latest service or feature that might be on the borderline of appearing in the real exam.
* **Plagiarism** in any form - Question or in Explanation will be **rejected.** Questions & Explanations should reflect your own professional experience & AWS skills. Author’s who indulge in plagiarism will be **blacklisted** & dropped from our author’s list.
* The ownership of the questions once approved & published on Whizlabs LMS platform, lies solely with Whizlabs Software Pvt. Ltd. You can’t share or publish it elsewhere in any circumstances.

**Sample Format of** **Questions**

----------------------------------------------------------------------------------------------------------------------------

**Question​ ​:​** #

**Main​ ​Topic​ ​:​** < >

**Sub​ ​Topic​ ​:​** [optional]

**Domain:** < >

**Question text**:

<Scenario based. Should be clear in terms of requirements. No ambiguity. No duplicate options. In case of multiple answers, at the end, you should include number of expected answers. e.g. (Select TWO) or (Select THREE) etc. For single answers this is NOT required>

1. Option A...
2. Option B...
3. Option C...
4. Option D...

**Answer:** A and C

**Explanation:**

**Option A is CORRECT because...**

**Option B is incorrect because...**

**Option C is CORRECT because...**

**Option D is incorrect because...**

[Insert the explanation in clear and lucid language here.]

**Diagram:** [Optional] [Insert the architectural or conceptual diagram here.]

**Reference:** [Insert the references here - which may include links to AWS Documentation, Blog, re:Invent video, Authority YouTube video].

----------------------------------------------------------------------------------------------------------------------------

**ML Specialty has 4 Domains**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Name of the Domain** | **Weight** | **Estimated No. of Questions**  (out of 65 As per weightage %) |
| 1 | Data Engineering | 20% | 13 |
| 2 | Exploratory Data Analysis | 24% | 16 |
| 3 | Modeling | 36% | 23 |
| 4 | ML Implementation and Operations | 20% | 13 |

--------------------------------------Question Section Starts-----------------------------------------------------

Question: 1

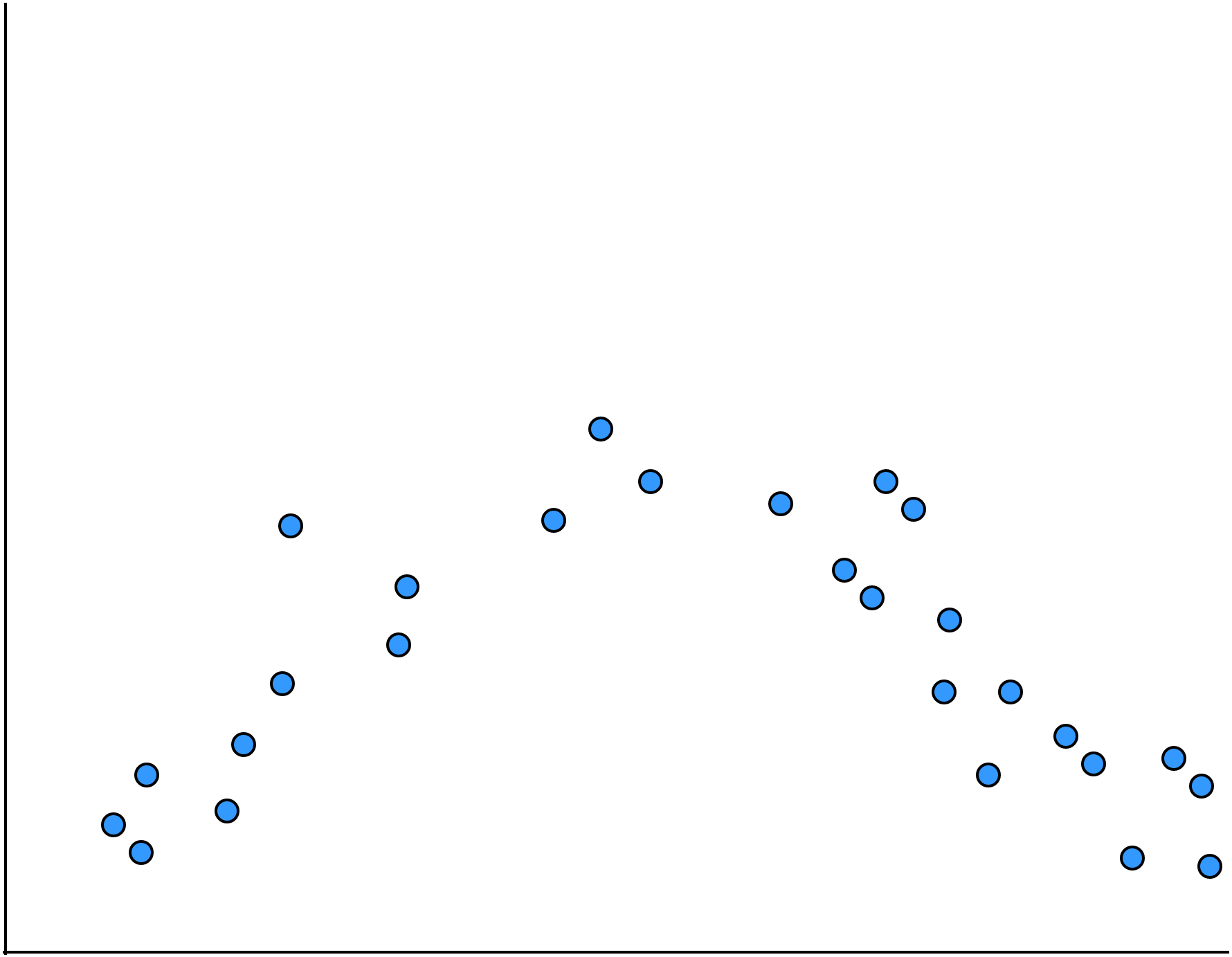
**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Analyze and visualize data for machine learning**

**Domain:** Exploratory Data Analysis

**Question text**:

You are a machine learning specialist working for an oil refinery company. Your team is working on a machine learning problem where you need to determine the relationship between oil well depth and oil well production. In order to select the appropriate machine learning algorithm to use to attempt to solve the oil well production problem, you need to gain a better understanding of your data. For example, what is the correlation between your oil well depth data and your oil well production data?

When you examine your data visually using the python matplotlib library, you find that your data has what looks like a non-Gaussian distribution of oil well depth and oil well production:  
[](https://www.draw.io/?page-id=kvxQOwqLXI_e5wS3bV51&scale=auto#G1J2dEHLgrH_-3FTQjtKVSFbzoDf21m9qY)

Which correlation coefficient would you use to best summarize the strength of the correlation between your oil well depth and oil well production?

1. Covariance correlation coefficient
2. Pearson’s correlation coefficient
3. Spearman’s correlation coefficient
4. Polychoric correlation coefficient

**Answer:** C

**Explanation:**

Option A is incorrect. Covariance is used when you have a Gaussian relationship between your variables.

Option B is incorrect. Pearson’s correlation coefficient is also used when you have a Gaussian relationship between your variables.

Option C is correct. Spearman’s correlation coefficient is used when you have a non-Gaussian relationship between your variables.

Option D is incorrect. The polychoric correlation coefficient is used to understand the relationship of variables gathered via surveys such as personality tests and surveys that use rating scales.

**Reference:**

Please see the Machine Learning Mastery page titled **How to Calculate Correlation Between Variables in Python** (<https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/>), the Wikipedia page titled **Correlation coefficient** (<https://en.wikipedia.org/wiki/Correlation_coefficient>), the the Wikipedia page titled **Polychoric correlation** (<https://en.wikipedia.org/wiki/Polychoric_correlation>), and the Medium article titled **What are Covariance and Correlation coefficients and their significance?** (<https://towardsdatascience.com/covariance-and-correlation-321fdacab168>)

Question: 2

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Analyze and visualize data for machine learning**

**Domain:** Exploratory Data Analysis

**Question text**:

You are a machine learning specialist working for a clothing manufacturer. You have been tasked with building a machine learning model to determine the return on investment (ROI) for advertising a specific clothing line on social media based on the labeled data of past social media campaigns for similar clothing lines.

You decide to run a Pearson’s correlation coefficient to better understand your data correlation. When you calculate your Pearson’s correlation coefficient of social media advertising ROI you get a value of 0.35. What conclusions can you draw from this result?

1. There is a favorable relationship between your past social media advertising and corresponding campaign ROI.
2. There is an unfavorable relationship between your past social media advertising and corresponding campaign ROI.
3. There is no correlation between past social media advertising and the associated ROI.
4. You cannot declare a notable correlation with confidence based on the resulting coefficient.

**Answer:** D

**Explanation:**

Option A is incorrect. Your coefficient value is not high enough to indicate a positive relationship. For a Pearson’s correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

Option B is incorrect. Your coefficient value is not low enough to indicate a negative relationship. For a Pearson’s correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

Option C is incorrect. A coefficient value of 0 or close to 0 indicates no correlation. Your value of 0.35 is not close enough to 0 to indicate no correlation.

Option D is correct. Your coefficient falls into the indeterminate range. For a Pearson’s correlation coefficient to indicate a notable correlation, the coefficient value should be above 0.5 for a positive correlation, or below -0.5 for a negative correlation. Your score is 0.35, which falls into the indeterminate range.

**Reference:**

Please see the Machine Learning Mastery page titled **How to Calculate Correlation Between Variables in Python** (<https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/>), the Wikipedia page titled **Correlation coefficient** (<https://en.wikipedia.org/wiki/Correlation_coefficient>), and the Medium article titled **What are Covariance and Correlation coefficients and their significance?** (<https://towardsdatascience.com/covariance-and-correlation-321fdacab168>)

Question: 3

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for the social media software development division of your company. The social media features of your web applications allow users to post text messages and pictures about their experiences with your company’s products. You need to be able to quickly block posts that contain inappropriate words. You have defined a vocabulary of words deemed inappropriate for your site.

Which algorithm is best suited to your task?

1. Multinomial Naive Bayes
2. Bernoulli Naive Bayes
3. Gaussian Naive Bayes
4. Polychoric Naive Bayes

**Answer:** B

**Explanation:**

Option A is incorrect. The Multinomial Naive Bayes algorithm is best suited for document classification tasks where you wish to know the frequency of a given word from your vocabulary in your observed text. You need to know whether a word from your vocabulary appears in the given post text or not.

Option B is correct. The Bernoulli Naive Bayes algorithm is used in document classification tasks where you wish to know whether a word from your vocabulary appears in your observed text or not. This is exactly what you are trying to accomplish, you need to know whether a word from your vocabulary of inappropriate words appears in the given post text or not.

Option C is incorrect. The Gaussian Naive Bayes algorithm works continuous values in your observations, not discrete values. Your classification problem uses discrete data, the occurrence of a word or not.

Option D is incorrect. There is no Polychoric Naive Bayes algorithm.

**Reference:**

Please see the DatumBox page titled **Machine Learning Blog & Software Development News** (<http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/>), the SebastianRaschka page titled **Naive Bayes and Text Classification – Introduction and Theory** (<http://sebastianraschka.com/Articles/2014_naive_bayes_1.html#3_3_multivariate>), the Medium page titled **Naive Bayes Classifier** (<https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>), the Packt page titled **Machine Learning Algorithms: Implementing Naive Bayes with Spark MLlib** (<https://hub.packtpub.com/machine-learning-algorithms-naive-bayes-with-spark-mllib/>), the Wikipedia article page titled **Naive Bayes classifier** (<https://en.wikipedia.org/wiki/Naive_Bayes_classifier>), and the Medium article page titled **Naive Bayes Explained!** (<https://medium.com/swlh/things-you-never-knew-about-naive-bayes-eb84b6ee039a>)

Question: 4

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for a government agency that uses a series of web application forms to gather citizen data for census purposes. You have been tasked with finding novel user entries as they are entered by your citizens. Where a novel user entry is defined as an outlier compared to the established set of citizen entries in your datastore.

You have cleaned your citizen datastore to remove any existing outliers. You now need to build a model to determine whether new entries on your web application are novel. Which algorithm best fits these requirements?

1. Multinomial Naive Bayes
2. Bernoulli Naive Bayes
3. Principal Component Analysis
4. Support Vector Machine

**Answer:** D

**Explanation:**

Option A is incorrect. The Multinomial Naive Bayes algorithm is best suited for classification tasks where you wish to know the frequency of a given observation. You are trying to determine whether you have a novel observation.

Option B is incorrect. The Bernoulli Naive Bayes algorithm is used in classification tasks where you wish to know whether a known class appears in your observation. You are trying to determine whether you have a novel observation.

Option C is incorrect. The Principal Component Analysis algorithm is used to reduce feature dimensionality. You are trying to determine whether you have a novel observation.

Option D is correct. The Support Vector Machine algorithm can be used when your training data has no outliers and you want to detect whether a new observation is a novel entry.

**Reference:**

Please see the SciKit Learn page titled **1.4. Support Vector Machines** (<https://scikit-learn.org/stable/modules/svm.html>), the SciKit Learn page titled **2.7. Novelty and Outlier Detection** (<https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection>), and the Amazon SageMaker developer guide titled **Principal Component Analysis (PCA) Algorithm** (<https://docs.aws.amazon.com/sagemaker/latest/dg/pca.html>)

Question: 5

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for a translation service company. Your company offers several mobile applications used for translation on smartphones and tablets. As a new feature of one of your translation apps, your company is offering a feature to generate handwritten notes from spoken text.

Which algorithm is the best choice for your new feature?

1. Long Short-Term Memory (LSTM)
2. Convolutional Neural Network
3. Multilayer Perceptron
4. Support Vector Machine

**Answer:** A

**Explanation:**

Option A is correct. The Long Short-Term Memory (LSTM) can work with sequences of spoken language and can be used to generate sequenced output such as handwritten text.

Option B is incorrect. Convolutional Neural Networks are primarily used to work with image data. You are working with sound data, spoken text.

Option C is incorrect. The Multilayer Perceptron algorithm is used primarily for classification predictions and regression predictions. Your problem to solve is to convert spoken text to handwritten text.

Option D is incorrect. The Support Vector Machine algorithm is primarily used for classification, regression, and anomaly detection. Your problem to solve is to convert spoken text to handwritten text.

**Reference:**

Please see the Machine Learning Mastery article titled **When to Use MLP, CNN, and RNN Neural Networks** (<https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>), the SciKit Learn page titled **1.4. Support Vector Machines** (<https://scikit-learn.org/stable/modules/svm.html>), and the Wikipedia page titled **Long short-term memory** (<https://en.wikipedia.org/wiki/Long_short-term_memory#:~:text=Long%20short%2Dterm%20memory%20(LSTM)%20is%20an%20artificial%20recurrent,the%20field%20of%20deep%20learning.&text=LSTM%20networks%20are%20well%2Dsuited,events%20in%20a%20time%20series.>)

Question: 6

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for a healthcare company where you are building a cancer detection model using a linear regression algorithm. You have gathered your data of hundreds of thousands of patients with over 100 features. However, when you train your model you notice that it appears to be over-fitting your data.

Which technique can you use to simultaneously correct the over-fitting and reduce your model complexity by removing less relevant features?

1. Use Ridge Regression
2. Use Lasso Regression
3. Use Stochastic Gradient Descent
4. Use a Gaussian Process

**Answer:** B

**Explanation:**

Option A is incorrect. The Ridge Regression approach would reduce the coefficients in your model but not all the way to 0. Therefore, it reduces complexity, but does not entirely eliminate any of the over 100 features in your data.

Option B is correct. The Lasso Regression approach would reduce some of the coefficients in your model to zero, effectively eliminating some of the over 100 features in your data. This will effectively reduce the complexity of your model.

Option C is incorrect. The Stochastic Gradient Descent approach can use a regularization parameter, but it cannot be used to eliminate features from your dataset.

Option D is incorrect. The Gaussian Process approach is used for regression problems, but it does not work well with high dimensional datasets, i.e. over a few dozen features. Your dataset has over 100 features. Also, it cannot be used to eliminate features from your dataset.

**Reference:**

Please see the Medium article titled **Ridge and Lasso Regression: L1 and L2 Regularization** (<https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b>), and the SciKit Learn page titled **1. Supervised learning** (<https://scikit-learn.org/stable/supervised_learning.html#supervised-learning>)

Question: 7

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for a sports gambling company where you are responsible for building a machine learning model to predict the point spread and over/under of NCAA and NFL games. You have built your custom deep learning model using TensorFlow in SageMaker. You have attempted to train your model on a single GPU but you have noticed that the amount of game data you need to train with exceeds the single GPU capacity.

How can you change your machine learning code to get it to use multiple GPUs with the least amount of effort on your part?

1. Rewrite your model to use the Factorization Machines algorithm
2. Rewrite your code in PySpark and use spark to run your code across multiple GPUs
3. Add Horovod to your code and use its distributed deep learning training framework for TensorFlow
4. Rewrite your model to use the DeepAR Forecasting algorithm

**Answer:** C

**Explanation:**

Option A is incorrect. The Factorization Machines algorithm is used for classification and regression problems, not deep learning predictions.

Option B is incorrect. Rewriting your model in PySpark would require more work compared to using the Horovod framework.

Option C is correct. Using the Horovod distributed deep learning training framework for TensorFlow allows you to easily distribute your training across many GPUs in parallel.

Option D is incorrect. Rewriting your code to use the DeepAR algorithm would require more work compared to using the Horovod framework.

**Reference:**

Please see the GitHub page titled **Horovod** (<https://github.com/horovod/horovod>), the Amazon SageMaker developer guide titled **Use Amazon SageMaker built-in algorithms** (<https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html>), and the Medium article titled **3 Methods for Parallelization in Spark** (<https://towardsdatascience.com/3-methods-for-parallelization-in-spark-6a1a4333b473>)

Question: 8

**Main​ ​Topic​ ​:​** Machine Learning

**Sub​ ​Topic​ ​:​ Select the appropriate model(s) for a given machine learning problem**

**Domain:** Modeling

**Question text**:

You are a machine learning specialist working for a credit card company where you are building a fraud detection model. You have your model built using the XGBoost algorithm and you are now attempting to find the best version of your model by performing automatic model tuning. You are creating your hyperparameter tuning job and you need to select the appropriate technique the job will use to find your best hyperparameters. You wish to run the least number of hyperparameter tuning training jobs as possible.

Which hyperparameter tuning technique is best suited to your requirements?

1. Bayesian optimization
2. Bayesian classification
3. Random search
4. Logistic regression

**Answer:** A

**Explanation:**

Option A is correct. The Bayesian optimization approach to hyperparameter tuning results in less tuning job runs than the random search method.

Option B is incorrect. The Bayesian technique used by the hyperparameter tuning job in SageMaker is Bayesian optimization, which solves the problem using regression, not classification.

Option C is incorrect. The random search method is a valid option when using SageMaker hyperparameter tuning, but the random search approach usually requires running many more training jobs to get the best hyperparameters when compared to the Bayesian optimization approach. You have a requirement to run the least number of hyperparameter tuning training jobs as possible.

Option D is incorrect. Logistic regression is not a valid option when using SageMaker hyperparameter tuning.

**Reference:**

Please see the Amazon SageMaker developer guide titled **How Hyperparameter Tuning Works** (<https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-how-it-works.html>), the AWS Machine Learning blog titled **Amazon SageMaker automatic model tuning now supports random search and hyperparameter scaling** (<https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-automatic-model-tuning-now-supports-random-search-and-hyperparameter-scaling/>), and the AWS Machine Learning blog titled **Simplify machine learning with XGBoost and Amazon SageMaker** (<https://aws.amazon.com/blogs/machine-learning/simplify-machine-learning-with-xgboost-and-amazon-sagemaker/>)